



# 复现并改进高维昂贵优化算法SA-MPSO

杜可菲

**摘要：**代理辅助进化算法(SAEAs)常用于昂贵优化计算，虽然大多数现有的代理辅助进化算法(SAEAs)很少关注高维的昂贵优化，但近年有研究人员针对高维的昂贵优化问题提出了一些代理辅助进化算法(SAEAs)，其中Liu等人在2021年发表的Surrogate-Assisted Multipopulation Particle Swarm Optimizer for High-Dimensional Expensive Optimization一文中提出了一种新算法——代理辅助的多种群粒子群优化器(SA-MPSO)，并且对30维到100维的常用基准测试问题和翼型设计问题的实验结果表明，SA-MPSO优于一些最先进的方法。本次在复现该算法的基础上对其进行了改进，取得了一定的改进效果。

**关键词：**昂贵优化；代理辅助进化算法(SAEAs)；多种群遗传算法

## 1 引言

代理辅助的进化算法(SAEAs)非常适合应用于昂贵优化，然而现有的代理辅助的进化算法大多只适用于低维或中维的高维优化，本次复现的这篇工作是Liu等人于2021年提出的一种新的用于高维昂贵优化问题的代理辅助进化算法——代理辅助的多种群粒子群优化器(SA-MPSO)。根据文章的实验结果，该算法从30维到100维的常用基准测试问题上的表现优于一些当前最先进的方法。

## 2 相关工作

### 2.1 进化算法

进化算法(EAs)是一个“算法簇”，作为一种成熟的具有高鲁棒性和广泛适用性的智能优化技术，因其自组织、自适应、自学习的特性，进化算法被广泛用于工程科学中的各个领域，与传统优化方法相比，进化算法在全局优化、复杂性问题的求解及易用性方面都显示出其优越性。进化算法发展到今天，出现了许多方法，如遗传算法(GA)、遗传编程(GP)、进化策略(ES)、进化编程(EP)、粒子群算法(PSO)、蚁群算法(ACO)和膜计算等等。在对进化算法的研究中，算法设计一直是研究工作的重点，这方面的研究，始终围绕两个主题，一是对进化算法应用领域的拓展，二是提高进化算法的工作效率。前者重点放在设计和发现进化算法的搜索策略上，使其能

解决过去不能解决或不能有效解决的问题，后者则着重改进已有的算法,使其效率进一步提高。

## 2.2 代理辅助的进化算法

进化算法(EAs)由于其简单的实现和高效的，被广泛应用于解决各种优化问题，并取得了一定的成功。然而，当涉及到涉及数百甚至数千个决策变量的高维优化问题时，进化算法面临的在有限的适应度评估(FEs)下找到最优结果面临着巨大的挑战。为了克服这一障碍，一些工作将进化算法(EAs)与代理模型相结合，提出了代理辅助的进化算法(SAEAs)。在这些工作中，代理模型被用于近似复杂模型，以在优化过程中省去一些真实函数适应度评估；代理模型还被用于预先筛选候选解决方案，这样一来只有那些有前途的解决方案才会被极其耗时的复杂模型评估。

一些代理辅助的进化算法侧重于低维度问题，例如Tang等人<sup>[1]</sup>针对昂贵的黑盒问题提出了一种基于代理的PSO算法，并将该算法应用于多个工程问题；Vincenzi等人<sup>[2]</sup>提出了一种改进的SAEA，它将差分进化(DE)与二次代理和填充采样策略相结合，该方法在基准函数和工程应用上进行了性能验证。一些研究人员则专注于将代理辅助的进化算法应用于解决高维昂贵问题，例如Regis等人<sup>[3]</sup>建立了一个全局RBF代理模型来预测有希望的PSO后代，并在当前最优解的子区域进行局部搜索；Liu等人<sup>[4]</sup>利用高斯过程预测当前最佳样本产生的后代，并使用降维(DR)技术解决50个变量问题；Wang等人<sup>[5]</sup>利用代元集合建立全局模型，并利用粒子群算法寻找模型的最优值，同时寻找最不确定的解；Sun等人<sup>[6]</sup>提出了一种代理辅助的协作群优化算法，将代理辅助的PSO和RBF模型辅助的社会学习粒子群优化(SL-PSO)联合使用；Yu等人<sup>[7]</sup>提出了一种用于高维昂贵问题的代理辅助分层粒子群优化算法(SHPSO)，使用局部RBF网络来指导新样本的选择。

