doi: 10.13413/j.cnki.jdxblxb.2025106

桁架穹顶复杂优化求解的遗传算法

张 蕾[1], 仲 洋[2], 曹梦萱[3], 卢 婧[4], 韩霄松[3]
(1. 吉林大学 发展规划处, 长春 130012; 2. 中国电力工程顾问集团 东北电力设计院有限公司, 长春 130021;
3. 吉林大学 软件学院, 长春 130012; 4. 吉林大学 研究生院, 长春 130012)

摘要: 针对传统遗传算法在复杂高维优化问题中适应度计算代价较高的问题, 提出一种基于流形学习与多元线性回归的改进遗传算法 Gamma. Gamma 算法通过流形学习对种群数据进行降维, 并结合 AP 聚类(affinity propagation clustering)与多元线性回归模型, 减少适应度函数的计算次数, 提高算法优化效率. 实验结果表明, Gamma 算法在桁架穹顶结构优化等复杂工程及多个经典 Benchmark 函数上, 均以较少的适应度调用次数达到了与传统方法相近的优化效果, 在处理高维优化问题上应用前景良好, 能有效提高计算效率, 降低时间成本.

关键词: 遗传算法; 流形学习; 代理模型; 复杂优化问题
中图分类号: TP181 **文献标志码:** A **文章编号:** 1671-5489(2025)05-1387-10

Genetic Algorithm for Complex Optimization Solution of Truss Dome

ZHANG Lei[1], ZHONG Yang[2], CAO Mengxuan[3], LU Jing[4], HAN Xiaosong[3]
(1. *Division of Development and Strategic Planning, Jilin University, Changchun* 130012, *China*;
2. *Northeast Electric Power Design Institute Co., Ltd., China Electric Power Engineering Consulting Group, Changchun* 130021, *China*; 3. *College of Software, Jilin University, Changchun* 130012, *China*;
4. *Gradute School, Jilin University, Changchun* 130012, *China*)

Abstract: Aiming at the problem of the high computational cost of fitness in traditional genetic algorithms for complex high-dimensional optimization problems, we proposed an improved genetic algorithm Gamma based on manifold learning and multiple linear regression. The Gamma algorithm reduced the dimensionality of the population data through manifold learning, and combined AP clustering with a multiple linear regression model to reduce the calculation times of fitness function and improve algorithm optimization efficiency. Experimental results show that the Gamma algorithm achieves optimization results similar to traditional methods with fewer fitness calls in complex engineering such as the optimization of truss dome structures and multiple classic Benchmark functions. It has a promising application prospect in handling with complex high-dimensional optimization problems, effectively enhancing computational efficiency and reducing time costs.

Keywords: genetic algorithm; manifold learning; surrogate model; complex optimization problem

遗传算法(genetic algorithm, GA)通过不断在解空间中迭代向更优的解集方向进化从而寻得最优解. GA 基于包含一系列潜在解的种群, 其中每个解被称为染色体, 由一组基因编码构成, 根据“适者

收稿日期: 2025-03-28.

第一作者简介: 张 蕾(1986—),女, 满族, 硕士, 副研究员, 从事智能算法、高教管理和校园规划的研究, E-mail: zhlei@jlu.edu.cn. **通信作者简介:** 韩霄松(1984—), 男, 汉族, 博士, 副教授, 从事机器学习和优化算法的研究, E-mail: hanxiaosong@jlu.edu.cn.

基金项目: 国家自然科学基金(批准号: 62372494; 62372209)、吉林省科技发展计划重点研发项目(批准号: 20220201145GX)、吉林省高教学会科研项目(批准号: JGJX2023C9)和吉林大学研究生教育教学改革项目(批准号: 2023JGY026).

生存，优胜劣汰”的规则，对种群进行选择、交叉、变异操作，产生潜在的包含更优解的种群. 当其解收敛于特定值或者迭代次数达到预先设定的最大迭代数时得到问题的一个最优解. GA因其简单的结构，优秀的全局搜索能力，已被成功应用于参数优化、结构优化及车间调度等领域.

传统遗传算法在处理复杂高维优化问题时，面临适应度函数计算代价高、收敛速度慢以及易陷入局部最优解等问题. 尤其在高维参数空间中，适应度函数的计算复杂度急剧增加，对计算资源的需求极大，从而导致算法效率显著降低. 此外，传统遗传算法在处理复杂约束条件问题时表现不足，这些局限性在工业和科学计算等实际应用中尤其明显. 因此，如何有效降低适应度函数的计算代价，提升遗传算法的整体优化性能，是GA亟待解决的重要问题. 本文提出一种基于流形学习与多元线性回归的改进遗传算法Gamma，以解决传统遗传算法在复杂高维优化问题中适应度计算代价高的问题.

Gamma算法通过流形学习对种群数据进行降维，并结合AP聚类(affinity propagation clustering)与多元线性回归模型，可显著减少适应度函数的计算次数，提高优化效率. 实验结果表明，Gamma算法在桁架穹顶结构优化以及多个经典Benchmark函数上，都以较少的适应度调用次数达到了与传统算法相近的优化效果，在处理复杂高维优化问题上具有良好的应用前景.

