

# 基于粒子群优化与高斯过程的协同优化算法

张 研<sup>1,2</sup>, 苏国韶<sup>1,3</sup>, 燕柳斌<sup>1</sup>

(1. 广西大学土木建筑工程学院, 广西 南宁 530004;

2. 广西大学工程防灾与结构安全教育部重点实验室, 广西 南宁 530004;

3. 广西防灾减灾与工程安全重点实验室, 广西 南宁 530004)

**摘要:** 对于适应度函数计算耗时较大的工程优化问题,采用仿生智能优化算法求解时常遇到由于适应度函数评价次数过大而导致计算量过高的瓶颈问题。针对上述问题,提出一种基于粒子群优化(particle swarm optimization, PSO)算法与高斯过程(Gaussian process, GP)机器学习方法的协同优化算法(PSO-GP)。该算法在寻优过程中采用 GP 近似模型来构建决策变量与适应度函数值之间的映射关系,在 PSO 全局寻优过程中不断地总结寻优历史经验的基础上,预测可能包含全局最优解的搜索区域,以优化粒子群飞行的方向。多个测试函数的优化结果表明,该算法是可行的,与基本 PSO 算法相比,在获得全局最优解的前提下,可显著减小寻优过程中的适应度函数评价次数,寻优效率较高,在高计算代价复杂工程优化问题的求解上具有良好的应用前景。

**关键词:** 优化算法; 粒子群优化; 高斯过程; 函数优化

中图分类号: TP 181

文献标志码: A

DOI:10.3969/j.issn.1001-506X.2013.06.36

## Cooperative optimization algorithm based on particle swarm optimization and Gaussian process

ZHANG Yan<sup>1,2</sup>, SU Guo-shao<sup>1,3</sup>, YAN Liu-bin<sup>1</sup>

(1. School of Civil Engineering and Architecture, Guangxi University, Nanning 530004, China;

2. Key Laboratory of Disaster Prevention and Structural Safety, Guangxi University, Nanning 530004, China;

3. Guangxi Key Laboratory of Disaster Prevention and Engineering Safety, Nanning 530004, China)

**Abstract:** The large numbers of fitness function evaluation are needed when the engineering optimization problems with time consuming fitness evaluations are solved using a bionic intelligent optimization algorithm. This poses a serious impediment to the field of the bionic intelligent optimization algorithm for the unacceptable high cost of calculation. A cooperative optimization algorithm based on particle swarm optimization (PSO) algorithm and Gaussian process (GP) machine learning for solving computationally expensive optimization problems is presented. GP is used as a surrogate of the real fitness function to prevent frequent fitness function evaluation and predict the most promising solutions before searching the global optimum solution using PSO during each iteration step. The results of study show that the proposed algorithm is much more economical to achieve reasonable accuracy with much less fitness evaluations when solving the optimization problems of the benchmark functions compared with the basic PSO. The proposed algorithm seems very promising to solve the time-consuming optimization problems.

**Keywords:** optimization algorithm; particle swarm optimization (PSO); Gaussian process (GP); function optimization

## 0 引言

近年来,为了可靠地解决全局优化问题,人们已经将单纯形、共轭梯度法、牛顿法等解析确定型局部优化算法的研

究兴趣,转向探讨随机性全局优化方法。有效且具有普遍适应性的随机全局优化方法,是模拟自然现象而发展起来的一系列仿生智能优化算法,如模拟退火算法<sup>[1]</sup>、遗传算法<sup>[2]</sup>、群智能算法<sup>[3]</sup>等。