### 3 本文方法

#### 3.1 本文方法概述

代理辅助的多种群粒子群优化器(SA-MPSO)采用一种无参数聚类技术——亲和传播聚类(Affinity Propagation Clustering)方法生成多个子种群；并提出了一种基于粒子群优化算法的代理辅助学习策略指导各个子种群进行搜索，用以辅助种群进化；还提出了一种子种群多样性维持方案以及基于代理模型的信任域局部搜索技术，前者使得子种群之间尽量不重合，以增强算法的探索能力，后者则是将搜索的区域缩小至代理模型的最优值附近，以提高算法的开发效率。

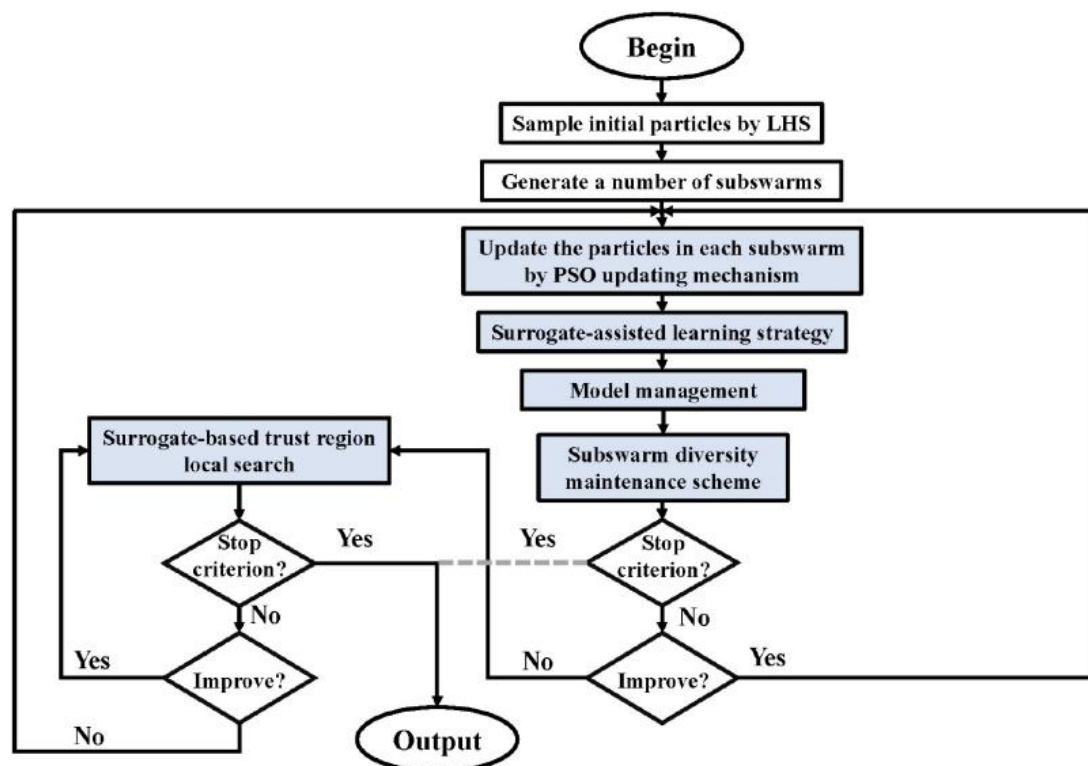


图 1 总体框架

#### 3.2 多种群遗传算法<sup>[8]</sup>

多种群遗传算法是在遗传算法的基础上经过改进并引入多种群的概念，主要有以下的改进：

(1)把单个种群改变为多个种群，每个种群都有着可控制的参数，例如交叉概率、变异概率；

(2)通过特定的操作因子来控制各种群之间的联系与协同进化，可以得出所有种群最优的进化结果；

(3)多种群的收敛条件可以根据每个种群进化的最优个体的数目来测定，各个种群中的最优个体可以增加人工选择算子来进行保留。

由于多种群遗传算法通过对各个种群不同参数的设定，例如交叉概率与变异概率的设定，可以使各个种群向着不同方向进行进化，以增强探索能力，因此多种群技术是保持种群多样性最有效的手段之一。