通过对遗传算法的流程步骤分析可知，经典遗传算法的每轮迭代都需大量的适应度计算，当面对一些适应度函数复杂的优化问题时，该算法的时间成本巨大，尤其许多工业领域的优化问题还常面临高维参数、计算复杂、适应度依赖于其他模型的问题，已成为影响遗传算法提高性能及应用推广的首要障碍. 代理模型可较好地解决该问题，训练代理模型计算部分染色体适应度，可降低真实适应度评价函数的调用次数，从而提高算法的寻优速度. Grefenstette等[1]提出了使GA通过部分预测适应度的一个模型，降低了实际适应度调用次数，并将其用于优化图像配准技术中. Papadrakakis等[2]提出了利用神经网络模型作为代理模型对优化问题的适应度进行预测，减少了真实适应度的调用次数. Rasheed等[3]提出了一种采用适应度逼近模型对遗传算法进行初始化操作，并指引其在解空间中进化方向的算法. Branke等[4]基于提前获得临近解空间的个体适应度值预测实际值，并通过在拟合中进行插值和回归，获得了较好的实验结果. 文献[5-6]根据之前适应度估计的理念，进行了一些基于非确定环境下的实验. Schmidt等[7]提出了一个基于混合进化适应度估计的模型，该模型参考了群智能算法的思想构建模型的参数，最后再采用该模型预测部分未知个体的适应度，得到了较好的实验结果. Funika等[8]提出了一种协同进化策略，利用部分训练集得到模型逼近适应度，并用于深度学习网络参数训练. 文献[9]利用AP聚类算法结合适应度估计策略改进遗传算法，提高了算法的性能.

上述成果均构建了代理模型替代部分适应度方法的计算，然而当优化参数维度较高时，训练代理模型的精度会下降，从而影响算法的性能. 因此，本文提出一种在低维空间中构建代理模型的策略，先利用流形学习(manifold)对群体进行降维，然后利用聚类算法选择染色体计算适应度，构建多元线性回归(multiple linear regression，MLR)模型作为代理模型计算部分染色体适应度，从而降低真实适应度计算次数，达到提高GA优化速度的效果. 本文算法称为基于流形学习与多元线性回归的改进遗传算法(genetic algorithm with multiple linear regression based on manifold)，简称Gamma算法.

1 相关算法

1.1 流形学习

流形学习是一类非线性降维技术，用于发现高维数据中的低维流形结构. 流形学习的目标是对高维数据进行学习从而计算出可能的低维空间结构，并构造出相应的函数关系，从而降低数据维度[10]. 等距离映射(isometric feature mapping，ISOMAP)[10]、局部线性嵌入(locally linear embedding，LLE)[11]、t分布随机邻居嵌入(t-distributed stochastic neighbor embedding，t-SNE)[12]是3种典型的流形学习算法. 局部线性嵌入思路是假设数据集中的所有点较聚集地分布在一个适当大小的局部邻域，其分布服从线性结构，故在该局部结构中，每个样本点均能通过临近点表示，线性表示所有临近点的矩阵称为权值矩阵. LLE目标是在降维过程中保持其局部权值不变，而根据重排邻域重叠信息得到一个全局上的拓扑排序. LLE构建过程如图1所示.

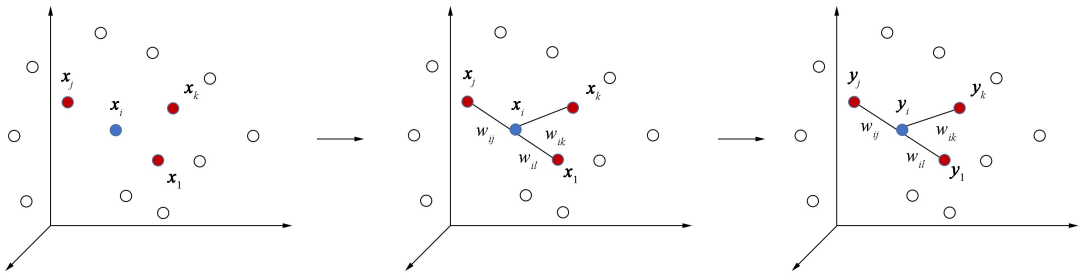


图 1　LLE 构建过程

Fig. 1　LLE construction process

LLE 先利用欧氏距离得到样本点的 k 个近邻点，然后计算样本点的局部重建权值矩阵 $\boldsymbol{W}$，重构误差为

$$\varepsilon(\boldsymbol{W})=\sum_i\left\|\boldsymbol{x}_i-\sum_j w_{ij}\boldsymbol{x}_j\right\|^2,\qquad \sum_j w_j=1, \tag{1}$$

$$C_{jk}=(\boldsymbol{x}_i-\boldsymbol{x}_j)\cdot(\boldsymbol{x}_i-\boldsymbol{x}_k), \tag{2}$$

其中 $\boldsymbol{x}_i$ 为样本点，$\boldsymbol{W}$ 为权值矩阵，C_{jk} 表示局部协方差元素，$\boldsymbol{x}_j$ 和 $\boldsymbol{x}_k$ 为 $\boldsymbol{x}_i$ 的近邻点. 最小化目标函数即式(1)，可得权值矩阵为

$$w_j=\Big(\sum_j C_{jk}^{-1}\Big)\Big/\sum_{l,m}C_{lm}^{-1}. \tag{3}$$

样本点映射到低维空间，映射函数满足

$$\min_{\boldsymbol{Y}}\Phi(\boldsymbol{Y})=\sum_i\left\|\boldsymbol{y}_i-\sum_j w_{ij}\boldsymbol{y}_j\right\|^2 \tag{4}$$