收稿日期:2012-02-16; 修回日期:2013-02-28; 网络优先出版日期:2013-06-05。

网络优先出版地址: <http://www.cnki.net/kcms/detail/11.2422.TN.20130605.1341.006.html>

基金项目:国家自然科学基金(51069001);广西理工科学实验中心重点项目(LGZX201001);广西重点实验室系统性研究项目(2012ZDX10)资助课题

采用仿生智能优化方法求解复杂优化问题时,一般利用适应度函数值来评价随机解的优劣,通常需要大量的适应度函数评价才能搜索出全局最优解。对于适应度函数计算代价较高的复杂优化问题,过多的适应度函数评价将导致计算代价过高,从而影响了仿生智能优化算法的实用性。例如,对于复杂工程结构优化问题,很难用一个显式来表达决策变量与适应度函数值之间的关系,通常需要借助数值仿真计算来建立这种关系,为了获得合理的适应度函数值,往往需要进行高精度的大规模数值计算。对于此类复杂优化问题的求解,由于每一次适应度函数评价的计算耗时较大(达数小时以上),寻优过程中大量的甚至是成千上万次的适应度函数评价必将导致总的结构数值计算量过于浩大,进而导致计算代价过高的问题,从而迫使获得全局最优解的愿望变得不切实际。因此,如何以较少的适应度函数评价次数获得全局最优解,成为复杂优化问题亟待解决的关键技术难题。

对于高计算代价的优化问题,国内外一些学者进行了深入的研究,其中,采用多项式回归模型、神经网络模型等近似模型来构建决策变量与适应度函数值之间的关系,避免大量的适应度函数评价,已成为当前解决此类优化问题的有效途径之一<sup>[4-5]</sup>。

但是,由于问题的高度复杂性,以往的方法还存在着一些不足之处。

(1) 在优化策略方面,前人的研究中,首先利用多个决策变量进行适应度函数计算,构建若干个样本,根据样本信息采用多项式响应面、神经网络或支持向量机等模型来构建真实适应度函数的近似模型,即在优化过程适应度评价中,采用近似模型直接取代真实适应度函数,进而利用优化算法搜索最优解<sup>[6]</sup>。值得指出的是,这种优化策略存在着过于依赖样本的问题,由于样本质量决定了近似模型的回归精度,一旦学习样本代表性差或偏离全局最优值较远,必将导致近似模型的预测精度不高,进而导致寻优精度不高或容易陷入局部最优解的问题。

(2) 在近似模型方面,对于高维度、多峰值的非线性函数,多项式回归模型回归精度低、对于复杂优化问题适应性较弱;神经网络存在着最优网络结构不易确定,小样本问题推广能力不高的局限性<sup>[7]</sup>;支持向量机存在着大规模训练样本难以实施,核函数参数难以确定的公开问题<sup>[8]</sup>。

针对上述不足,基于群智能优化与机器学习相集成的思路,提出一种粒子群优化(particle swarm optimization, PSO)与高斯过程(Gaussian process, GP)机器学习相结合的智能协同优化新算法(PSO-GP),通过利用GP技术提高PSO寻优过程的“智能化”,使PSO全局寻优过程中尽可能减小适应度函数评价次数,达到高效快速逼近全局最优解的目的,为解决高计算代价的复杂优化问题提供一条新的途径。

## 1 PSO 算法

PSO算法具有容易实现、算法简单、参数较少且能有效

解决全局优化问题等优点<sup>[9-12]</sup>。算法首先随机产生一群随机粒子,然后获得每一次寻优迭代中的两个“极值”:一个是考虑粒子本身经验,粒子个体目前发现的最优值 $P_{id}$ ;另一个是考虑粒子社会经验,整个粒子群当前找到的最优值 $P_{gd}$ ;习惯上将前者称为“个体极值”,后者称为“全局极值”。借助上述两个“极值”通过式(1)来调整赋予每个粒子的两个指标:速度和位置,达到指导粒子飞行状态,最终寻找到全局最优值的目的。

$$\begin{cases} v_{id}(k+1) = \omega v_{id}(k) + c_1 r_1 (p_{id} - x_{id}(k)) + c_2 r_2 (p_{gd} - x_{id}(k)) \\ x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \end{cases} \quad (1)$$