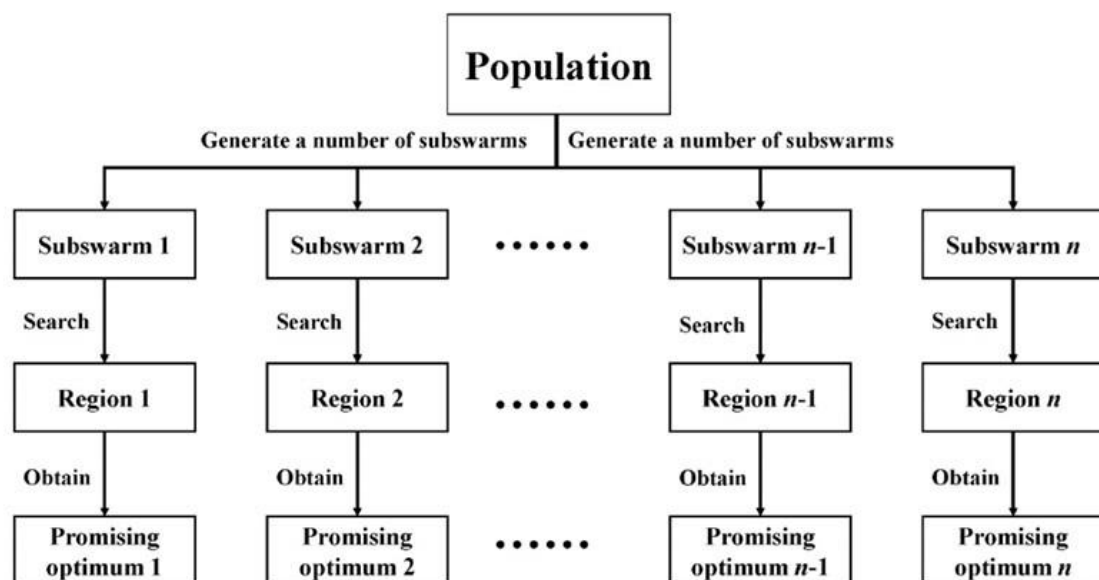


图 2 多种群遗传算法

图2是在这项工作中使用的多种群技术的框架。从图2可以看出，种群被划分为几个子种群，每个子种群同时独立地搜索自己的区域。由于每个子种群都由自己的进化策略来指导搜索，因此多种群技术可以找到多个分布在不同区域的最优解。

### 3.3 亲和传播聚类<sup>[9]</sup>

AP聚类(亲和传播聚类)于2007年提出, 特别适合高维、多类数据的快速聚类, 相比传统的聚类算法, 从聚类性能和效率方面都有大幅度的提升。

AP聚类算法是基于数据点间的"信息传递"的一种聚类算法, 把一对数据点之间的相似度作为输入, 在数据点之间交换真实有价值的信息, 直到一个最优的类代表点集合(聚类中心)和聚类逐渐形成, 此时所有的数据点到其最近的类代表点的相似度之和最大。与k-均值算法或k中心点算法不同, AP算法不需要在运行算法之前确定聚类的个数, 因为聚类中心同时也是数据集合中实际存在的点。