条件，其中 $\Phi(\boldsymbol{Y})$ 可简写为

$$\begin{aligned}&\Phi(\boldsymbol{Y})=\sum_{i,j}M_{ij}(\boldsymbol{y}_i\cdot\boldsymbol{y}_j),\quad M=(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{W}),\\&\text{s.t.}\ \sum_i y_i=1,\quad \frac{1}{N}\sum_i\boldsymbol{y}_i\boldsymbol{y}_i^{\mathrm{T}}=\boldsymbol{I}\end{aligned} \tag{5}$$

的形式，再考虑中心化和单位协方差的限制条件，则问题转化为

$$M\boldsymbol{Y}=\lambda\boldsymbol{Y} \tag{6}$$

的矩阵分解任务，即取 $\boldsymbol{Y}$ 为 M 的最小 m 个非零特征值对应的特征向量.

LLE 计算简单，后续步骤类似于主成分分析，但可处理非线性数据，并能很好地表达数据的内在流形结构，确保数据的本质特征，具有旋转不变性、缩放不变性和平移不变性.

1.2　聚类算法

亲和力传播(affinity propagation，AP)算法是一种新型聚类算法[13]. 该算法的基本思想是将全部数据点都视为潜在的聚类中心(称为 exemplar)，然后通过计算数据点两两之间连线构成一个网络(相似度矩阵)，再通过网络中各条边的消息(responsibility 和 availability)传递计算出各样本的聚类中心. 相比于传统的 K-means 聚类算法，AP 算法一个显著特点是并不需要设定聚簇数，且最后完成迭代所输出的分类结果通常较固定，聚类中心是真实存在的点. 因此，需要处理中等或较小型规模的样本时，常能获得较好效果.

先为 AP 算法指定一种恰当的度量方式，如欧氏距离或余弦相似度等，然后根据选定的度量方式，构建一个相似度矩阵 $\boldsymbol{S}=(s(i,k))_{n\times n}$，再对矩阵中对角线上的元素 $s(k,k)$ 进行设定，该参数称为自相似性，设定的值越大，所对应的个体被选择为聚簇中心的概率越高. 此外，同理计算吸引矩阵 $\boldsymbol{R}=(r(i,k))_{n\times n}$ 及归属矩阵 $\boldsymbol{A}=(a(i,k))_{n\times n}$. 因此，该算法是上述两个矩阵不断更新的过程，更新公式为

$$\begin{aligned}&r_{t+1}(i,k)=s(i,k)-\max_{k'\neq k}\{a_t(i,k')+s(i,k')\},\\&a_{t+1}(i,k)=\min\Big\{0,r_t(k,k)+\sum_{i'\notin\{i,k\}}\max\{0,r_t(i',k)\}\Big\},\quad i\neq k,\\&a_{t+1}(k,k)=\sum_{i'\neq k}\max\{0,r_t(i',k)\},\end{aligned} \tag{7}$$

其中 i 和 k 表示样本.

1.3 多元线性回归

多元线性回归(multiple linear regression, MLR)包含多个变量,可直观表达不同属性在预测时的重要性. 假设 Y 为应变量,$X_1,X_2,\cdots,X_p$ 为 p 个自变量,则可得一个多元回归模型:

$$Y=B_0+B_1X_1+B_2X_2+\cdots+B_pX_p+e, \tag{8}$$

其中 e 表示回归误差,参数 $B_0,B_1,B_2,\cdots,B_p$ 为回归系数. 即每当其余的回归变量均为常数时,X_i 的变化与预测中 Y 的变化成比例. 为方便,可在公式中附加一个虚变量 X_0 及截距 B_0,对所有的 n 次预测,X_0 的取值均为 1,则式(8)可改写为

$$Y=B_0X_0+B_1X_1+B_2X_2+\cdots+B_pX_p+e. \tag{9}$$

最小二乘法(ordinary least squares, OLS)是解决线性回归问题的有效方法,其计算方式是利用最小误差平方和,求解回归系数的偏微分方程以找到最符合所有数据的直线. 为提高 OLS 的泛化能力,本文采用加权最小二乘法(weighted OLS, WLS),WLS 给每个观测值分配一个反映测量不确定度的权重,其最小二乘准则为

$$\varepsilon_\omega=\sum_i\omega_i(Y_i-\hat{Y}_i)^2=\sum_i\omega_i[Y_i-(a+bX_i)]^2. \tag{10}$$

2 流行学习下基于多元线性回归的遗传算法(Gamma)

Gamma 算法对经典遗传算法进行改进,工作流程如下:首先,设定算法各参数(如种群大小、最大迭代次数、染色体长度、基因突变概率、染色体交叉概率等),基于上述参数对群体进行初始化,初代群体中染色体随机产生. 其次,每次迭代时算法基于设定的基因突变概率和交叉概率,运行变异算子和交叉算子操作产生新的个体,从而得到一个临时种群. 最后,在适应度评价阶段,如果适应度函数计算十分复杂,则遗传算法的寻优速度将十分缓慢,因此本文构建了代理模型策略部分替代适应度函数计算.

本文利用局部线性嵌入算法对群体进行降维. 遗传算法作为一个高维曲面上的搜索算法,在实际应用中,该高维面很可能非常繁杂且不连续,甚至无法通过特定的数学公式表达,多元线性回归拟合这类问题较困难. 根据拓扑流形的概念,若上述曲面可在局部同胚于低维流形,则可将该局部的拓扑结构映射到低维空间中,此时对所得低维流形上的重叠坐标部分进行替换,便能得出全局上的拓扑结构. 在低维流形上进行代理模型构建,可在极大减小计算代价的情况下保持相对较高的模型精度.