式中, $v_{id}(k+1)$ 、 $v_{id}(k)$ 分别为第 $i$ 粒子在第 $k+1$ 次、第 $k$ 次迭代中第 $d$ 维度上的速度; $x_{id}(k+1)$ 、 $x_{id}(k)$ 分别为第 $i$ 粒子在第 $k+1$ 次、第 $k$ 次迭代中第 $d$ 维度上的位置; $c_1$ 和 $c_2$ 为学习因子,通常 $c_1 = c_2 = 1.8 \sim 2.0$ ;  $r_1$ 、 $r_2$ 为均匀分布在 $(0, 1)$ 之间的随机数; $\omega$ 是惯性权重,为了避免算法后期在全局最优解附近难以收敛这个问题,其值应随算法迭代的进行而减小,一般依据下式进行线性减小,即

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} t \quad (2)$$

式中, $\omega_{\max}$ 、 $\omega_{\min}$ 分别为最大、最小惯性权重,一般取 $\omega_{\max} = 0.9$ 、 $\omega_{\min} = 0.4$ ;  $t_{\max}$ 为最大迭代数; $t$ 为当前迭代数。

## 2 GP 机器学习原理

GP是近年来在机器学习领域流行的一种新型方法,它是基于高斯随机过程与贝叶斯学习理论的一种重要的随机过程。由于其具有参数自适应获取、良好的适应性等突出优点以及能够很好的处理高维数、小样本、非线性等复杂问题,近年来备受关注<sup>[13-14]</sup>。

GP是随机变量的一种集合,该集合中任意个数随机变量的组合均服从联合高斯分布。假设对于 $n \geq 1$ 的一族随机变量 $\mathbf{X}$ ,则其相应的GP集合 $f(\mathbf{X})$ 符合联合 $n$ 维高斯分布。GP是由均值 $m(\mathbf{X})$ 和协方差函数 $k(\mathbf{X}, \mathbf{X}')$ 唯一确定的,表示如下:

$$f(\mathbf{X}) \sim \text{GP}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X}')) \quad (3)$$

假设有 $n$ 个学习样本组成的训练集合 $\mathbf{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, 2, \dots, n\}$ ,  $\mathbf{x}_i$ 是 $d$ 维输入向量,  $y_i \in \mathbf{R}$ 是输出目标值标量。常用 $d \times n$ 维矩阵 $\mathbf{X}$ 表示 $n$ 个 $d$ 维输入向量,向量 $\mathbf{y}$ 表示 $i$ 个输出标量,则训练集合简写为 $\mathbf{D} = (\mathbf{X}, \mathbf{y})$ 。

实际中通常要考虑噪声对模型的影响,则加上高斯噪声的标准回归模型为

$$\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\varepsilon} \quad (4)$$

噪声 $\boldsymbol{\varepsilon}$ 符合均值为0,方差为 $\sigma_n^2$ 的高斯分布,即

$$\boldsymbol{\varepsilon} \sim N(0, \sigma_n^2) \quad (5)$$

当有新的测试输入矩阵 $\mathbf{X}^*$ ,GP模型根据总结的先验知识预测与 $\mathbf{X}^*$ 对应的输出向量 $\mathbf{y}^*$ 。输出向量 $\mathbf{y}$ 的先验分布为

$$\mathbf{y} \sim N(0, \mathbf{K} + \sigma_n^2 \mathbf{I}) \quad (6)$$

式中,  $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X})$  为满足对称正定要求的  $n \times n$  阶协方差矩阵, 其中任一元素  $k_{ij}$  度量了  $x_i$  和  $x_j$  的相关性;  $\mathbf{I}$  为  $n \times n$  阶单位矩阵。

训练样本的输出向量  $\mathbf{y}$  与测试输出向量  $\mathbf{y}^*$  对应的回归函数  $f^*$  组成的联合高斯先验分布为

$$\begin{bmatrix} \mathbf{y} \\ f^* \end{bmatrix} \sim N \left( 0, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & \mathbf{K}(\mathbf{X}, \mathbf{X}^*) \\ \mathbf{K}(\mathbf{X}, \mathbf{X})^\top & k(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix} \right) \quad (7)$$