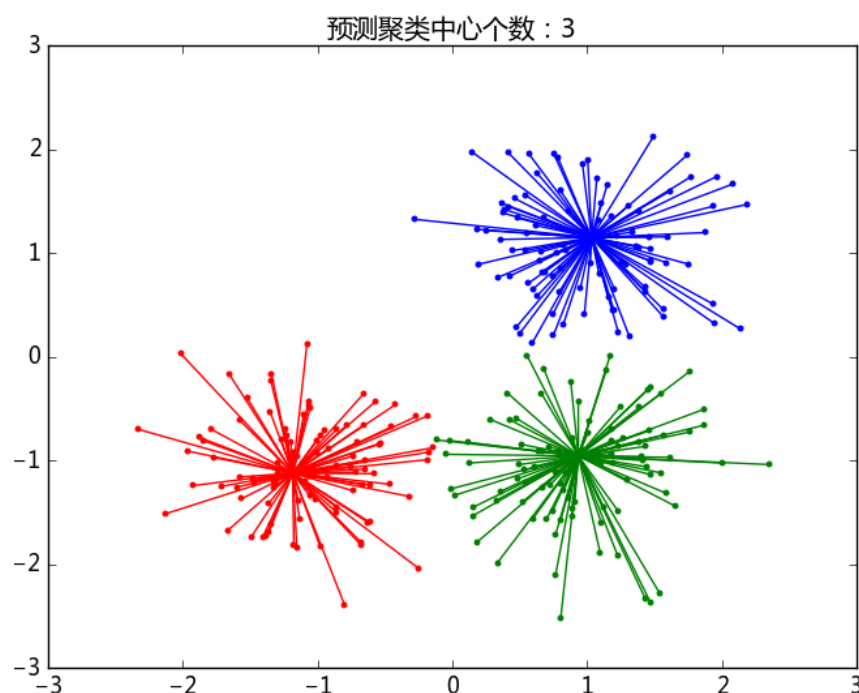


图 3 AP 聚类

如图3所示, 亲和传播集群可以从消息传递过程中产生适当的集群数量, 因此不需要为亲和传播集群预先定义子集群的数量。此外, 亲和传播聚类不消耗任何实际的FE, 所以在这项工作中采用亲和传播聚类方法来生成大量的子种群。

### 3.4 RBF神经网络<sup>[10]</sup>

RBF模型对响应特性无要求，能够较好地拟合任何种类的函数，包括非线性程度较高的函数，鲁棒性和适应性都较强，同时收敛速度较快，计算成本较低。

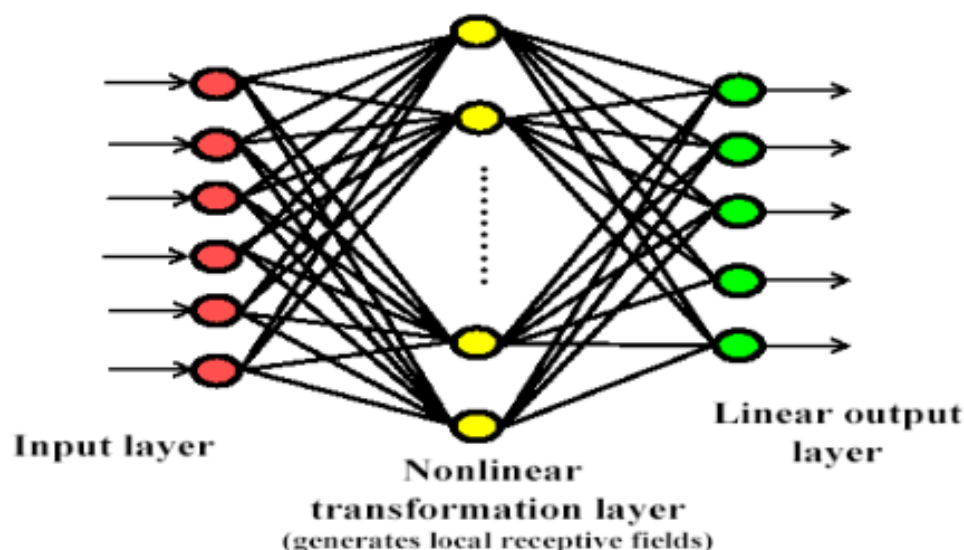


图 4 RBF 神经网络

如图4所示，RBF神经网络一共分为三层，第一层为输入层即Input Layer，由信号源节点组成；第二层为隐藏层即图中中间的黄球，隐藏层中神经元的变换函数即径向基函数是对中心点径向对称且衰减的非负线性函数，该函数是局部响应函数，所以一般要根据具体问题设置相应的隐藏层神经元个数；第三层为输出层，是对输入模式做出的响应，输出层是对线性权进行调整，采用的是线性优化策略，因而学习速度较快。

## 4 复现细节

### 4.1 引用代码

此次复现工作在原论文开源代码的基础上实现改进，使用了其定义的主函数SA\_MPSO.m、初始化函数initialize\_pop.m、全局搜索函数Global\_search.m、局部置信域搜索函数CSO\_LRA.m、亲和传播聚类函数

AP.m，该工作对SA\_MPSO.m、initialize\_pop.m、Global\_search.m、CSO\_LRA.m分别进行了改进，并对比实验结果。