代理模型训练集的构建对模型效果有重要影响,如果能选取有代表性的染色体进行构建多元线性回归模型,则可能得到更精确的预测效果. 由于 AP 聚类的聚类中心是真实存在的染色体,因此 Gamma 算法在低维流形下,利用 AP 聚类选择训练集中有代表性的染色体(即聚类中心). 本文采用先进先出(first in first out, FIFO)原则更新训练集. 根据遗传算法的特点,代理模型实际上是根据时间维度上的局部拓扑结构构建模型,不需要记录所有染色体的历史信息. 通过采用 FIFO 原则,可淘汰部分过时染色体,避免训练时间过长,并保持与当前群体的高相关性,从而保证预测精度.

Gamma 算法在低维空间中利用 AP 聚类构建训练集,利用 MLR 构建代理模型预测部分染色体适应度. 选择 MLR 是考虑到低维空间中,MLR 可快速构建,从而降低模型训练带来的时间代价. 最后再基于之前直接计算以及预测出的适应度值进行选择操作,形成新一代群体,并进行下一轮迭代,算法流程如图 2 所示.

算法 1 Gamma 算法.

步骤 1) 初始化:设置进化代数计数器 $t=0$,设置最大进化代数 T,设置适应度评价函数 f,函数随机生成 M 个染色体作为初始种群 $P(t=0)$;

步骤 2) 个体评价:利用适应度函数 f 计算种群 $P(t)$ 中每个染色体的适应度;

步骤 3) 传统算子作用在种群上:

选择运算:将选择算子作用于种群;

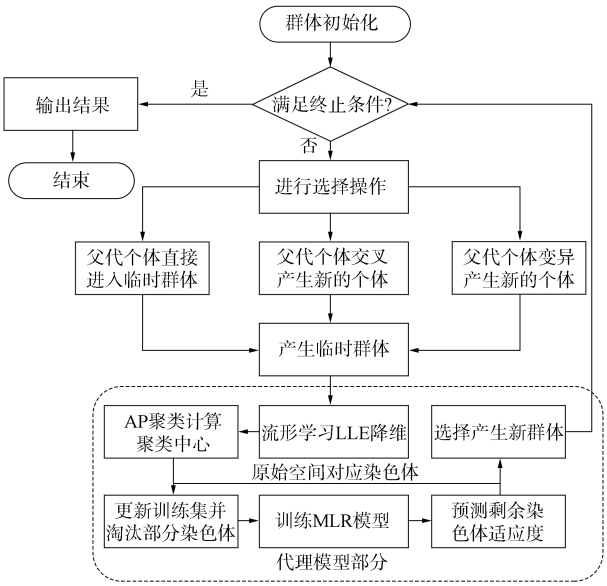


图 2　Gamma 算法流程

Fig. 2　Flow chart of Gamma algorithm

交叉运算：将交叉算子作用于种群；

变异运算：将变异算子作用于种群；

步骤 4）种群 $P(t)$经过选择、交叉、变异运算后得到临时种群 $P(t')$；

步骤 5）if Size(Train(t))$<4\times$Size($P(t)$)

不使用代理模型策略，利用 f 计算适应度，群体加入训练集 Train(t)；

else

基于临时种群 $P(t')$构建代理模型 $S(t)$；

利用 LLE 算法对群体进行降维；

低维空间中 AP 聚类算法获得聚类中心；

调用适应度函数计算聚类中心对应原始空间染色体适应度；

利用 FIFO 原则更新训练集 Trian(t)；

基于 Trian(t)训练 MLR 模型 $S(t)$；

利用 $S(t)$计算其他染色体适应度；

步骤 6）选择算子从临时种群 $P(t)$选择个体产生新种群 $P(t+1)$；

步骤 7）终止条件判断：若 $t=T$，则以进化过程中所得的具有最大适应度个体作为最优解输出，终止计算.

3　数值实验

3.1　Benchmark 函数实验设计

本文基于 Python 环境的遗传算法工具箱 Geatpy2 进行开发，在上述 Benchmark 函数上进行测试，对比经典遗传算法(SGA)和 Gamma 算法的性能. 为验证流形学习和 AP 算法的效果，将 Gamma 算法剪裁成去除 AP(Gamma^{-})和去除 LLE 与 AP(Gamma^{--})两种算法，并进行对比. 为对比不同流形学习降维效果，在 Gamma 算法中将 LLE 换成 t-SNE(Gamma$^{\text{t-SNE}}$)和 IOSMAP(Gamma$^{\text{IOSMAP}}$)进行对比. 对不同算法，实验设置相同的种群大小、训练集容量及迭代次数，采用实数编码，每次实验运行 10 次计算平均结果. 各算法参数设置如下：遗传算法的群体大小为 50，进化迭代次数为 100，训练集选择概率为 0.5，训练集大小为 200，问题维数为 30，算法重复运算 10 次；LLE 降维后维度为 3，eigen_solver＝dense；t-SNE 降维后维度为 3，init＝pca，method＝barnes_hut，angle＝0.2，n_iter＝

1 000；ISOMAP降维后维度为3；AP聚类算法的Preference系数设置为中位数，damping系数设置为0.5. 本文选择11个Benchmark函数作为测试函数，各函数信息列于表1.