式中,  $\mathbf{K}(\mathbf{X}, \mathbf{X}^*)$  为测试输入矩阵  $\mathbf{X}^*$  与输入向量矩阵  $\mathbf{X}$  形成的  $n \times 1$  阶协方差矩阵, 简称为  $\mathbf{K}(\mathbf{X}^*)$ ;  $k(\mathbf{X}^*, \mathbf{X}^*)$  为测试输入矩阵  $\mathbf{X}^*$  本身的协方差矩阵, 简称为  $k(\mathbf{X}^*)$ 。

GP 模型常用的协方差函数为

$$k_y(x_p, x_q) = \sigma_f^2 \exp \left( -\frac{1}{2l^2} (x_p - x_q)^2 \right) + \sigma_n^2 \delta_{pq} \quad (8)$$

式中,  $l, \sigma_f, \sigma_n$  为超参数, 不同的超参数组合会影响学习与预测效果。通常采用极大似然法自适应获取最优超参数组合<sup>[15]</sup>。

当给定新的测试输入矩阵  $\mathbf{X}^*$ , 依据贝叶斯原理, 利用先验知识预测与  $\mathbf{X}^*$  对应的可信度最高的输出值。采用完成学习的 GP 模型预测  $f^*$  的均值和方差为

$$\bar{f}^* = \mathbf{K}^\top(\mathbf{X}^*) (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \quad (9)$$

$$\sigma^{*2} = k(\mathbf{X}^*) - \mathbf{K}^\top(\mathbf{X}^*) (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{X}^*) \quad (10)$$

### 3 PSO-GP 协同优化算法

算法的构建思想是: 为了有效减少优化过程中适应度函数评价次数, 在 PSO 寻优过程中通过 GP 机器学习技术不断地总结寻优经验, 凭借粒子寻优过程中的优化踪迹建立寻优经验知识库, 借助 GP 对知识库进行学习, 总结优化经验, 从而建立表达决策变量与适应度函数值之间非线性映射关系的 GP 模型, 将该模型作为具有函数对应关系的“黑箱”代替真实适应度函数行使评价粒子优劣的权力; 接着, 采用 GP 模型预测个体品质最优的粒子, 以该粒子标记可能最优区域, 指引粒子群飞向该区域, 当粒子群在该区域搜索到可能最优解信息时, 利用该信息作为新知识源顶替知识库中品质最差的粒子, 通过更新知识、逐步逼近最优区域提高 GP 模型的预测能力, 从而预测出更加接近最优解的粒子, 如此循环迭代直至寻找到全局最优解。

该算法主要有以下 3 个关键技术。

#### (1) 近似适应度函数评价的 GP 模型

粒子群准备飞行时, 对随机初始几代粒子进行真实适应度函数评价, 利用此信息组建寻优经验知识库。借助 GP 对知识库进行学习, 建立表达决策变量与适应度函数值之间非线性映射关系的 GP 模型。在 PSO 寻优过程中, 利用 GP 模型近似取代真实适应度函数, 对随机解的适应度进行评价。

#### (2) 预测全局最优解

为了充分发挥 GP 模型对粒子适应度进行评价无需调

用真实适应度函数这一优点, 在粒子群飞行过程中循环产生  $q$  代粒子, 多次采用 GP 模型对粒子适应度进行评价, 寻找  $q$  代粒子中适应度最小的粒子即当前预测最优粒子, 并将此最优粒子代入真实适应度函数, 获得真实适应度, 再将此真解与历史最优解进行比较, 二者中较优的个体作为当前迭代步的最优粒子, 以该粒子标记可能最优区域, 指引下一代粒子群飞向该区域。

#### (3) 动态更新寻优经验知识库

为了克服算法对学习样本过于依赖的缺点, 提高 GP 模型在寻优过程中的预测能力, 该算法运用了动态更新寻优经验知识库的策略。对寻优经验知识库的动态更新从以下两个途径实现: ①对  $q$  代粒子中适应度最小的粒子进行了一次真实适应度评价, 不浪费此次适应度评价, 将该粒子个体信息作为新的知识源替换寻优经验知识库中最差粒子, 更新寻优经验知识库; ②当不满足精度要求进入外循环时, 在最优粒子指引下主 PSO 结构产生第  $2+p$  代粒子, 经过适应度评价, 将此代粒子信息与原寻优经验知识库的所有样本信息按照适应度由小到大顺序进行排列, 选择与原知识库数量相等的适应度较小的粒子组建新的寻优经验知识库, 该知识库中的样本更加逼近最优区域。然后, 采用 GP 模型对新知识库进行学习, 通过保持知识库的容量不变并提高知识库质量, 达到在寻优过程中不断提高 GP 模型预测能力的目的。