## 4.2 实验环境搭建

本次复现工作实验环境如下：Windows10、MatlabR2019b。

## 4.3 创新点

本次论文复现的创新点如下：

- (1)更换代理模型的核函数，对比算法性能；
- (2)增强局部探索能力，对比算法性能；
- (3)优化初始化方法，对比算法性能；
- (4)尝试集成多个代理模型，对比算法性能。

通过从代理模型、探索能力、初始化方法三个方面对原论文进行改进，本次论文复现工作实现了目标1、2、3，由于时间和能力等因素，未能实现目标4的改进。

# 5 实验结果分析

## 5.1 更换代理模型核函数

本工作首先对代理模型(RBF)的核函数进行更换，分别将全局代理模型和局部代理模型的核函数由默认的三次函数(Cubic)更换为高斯函数(Gaussian)、线性函数(Linear)。

在5个10维至50维的基准测试问题上实验的结果表明，在更换全局代理模型的核函数对算法的改进效果并不明显，而更换局部代理模型的核函数对算法的改进效果虽然比前者要好一些，但并不明显，因此没有在100维的基准测试问题上继续进行实验。



表1 更换代理模型核函数部分实验结果

problem	D	SA-MPSO	SA-MPSO_L1(linear)	SA-MPSO_L2(gaussian)
F1	10	2.68E-01(2.76E-01)	3.38E-01(2.57E-01)	4.63E-01(2.81E-01)
	30	5.67E-01(3.17E-01)	<b>4.44E-01(4.24E-01)</b>	1.01E+00(5.94E-01)
	50	9.60E+00(4.95E+00)	1.27E+01(4.80E+00)	1.44E+01(6.37E+00)
F2	10	1.94E+01(5.89E+00)	3.37E+01(1.11E+01)	3.57E+01(2.05E+01)
	30	8.31E+01(2.89E+01)	9.60E+01(2.05E+01)	9.49E+01(3.26E+01)
	50	1.49E+02(5.35E+01)	2.02E+02(2.74E+01)	2.13E+02(6.39E+01)
F3	10	1.00E+00(7.67E-02)	1.04E+00(3.10E-02)	<b>9.85E-01(1.13E-01)</b>
	30	7.03E-01(1.31E-01)	7.04E-01(1.80E-01)	7.19E-01(1.63E-01)
	50	9.17E-01(5.08E-02)	1.01E+00(1.82E-01)	9.64E-01(6.68E-02)
F4	10	4.31E+00(8.49E-01)	5.55E+00(8.11E-01)	5.17E+00(1.43E+00)
	30	1.67E+00(2.12E-01)	1.69E+00(2.68E-01)	1.78E+00(3.73E-01)
	50	3.92E+00(3.26E-01)	<b>3.83E+00(8.43E-01)</b>	<b>3.71E+00(1.61E-01)</b>
F5	10	6.76E+01(1.08E+01)	<b>6.73E+01(9.97E+00)</b>	<b>6.31E+01(2.19E+01)</b>
	30	1.73E+02(2.13E+01)	1.82E+02(2.20E+01)	1.83E+02(2.45E+01)
	50	3.92E+02(3.80E+01)	<b>3.59E+02(3.65E+01)</b>	<b>3.86E+02(4.78E+01)</b>
+/-/~		4/2/9		4/2/9

## 5.2 增强局部探索能力

接下来，本工作采取了两个策略以增强局部探索能力，其一是在每个下一代种群中增加一个距离当前种群最优解最远的点，其二是在每个下一代种群中增加一个距离当前种群样本点集距离最大的点。

在5个10维至50维的基准测试问题上实验的结果表明，两个增强局部探索能力的策略都取得了不错的改进效果，因此在100维的基准测试问题上继续进行实验。然而实验结果表明，面对100维的基准测试问题，增强局部探索能力对算法性能并没有提升效果，反而还会降低算法性能。