表1 Benchmark函数信息

Table 1 Information of Benchmark functions

序号	函数名称	函数方程	定义域
1	Step	$f(x)=\sum_{i=1}^{n-1}(\lfloor x_i+0.5\rfloor)^2$	$[-100,100]^n$
2	Sphere	$f(x)=\sum_{i=1}^{n}(x_i)^2$	$[-100,100]^n$
3	Schwefel	$f(x)=418.982\ 9\times n+\sum_{i=1}^{n}[-x_i\sin(\sqrt{\lvert x_i\rvert})]$	$[-500,500]^n$
4	Ackley	$f(x)=-20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i}^{n}x_i^2}\right\}-\exp\left\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right\}+20+\mathrm{e}$	$[-32,32]^n$
5	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)+(x_i-1)^2]$	$[-2.048,2.048]^n$
6	Griewank	$f(x)=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-512,512]^n$
7	Ridge	$f(x_1\cdots x_n)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$，$-64\leqslant x_i\leqslant 64$，minimun $f(0,\cdots,0)=0$	$[-64,64]^n$
8	Whitley	$f(x_1\cdots x_n)=\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{[100(x_i^2-x_j)^2+(1-x_j)^2]^2}{4\ 000}-\cos[(100(x_i^2-x_j)^2+(1-x_j)^2)+1]$	$[-10.24,10.24]^n$
9	Mod double	$f(x_1\cdots x_n)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}(x_j-j)^2\right)$，$-10.24\leqslant x_i\leqslant 10.24$，minimun $f(1,2,\cdots,n)=0$	$[-10.24,10.24]^n$
10	Quartic	$f(x_0\cdots x_n)=\sum_{i=0}^{n}ix_i^4+\mathrm{random}[0,1]$，$-1.28\leqslant x_i\leqslant 1.28$，minimun $f(0,\cdots,0)=0+\mathrm{noise}$	$[-1.28,1.28]^n$
11	Rastrigin	$f(x_1\cdots x_n)=10n+\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i))$，$-5.12\leqslant x_i\leqslant 5.12$，minimun $f(0,\cdots,0)=0$	$[-5.12,5.12]^n$

3.2 数值实验结果

数值实验结果列于表2. 图3为每个Benchmark函数的进化曲线对比结果. 表2列出了SGA，Gamma，Gamma^{-}，Gamma^{--}，$\mathrm{Gamma}^{\text{t-SNE}}$和$\mathrm{Gamma}^{\text{IOSMAP}}$算法在不同Benchmark函数下的性能指标值，包括最佳适应度(best)、平均适应度(avg)、适应度标准差(std)和算法平均运行时间(avg time). 由表2可见，SGA运行最快，效果最好，但这并不影响本文结论，因为在本文假设中，代理模型中涉及的多元线性回归算法、流形学习算法以及聚类算法事实上都有一定的算法时间复杂度，而上述实验中使用的测试函数均为相对简单的函数，故在使用了代理模型后，由于增加了整体算法的复杂度，时间消耗升高，因此本文主要对比适应度实际调用次数. 实验中SGA的适应度调用次数为5 000次，其余算法在2 500次左右，实验结果表明，当适应度计算超过0.005 s时，Gamma算法在计算速度上有明显优势. 由算法的时间代价可见，除SGA外Gamma^{--}最快是由于其剪裁了耗时的LLE和AP聚类，降低了算法复杂度，$\mathrm{Gamma}^{\text{t-SNE}}$最慢则是由于t-SNE算法复杂度较高，其余3种算法时间复杂度相差较小. 由算法的优化结果可见，使用了LLE降维的Gamma和Gamma^{-}算法性能最优，也最稳定，表明LLE算法代理模型的效果优于t-SNE和ISOMAP算法，且t-SNE时间代价太大，实际应用中应在LLE和IOSMAP中选择. 而Gamma算法整体性能优于Gamma^{-}，表明AP聚类选择的训练集优于随机选择，且AP算法带来的时间损耗较小. 综上可见，基于LLE和AP算法的Gamma算法综合性能最优.