该算法的伪代码如图 1 所示。其框架主要由两个嵌套的 PSO 算法组成, 主 PSO 结构由外层循环构建, 循环次数为  $p$  (也是算法的迭代次数), 内层循环构建子 PSO 结构, 循环次数为  $q$ 。算法程序采用 Matlab 语言编程开发。具体实现步骤如下:

**步骤 1** 启动 Matlab 环境下的优化程序, 设置 PSO 算法的相关参数。为了保证随机解的互异性, 依照 PSO 规则随机初始化第 1 代和第  $2+p$  代粒子作为随机解。将随机解代入真实适应度函数进行适应度评价。

**步骤 2** 当步骤 1 中  $p=0$  时, 基于第 1 代和第 2 代粒子建立寻优经验知识库; 当  $p \neq 0$  时, 筛选所有学习样本中较优的  $2 \cdot NP$  个样本, 组建新的寻优经验知识库。

**步骤 3** 采用 GP 模型对知识库中样本进行学习, 建立自变量与函数值的非线性映射关系。依据 PSO 规则, 获取个体极值和全局极值。

**步骤 4** 根据 PSO 进化规则在内循环产生  $q$  代粒子, 借助学习后的 GP 模型取代真实适应度函数对粒子进行适应度预测。

**步骤 5** 选取  $q$  代预测适应度最小粒子作为当前预测最优粒子, 考虑到最优解的预测值与真实适应度函数计算解之间必然存在一定的误差, 将最优粒子代入真实适应度函数进行正向计算, 获得对应的真实适应度函数值, 并将最优粒子及其真实适应度作为新的样本, 替换寻优经验知识库中的最差样本。依据 PSO 规则, 获取个体极值和全局极值。

**步骤 6** 对当前最优解进行迭代终止条件判断,如果满足终止条件则退出计算;否则,返回步骤 2。

适应度函数。分别采用基本 PSO 算法与 PSO-GP 算法进行寻优,以函数极值小于某一精度作为迭代终止条件,取每种算法独立运行 30 次对应的函数评价次数平均值作为比较指标。

初始化

- 设置 PSO 参数。
- 随机产生  $NP$  个第 1 代粒子(自变量),以真实适应度函数作为适应度评价工具,获得每个粒子的适应度。
- 循环次数  $p=0$ 。

**While** 满足迭代终止条件

- ① 根据 PSO 进化规则产生  $NP$  个外循环中的第 2 +  $p$  代粒子,以真实适应度函数作为适应度评价工具,获得每个粒子的适应度。
- ② **If**  $p \neq 0$   
筛选所有学习样本中较优的  $2 \cdot NP$  个样本,淘汰其它较差样本,组建新的寻优经验知识库。
- End if**
- ③ 采用 GP 模型对  $2 \cdot NP$  个学习样本进行学习,建立真实适应度函数的近似模型—GP 模型。
- ④ 获取个体极值和全局极值。
- ⑤ **For**  $n=1$  to  $q$ 
  - 根据 PSO 进化规则产生  $NP$  个内循环中的第  $n$  代粒子。
  - 以 GP 模型作为适应度评价工具,预测每个粒子的适应度。
- End for**
- ⑥ 获得当前预测最优粒子,代入真实适应度函数,获得真实适应度,将此当前最优粒子及其真实适应度作为新的样本,替换寻优经验知识库中的最差样本。
- ⑦ 获取个体极值和全局极值。
- ⑧  $p=p+1$ 。