表2 增强局部探索能力部分实验结果

problem	D	SA-MPSO	SA-MPSO_dis1	SA-MPSO_dis2
F1	10	2.68E-01(2.76E-01)	<b>2.00E-01(2.31E-01)</b>	2.97E-01(2.54E-01)
	30	5.67E-01(3.17E-01)	6.35E-01(3.45E-01)	6.93E-01(7.21E-01)
	50	9.60E+00(4.95E+00)	<b>6.20E+00(3.78E+00)</b>	<b>6.72E-01(4.38E-01)</b>
	100	7.01E+01(3.25E+01)	1.11E+03(4.09E+02)	1.11E+03(4.09E+02)
F2	10	1.94E+01(5.89E+00)	3.28E+01(1.52E+01)	2.53E+01(9.14E+00)
	30	8.31E+01(2.89E+01)	9.29E+01(4.66E+01)	9.13E+01(1.97E+01)
	50	1.49E+02(5.35E+01)	1.98E+02(2.97E+01)	<b>8.65E+01(2.32E+01)</b>
	100	2.78E+02(7.56E+01)	1.17E+03(1.49E+02)	1.17E+03(1.49E+02)

<b>F3</b>	10	1.00E+00(7.67E-02)	1.04E+00(3.06E-02)	1.02E+00(1.42E-02)
	30	7.03E-01(1.31E-01)	<b>5.51E-01(1.15E-01)</b>	<b>6.76E-01(2.46E-01)</b>
	50	9.17E-01(5.08E-02)	9.30E-01(8.43E-02)	<b>5.97E-01(1.71E-01)</b>
	100	4.26E+00(4.15E+00)	3.02E+01(7.09E+00)	3.02E+01(7.09E+00)
<b>F4</b>	10	4.31E+00(8.49E-01)	<b>4.16E+00(6.47E-01)</b>	4.88E+00(9.29E-01)
	30	1.67E+00(2.12E-01)	<b>1.64E+00(4.37E-01)</b>	<b>1.53E+00(4.72E-01)</b>
	50	3.92E+00(3.26E-01)	<b>3.41E+00(3.44E-01)</b>	<b>1.20E+00(6.12E-01)</b>
	100	8.25E+00(2.45E-01)	1.55E+01(1.06E+00)	1.55E+01(1.06E+00)
<b>F5</b>	10	6.76E+01(1.08E+01)	<b>5.83E+01(2.08E+01)</b>	<b>6.56E+01(6.82E+00)</b>
	30	1.73E+02(2.13E+01)	<b>1.70E+02(4.52E+01)</b>	1.90E+02(2.82E+01)
	50	3.92E+02(3.80E+01)	<b>3.68E+02(2.73E+01)</b>	<b>1.92E+02(2.50E+01)</b>
	100	8.69E+02(4.32E+01)	9.68E+02(3.97E+01)	9.68E+02(3.97E+01)
<b>+/-/≈</b>		<b>9/4/7</b>		<b>8/4/8</b>

### 5.3 优化初始化方法

最后，本工作从两个角度对初始化方法进行了优化，其一是改变初始化种群的样本数量(4D-1、5D-1、6D-1、7D-1，其中D为问题的维度)，其二是优化初始化函数lhsdesign()。在5个10维至50维的基准测试问题上实验的结果表明，改变初始化种群的样本数量对算法的改进效果并不明显，而优化初始化函数lhsdesign()取得了不错的改进效果，因此在100维的基准测试问题上继续进行实验。

表3 优化初始化方法部分实验结果

problem	D	SA-MPSO	SA-MPSO_N4D-1	SA-MPSO_lhs
<b>F1</b>	10	2.68E-01(2.76E-01)	4.63E-01(2.81E-01)	3.35E-01(2.24E-01)
	30	5.67E-01(3.17E-01)	<b>4.32E-01(2.08E-01)</b>	<b>3.92E-01(1.88E-01)</b>
	50	9.60E+00(4.95E+00)	<b>8.14E+00(1.80E+00)</b>	<b>7.84E+00(2.92E+00)</b>
<b>F2</b>	10	1.94E+01(5.89E+00)	3.57E+01(2.05E+01)	2.58E+01(8.26E+00)
	30	8.31E+01(2.89E+01)	<b>7.92E+01(2.62E+01)</b>	<b>5.65E+01(2.31E+01)</b>
	50	1.49E+02(5.35E+01)	2.34E+02(3.56E+01)	<b>1.45E+02(3.51E+01)</b>
<b>F3</b>	10	1.00E+00(7.67E-02)	<b>9.85E-01(1.13E-01)</b>	<b>9.84E-01(9.71E-02)</b>
	30	7.03E-01(1.31E-01)	7.78E-01(2.01E-01)	<b>6.14E-01(1.11E-01)</b>
	50	9.17E-01(5.08E-02)	9.59E-01(5.63E-02)	9.54E-01(2.88E-02)
<b>F4</b>	10	4.31E+00(8.49E-01)	5.17E+00(1.43E+00)	5.26E+00(1.59E+00)
	30	1.67E+00(2.12E-01)	1.69E+00(6.33E-01)	1.74E+00(4.79E-01)
	50	3.92E+00(3.26E-01)	4.91E+00(7.32E-01)	<b>3.52E+00(3.47E-01)</b>
<b>F5</b>	10	6.76E+01(1.08E+01)	<b>6.31E+01(2.19E+01)</b>	<b>8.40E+01(1.27E+01)</b>
	30	1.73E+02(2.13E+01)	2.06E+02(4.10E+01)	1.77E+02(4.95E+01)
	50	3.92E+02(3.80E+01)	4.20E+02(3.18E+01)	4.01E+02(2.53E+01)
<b>+/-/≈</b>		<b>5/0/10</b>		<b>8/0/7</b>