表 2　数值实验结果

Table 2　Numerical experimental results

函数名称	统计量	SGA	$Gamma^{--}$	$Gamma^{-}$	Gamma	$Gamma^{ISOMAP}$	$Gamma^{t\text{-}SNE}$
Step	best	237	26 944	23 265	18 216	32 076	24 724
	avg	428.8	20 893.4	18 697.6	14 253.8	23 928.5	20 504.3
	std	120.329	2 555.366	2 334.586	2 207.307	4 091.291	2 785.198
	avg time	0.131	0.299	3.741	9.573	6.848	209.577
Sphere	best	170	26 303	23 169	20 917	30 004	28 913
	avg	407.3	20 024.4	18 441.1	16 452.9	23 973	21 838.8
	std	146.889	2 910.424	2 566.201	2 615.641	4 574.733	3 663.102
	avg time	0.165	0.440	3.862	9.038	6.588	209.663
Schwefel	best	6 957	7 401	7 652	7 462	7 377	7 536
	avg	7 911.2	6 911.9	7 031.3	6 847	7 048.8	7 068.7
	std	348.939	361.036	401.301	406.446	192.029	389.11
	avg time	0.328	0.361	3.769	9.271	6.630	213.026
Ackley	best	5	18	18	17	19	18
	avg	5.8	17.4	17.5	16.8	1 738	17.5
	std	0.6	0.489	0.67	0.4	0.6	0.67
	avg time	0.419	0.405	3.922	9.092	6.786	209.627
Rosenbrock	best	67	1 869	1 483	1 147	1 948	2 071
	avg	110.6	1 454.8	1 106.9	988.7	1 585.7	1 516.2
	std	24.662	268.634	190.82	109.576	195.421	363.62
	avg time	0.130	0.262	3.720	8.539	6.714	213.601
Griewank	best	2	182	157	111	175	176
	avg	3.5	139.3	127.7	96.8	136.7	145.6
	std	0.8	20.852	15.511	11.906	26.914	18.916
	avg time	0.230	0.309	3.796	9.270	6.881	212.966
Ridge	best	10 766	21 135	23 928	21 344	22 511	21 342
	avg	15 916.2	18 590.9	18 182	17 669.5	18 105.5	17 968
	std	2 986.82	1 378.466	2 872.178	1 852.52	2 332.813	2 391.832
	avg time	0.194	0.319	3.787	8.390	6.579	204.042
Whitley	best	1 791	580 756	406 467	278 411	723 532	493 780
	avg	2 429.1	328 563.8	292 212.8	227 317.1	447 851.9	370 578.5
	std	424.242	120 799.227 9	70 932.259 86	43 376.401	130 313.233 2	86 763.643 8
	avg time	4.212	3.385	6.696	12.349	6.258	207.285
Moddouble	best	15 600	34 060	30 250	23 144	36 806	34 781
	avg	15 655.8	30 063.9	25 795.5	21 389.8	32 450.7	30 569.6
	std	34.507	2 398.571	2 416.838	1 437.424	2 058.639	2 953.823
	avg time	0.312	0.472 6	3.915	9.723	6.881	209.037
Quartic	best	0	25	21	15	32	26
	avg	0	15.4	11.6	9.4	21.6	20.1
	std	0	4.903	4.454	3.2	6.374	4.158
	avg time	0.133	0.283	3.711	9.337	6.316	211.52
Rastrigin	best	204	263	273	256	280	262
	avg	226.5	244.7	246.1	234.2	260.5	244.9
	std	10.623	11.899	18.338	15.203	16.847	12.668
	avg time	0.407	0.409	3.853	9.095	6.826	205.254