**End While**

图 1 PSO-GP 算法的伪代码

## 4 算法测试

为了验证算法的可行性,选取 8 个典型的 BenchMark 测试函数<sup>[16]</sup>,其中 5 个为单峰值函数,3 个为多峰值函数,维数均取 30。测试函数的理论最小值均为 0,此时对应的全局最优解除  $f_2$  函数是  $x=[-0.5, -0.5]^{30}$  以外,其他函数均为  $x=[0, 0]^{30}$ ,自变量的取值区间为  $[-2, 2]^{30}$ 。由于测试函数优化为无约束优化问题,采用测试函数本身作为

### 4.1 单峰值函数测试

测试函数如下:

(1) Sphere Function

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad (11)$$

(2) Step Function

$$f_2(x) = \sum_{i=1}^n (x_i + 0.5)^2 \quad (12)$$

(3) Cigar Function

$$f_3(x) = x_1^2 + 10 \sum_{i=2}^n x_i^2 \quad (13)$$

(4) Sumsquares Function

$$f_4(x) = \sum_{i=1}^n ix_i^2 \quad (14)$$

(5) Tablet Function

$$f_5(x) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2 \quad (15)$$

以精度满足  $f < 1 \times 10^{-3}$  作为终止条件,最大迭代次数  $p=20000$ 。PSO 算法参数设置为: $d=30, NP=50, c_1=c_2=2.0, V_{\max}=[-1, 1]^{30}$ 。

PSO-GP 算法参数设置为:PSO 算法的参数设置同上, $q=10$ ;GP 模型初始超参数  $\ln l=[-1, -1]^{30}, \ln \sigma_f=-1, \ln \sigma_n=\ln(1 \times 10^{-6})$ 。

PSO 算法与 PSO-GP 算法优化性能比较如表 1 所示。表中的成功率指算法寻优收敛的运行次数占总运行次数的百分数,成功率反映算法的稳健性;函数评价次数平均值反映算法的收敛速度和计算代价。从算法的寻优稳健性来看,对于测试函数中的任一函数,PSO-GP 算法与 PSO 算法的成功率均为 100%,说明 PSO-GP 算法的寻优稳健性与 PSO 算法相当。从算法的计算代价来看,PSO 算法的函数评价次数是 PSO-GP 算法的 7~13 倍,说明 PSO-GP 算法的计算代价明显低于 PSO 算法。

表 1 两种算法的单峰值函数优化性能比较

测试函数	PSO				PSO-GP				函数评价次数相差倍数
	最好值	最差值	成功率/%	函数评价次数平均值	最好值	最差值	成功率/%	函数评价次数平均值	
$f_1$	8.42E-04	9.56E-04	100	414 567	8.91E-04	9.86E-04	100	31 687	13
$f_2$	9.01E-04	9.94E-04	100	404 300	9.80E-04	9.96E-04	100	34 849	12
$f_3$	8.89E-04	9.81E-04	100	486 833	9.98E-04	9.98E-04	100	65 075	7
$f_4$	8.62E-04	9.91E-04	100	440 167	8.11E-04	8.60E-04	100	44 437	10
$f_5$	8.07E-04	9.65E-04	100	413 750	9.76E-04	9.76E-04	100	41 462	10

### 4.2 多峰值函数测试

测试函数如下:

(6) Ackley Function

$$f_6(x) = 20 + \exp(1) - 20 \exp\left(-0.2 \times \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) \quad (16)$$

(7) Griewank Function

$$f_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (17)$$

(8) Rastrigin Function

$$f_8(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (18)$$

$f_6$  函数在搜索空间中存在大量局部极小值点,大量的局部极值点使得算法难以搜索到函数最优解; $f_7$  函数是标准的多模态函数; $f_8$  函数是一个经典的多峰值复杂优化问题函数,它具有非常多的局部最小值点和局部最大值点,大量的局部极值点使得算法难以搜索到函数最优解,很容易陷入局部最优,常被用来评价优化算法的寻优效率。

由于以上多峰值函数具有大量的局部极小点和强振荡特性, $f_6, f_7$  函数以精度满足  $f < 1 \times 10^{-3}$  作为终止条件,而  $f_8$  函数收敛精度扩大为满足  $f < 30$  作为终止条件,除最大迭代次数  $p=50000$  以外,算法参数与单峰值函数测试设置相同。