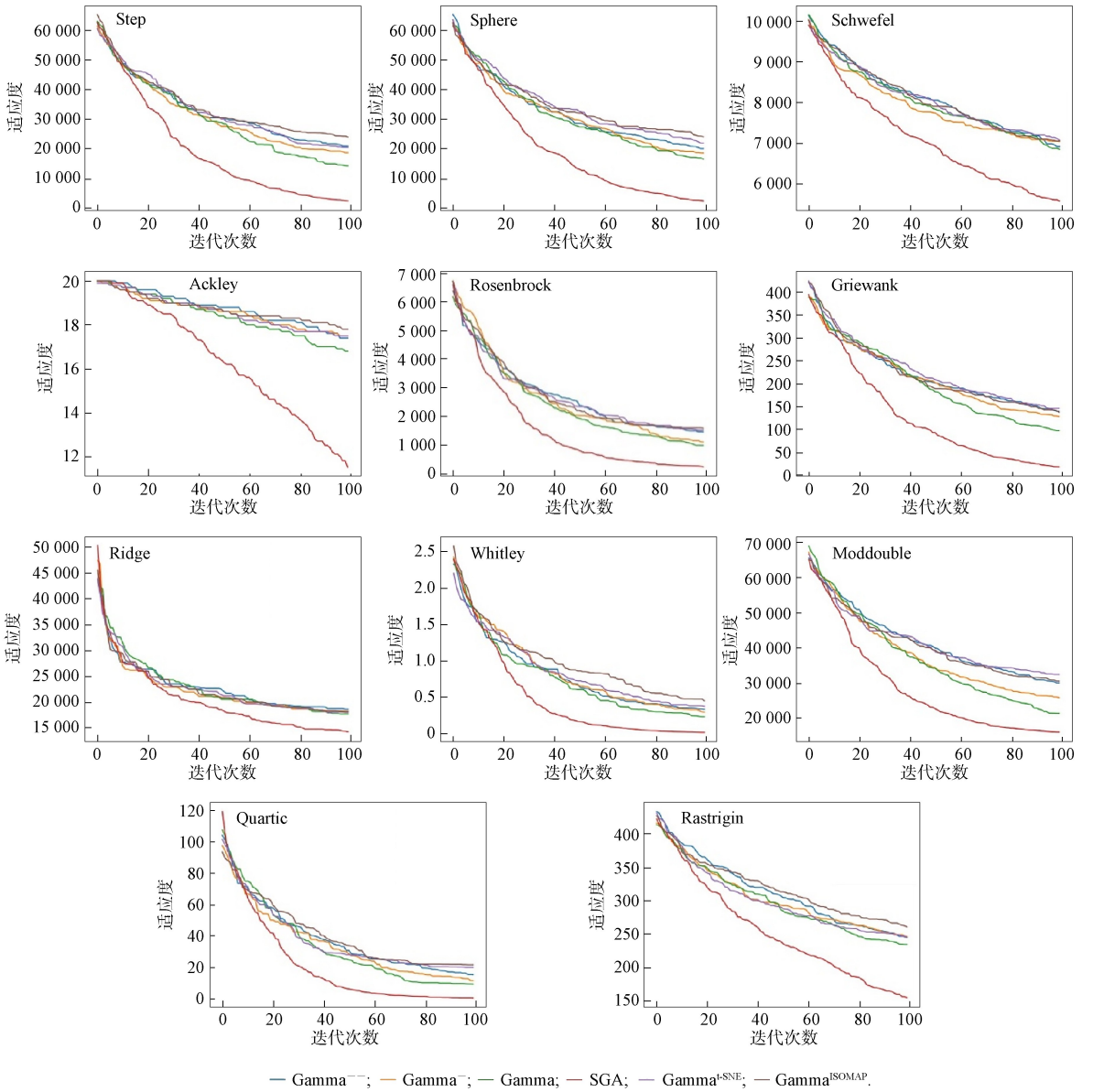


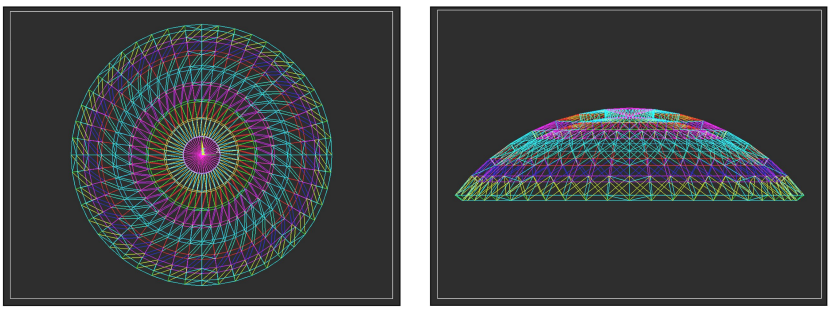
图3 Benchmark函数进化曲线

Fig. 3 Evolution curves of Benchmark functions

4 结构优化实验

在高校校园建设中，桁架穹顶结构主要用于满足大跨度、轻量化和高强度建筑的需求，多应用于多功能体育馆、礼堂和图书馆等建筑. 因此，如何在结构优化中应用高效算法，降低用料成本并提高建筑的结构稳定性十分重要. 下面将Gamma算法用于结构优化实例中，以进一步验证Gamma算法的性能. 结构优化是指在一定约束条件下，根据预定目标得到最优解. 实验使用SGA，Gamma，$\text{Gamma}^{\text{t-SNE}}$和$\text{Gamma}^{\text{IOSMAP}}$算法优化如图4所示桁架结构穹顶中连杆的直径，目标为在0.5 m厚积雪静力作用且形变满足约束条件下整体用料最少. 该优化问题的目标依赖于利用有限元法进行静力分析，使用ANSYS软件实现. 由于用ANSYS软件进行静力分析较耗时，因此先利用基于代理模型的Gamma算法进行优化.

实验采用的桁架结构为双层穹顶，内外半径分别为24，25 m，外表面高度为10 m，内外顶点距离1 m，内底距外底0.8 m. 每个曲面有一个顶点和8层(环)，每层48个节点，该穹顶共有770个节点和



(A) 俯视图　　(B) 侧视图

图 4　桁架穹顶结构

Fig. 4　Truss dome structure

2 849 根连杆. 将相同长度的连杆分为 24 种，内底层上的 48 个节点和外底层上形成六边形的 8 个节点用 3 个连杆固定.

计算 0.5 m 厚积雪作用下，每个节点的作用力为

$$S=2\times\pi\times R\times H, \tag{11}$$

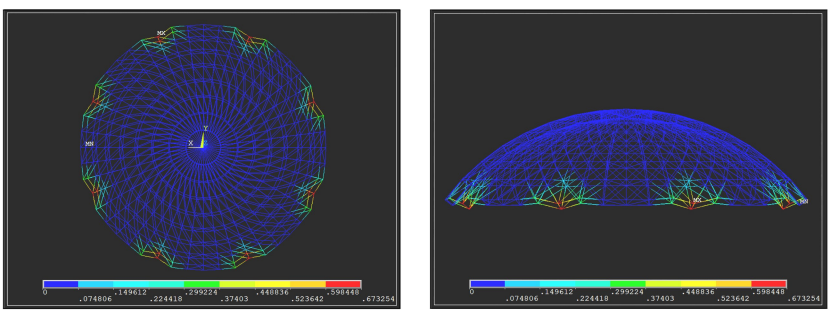
$$F=(\rho\times h\times g\times S)/385,$$

其中 S 为穹顶外层表面积，R 为半径(25 m)，H 为穹顶高度(10 m)，F 为外层每个节点平均受力，ρ 为积雪密度(87.67 kg/m^3)，V 为积雪深度(0.5 m)，g 为重力加速度(9.8 N/kg)，385 为外层节点数量. 则本文实验适应度为

$$G'(g_i-\theta)=\begin{cases}\exp\{-(g_i-\theta)^2/(2\sigma^2)\}, & g_i-\theta\leqslant 0,\\ \exp\{-(g_i-\theta)^2/\sigma^2\}, & g_i-\theta>0,\end{cases} \tag{12}$$

$$f=\Big(\sum_{i=1}^{N}A_i\Big)^{-1}+C\cdot\sum_{i=1}^{N}G'(g_i-\theta),$$

其中 N 为穹顶连杆数量(2 849 个)，A_i 为第 i 根连杆截面积，C 为常系数(本文实验设定为 $C=2$)，g_i 为利用有限元法计算得到的第 i 根连杆应力，θ 为连杆最大承受应力(2.35×10^9 N)，G'的定义是为惩罚过度的压力并防止适应性波动. 可见，适应度函数是一个依赖于 ANSYS 软件的黑箱且计算代价巨大的模型. 图 5 为穹顶结构 ANSYS 受力分析结果.



(A) 俯视图　　(B) 侧视图

图 5　穹顶结构 ANSYS 受力分析结果

Fig. 5　ANSYS force analysis results of dome structure

实验结果表明，经过 80 次计算，Gamma，Gamma$^{\text{t-SNE}}$和 SGA 算法的适应度几乎相同，并且优于 Gamma$^{\text{IOSMAP}}$算法，如图 6 所示. 由图 6 可见，Gamma 和 SGA 算法的最佳适应度相对接近，且优于 Gamma$^{\text{t-SNE}}$算法. Gamma 算法在迭代前期的效果优于其他算法，如果需要快速得到优化结果，则 Gamma 算法是一个较好的选择. 由时间代价可见，由于 ANSYS 计算复杂度较高，SGA 算法调用 ANSYS 的次数最多，时间消耗最长，100 次迭代耗时约 330 s，与 SGA 算法相差较小是由于 t-SNE 消耗较大，平均约 300 s，Gamma 和 Gamma$^{\text{IOSMAP}}$算法时间消耗最少，约 164 s. 实验结果充分展示了 Gamma 算法在处理复杂工程问题时的计算精度和时间代价的优势.

综上所述，针对传统遗传算法在复杂高维优化问题中适应度计算代价较高的问题，本文提出了

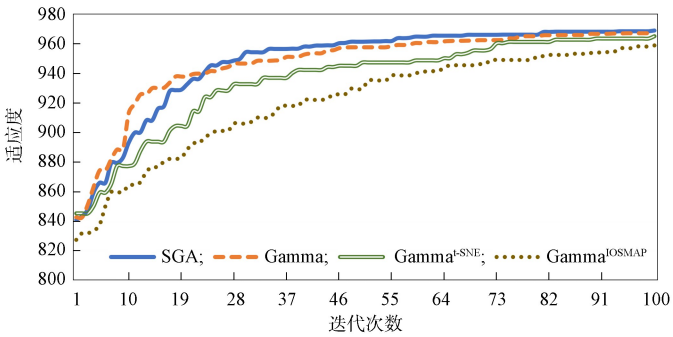


图6 穹顶优化算法进化曲线

Fig. 6 Evolution curves of dome optimization algorithm

一种基于流形学习与多元线性回归的改进遗传算法Gamma. Gamma算法引入了流形学习、多元线性回归与聚类分析等方法构建代理模型. 实验结果表明，Gamma算法可有效降低适应度函数调用次数，并保持较高的预测精度. 在穹顶桁架结构优化问题中的应用进一步验证了Gamma算法在处理复杂优化问题中的高效性和实用价值.

参 考 文 献

[1] GREFENSTETTE J J, FITZPATRICK J M. Genetic Search with Approximate Function Evaluations [C]// Proceedings of the First International Conference on Genetic Algorithms and Their Applications. [S. l.]: Psychology Press, 2014: 112-120.

[2] PAPADRAKAKIS M, LAGAROS N D, TSOMPANAKIS Y. Structural Optimization Using Evolution Strategies and Neural Networks [J]. Computer Methods in Applied Mechanics and Engineering, 1998, 156: 309-333.

[3] RASHEED K, HIRSH H. Informed Operators: Speeding Up Genetic-Algorithm-Based Design Optimization Using Reduced Models [C]//Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation. New York: ACM, 2000: 628-635.

[4] BRANKE J, SCHMIDT C. Faster Convergence by Means of Fitness Estimation [J]. Soft Computing, 2005, 9(1): 13-20.

[5] JIN Y C, BRANKE J. Evolutionary Optimization in Uncertain Environments—A Survey [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(3): 303-317.

[6] PAENKE I, BRANKE J, JIN Y C. Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(4): 405-420.

[7] SCHMIDT M D, LIPSON H. Coevolution of Fitness Predictors [J]. IEEE Transactions on Evolutionary Computation, 2008, 12(6): 736-749.

[8] FUNIKA W, KOPEREK P. Co-evolution of Fitness Predictors and Deep Neural Networks [C]//International Conference on Parallel Processing and Applied Mathematics. Cham: Springer International Publishing, 2017: 555-564.

[9] HAN X S, LIANG Y C, LI Z G, et al. An Efficient Genetic Algorithm for Optimization Problems with Time-Consuming Fitness Evaluation [J]. International Journal of Computational Methods, 2015, 12(1): 1350106-1-1350106-13.

[10] TENENBAUM J B, SILVA V, LANGFORD J C. A Global Geometric Framework for Nonlinear Dimensionality Reduction [J]. Science, 2000, 290(5500): 2319-2323.

[11] ROWEIS S T, SAUL L K. Nonlinear Dimensionality Reduction by Locally Linear Embedding [J]. Science, 2000, 290(5500): 2323-2326.

[12] VAN DER LAURENS M, HINTON G. Visualizing Data Using t-SNE [J]. Journal of Machine Learning Research, 2008, 9(86): 2579-2605.

[13] BRENDAN J F, DELBERT D. Clustering by Passing Messages between Data Points [J]. Science, 2007, 315(5814): 972-976.

(责任编辑：韩 啸)