表 2 两种算法的多峰值函数优化性能比较

测试函数	PSO				PSO-GP				函数评价次数相差倍数
	最好值	最差值	成功率/%	函数评价次数平均值	最好值	最差值	成功率/%	函数评价次数平均值	
$f_6$	9.83E-04	9.98E-04	100	465 283	9.60E-04	9.90E-04	100	56 133	8
$f_7$	8.58E-04	9.98E-04	100	344 600	9.87E-04	9.87E-04	100	19 175	18
$f_8$	2.93E+01	2.94E+01	100	720 417	2.92E+01	2.99E+01	100	12 035	60

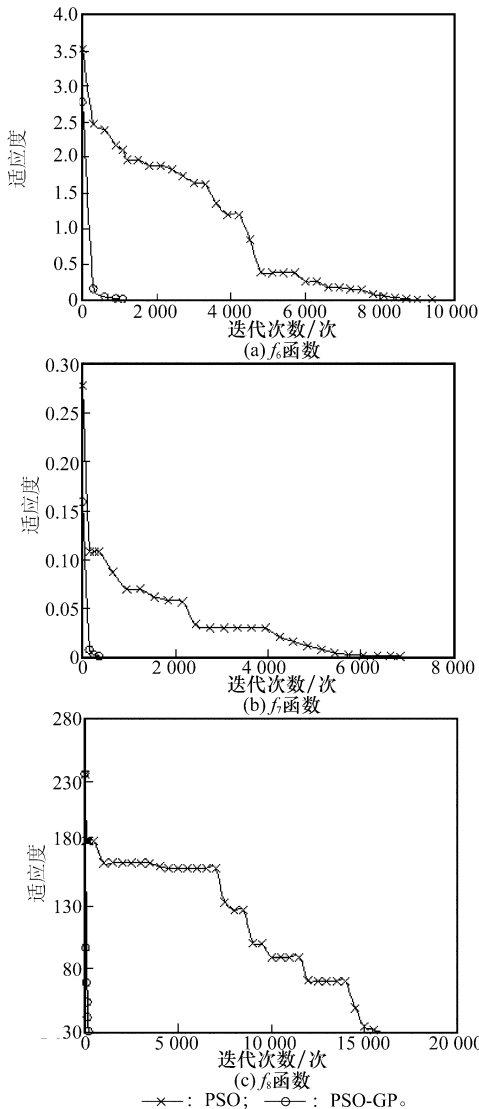


图 2 两种算法的多峰值函数适应度变化曲线

表 2 给出了两种算法的优化结果。对于函数  $f_6$ , PSO 算法的函数评价次数为 PSO-GP 算法的 8 倍;对于函数  $f_7$ , PSO 算法的函数评价次数为 PSO-GP 算法的 18 倍;对于公认的极值最难寻找的  $f_8$  函数, PSO 算法的函数评价次数为 PSO-GP 算法的 60 倍。3 个测试函数的适应度变化曲线如图 2 所示,从图中可以看出,本文算法收敛速度明显高于 PSO 算法。以上说明 PSO-GP 算法对求解高计算代价复杂多峰值函数优化问题具有较强的适用性,与 PSO 算法相比较,可以显著降低调用适应度函数的次数。

### 5 结 论

本文提出了一种基于 PSO 仿生优化算法与 GP 机器学习的协同优化算法。算法采用参数可自适应获取的 GP 模型作为适应度函数的近似模型,通过动态更新学习样本的策略实现了寻优过程中 GP 模型预测能力不断提升的目的,克服了传统的神经网络-遗传算法的过度依赖学习样本、人为设置参数多的缺点。研究结果表明,该算法是可行的,与基本 PSO 算法相比较,在获得全局最优解的前提下,可显著降低复杂函数优化过程中的适应度函数评价次数,有效地提高了计算效率,对于解决大型复杂工程结构优化设计、结构优化反分析等高计算代价全局优化问题具有良好的应用前景。

### 参考文献:

[1] Luo Y Z, Tang G J, Zhou L N. Simulated annealing based global optimization of the general bilevel nonlinear programming problems[J]. *Systems Engineering and Electronics*, 2004, 26 (12):1922 - 1926. (罗亚中,唐国金,周黎妮.一般两层非线性规划问题的模拟退火全局优化[J].系统工程与电子技术,2004, 26 (12):1922 - 1926.)

[2] Xing L N, Chen Y W, Cai H P. An intelligent genetic algorithm designed for global optimization of multi-minima functions[J]. *Applied Mathematics and Computation*, 2006,178(2):355 - 371.

[3] Zhang Y J, Shao S F, Niyongabo J. Cloud hypermutation particle swarm optimization algorithm based on cloud model[J]. *Pattern Recognition and Artificial Intelligence*, 2011, 24(1):90 - 96. (张英杰,邵岁锋, Niyongabo Julius.一种基于云模型的云变异粒子群算法[J].模式识别与人工智能,2011, 24(1):90 - 96.)

[4] Jin Y C, Olhofer M, Sendhoff B. A framework for evolutionary

- optimization with approximate fitness functions [J]. *IEEE Trans. on Evolutionary Computation*, 2002, 6(5): 481 - 494.
- [5] Ong Y S, Nair P B, Keane A J. Evolutionary optimization of computationally expensive problems via surrogate modeling[J]. *American Institute of Aeronautics and Astronautics Journal*, 2003, 41(4): 687 - 696.
- [6] Cheng J. An artificial neural network based genetic algorithm for estimating the reliability of long span suspension bridges[J]. *Finite Elements in Analysis and Design*, 2010, 46(8): 658 - 667.
- [7] Vapnik V N. *The nature of statistical learning theory* [M]. New York: Springer-Verlag, 1995.
- [8] Alex J S, Bernhard S. A tutorial on support vector regression [J]. *Statistics and Computing*, 2004, 14(3): 199 - 222.
- [9] Kennedy J, Eberhart R. Particle Swarm Optimization [C] // *Proc. of the IEEE International Conference on Neural Networks*, 1995: 1942 - 1948.
- [10] Behrang M A, Assareh E, Noghrehabadi A R, et al. New sunshine-based models for predicting global solar radiation using PSO (particle swarm optimization) technique [J]. *Energy*, 2011, 36(5): 3035 - 3049.
- [11] Kiranyaz S, Pulkkinen J, Gabbouj M. Multi-dimensional particle swarm optimization in dynamic environments [J]. *Expert Systems with Applications*, 2011, 38(3): 2212 - 2223.
- [12] Jiang Y, Liu C M, Huang C C, et al. Improved particle swarm algorithm for hydrological parameter optimization [J]. *Applied Mathematics and Computation*, 2010, 217(7): 3207 - 3215.
- [13] Seeger M. Gaussian processes for machine learning [J]. *International Journal of Neural System*, 2004, 14(2): 69 - 106.
- [14] Su G S, Song Y C, Yan L B. Application of Gaussian process machine learning to slope stability evaluation [J]. *Rock and Soil Mechanics*, 2009, 30(3): 675 - 679, 687. (苏国韶, 宋咏春, 燕柳斌. 高斯过程机器学习在边坡稳定性评价中的应用 [J]. 岩土力学, 2009, 30(3): 675 - 679, 687.)
- [15] Sofiane B B, Amine B. Gaussian process for nonstationary time series prediction [J]. *Computational Statistics & Data Analysis*, 2004, 47(4): 705 - 712.
- [16] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm [J]. *Applied Mathematics and Computation*, 2009, 214(1): 108 - 132.

### 作者简介:

张 研(1983 -), 男, 博士研究生, 主要研究方向为智能方法及其土木水利工程应用。

E-mail: yanzi22858@126.com

苏国韶(1973 -), 男, 教授, 博士, 主要研究方向为智能方法及其土木水利工程应用。

E-mail: suguoshao@163.com

燕柳斌(1955 -), 男, 教授, 博士, 主要研究方向为计算力学。

E-mail: yanliubin@126.com