在5个10维至100维的基准测试问题上实验的结果表明，优化初始化函数lhsdesign()这一策略在原算法的基础上取得了不错的改进效果。于是，本工作将优化初始化函数lhsdesign()与上一节中增强局部探索能力的两个策略分别进行了更进一步的结合，在5个10维至100维的基准测试问题上实验的结果表明，优化初始化函数lhsdesign()与在每个下一代种群中增加一个距离当前种群最优解最远的点相结合的改进效果比优化初始化函数lhsdesign()与在每个下一代种群中增加一个距离当前种群样本点集距离最大的点相结合的改进效果稍好一些，但它们的效果都不如仅优化初始化函数lhsdesign()。

表4 优化初始化函数实验结果

problem	D	SA-MPSO	SA-MPSO_lhs	SA-MPSO_dis1lhs	SA-MPSO_dis2lhs
F1	10	2.68E-01(2.76E-01)	3.35E-01(2.24E-01)	<b>1.55E-01(1.11E-01)</b>	<b>1.10E-01(3.99E-02)</b>
	30	5.67E-01(3.17E-01)	<b>3.92E-01(1.88E-01)</b>	<b>1.25E-01(8.74E-02)</b>	6.52E-01(3.69E-01)
	50	9.60E+00(4.95E+00)	<b>7.84E+00(2.92E+00)</b>	<b>8.01E+00(1.98E+00)</b>	<b>8.34E+00(3.75E+00)</b>
	100	7.01E+01(3.25E+01)	8.46E+01(3.16E+01)	1.44E+03(5.32E+02)	1.44E+03(5.32E+02)
F2	10	1.94E+01(5.89E+00)	2.58E+01(8.26E+00)	3.41E+01(1.63E+01)	<b>1.93E+01(8.88E+00)</b>
	30	8.31E+01(2.89E+01)	<b>5.65E+01(2.31E+01)</b>	1.09E+02(2.78E+01)	<b>7.85E+01(3.04E+01)</b>
	50	1.49E+02(5.35E+01)	<b>1.45E+02(3.51E+01)</b>	2.36E+02(3.14E+01)	2.46E+02(4.98E+01)
	100	2.78E+02(7.56E+01)	<b>2.33E+02(3.41E+01)</b>	1.41E+03(4.17E+02)	1.41E+03(4.17E+02)
F3	10	1.00E+00(7.67E-02)	<b>9.84E-01(9.71E-02)</b>	<b>9.86E-01(3.24E-02)</b>	1.02E+00(6.66E-03)
	30	7.03E-01(1.31E-01)	<b>6.14E-01(1.11E-01)</b>	7.03E-01(8.51E-02)	7.05E-01(1.64E-01)
	50	9.17E-01(5.08E-02)	9.54E-01(2.88E-02)	9.52E-01(2.62E-02)	9.95E-01(1.43E-02)
	100	4.26E+00(4.15E+00)	<b>2.87E+00(1.59E+00)</b>	3.65E+01(1.52E+01)	3.65E+01(1.52E+01)
F4	10	4.31E+00(8.49E-01)	5.26E+00(1.59E+00)	4.65E+00(1.58E+00)	<b>3.65E+00(6.12E-01)</b>
	30	1.67E+00(2.12E-01)	1.74E+00(4.79E-01)	<b>1.34E+00(2.72E-01)</b>	1.97E+00(9.38E-01)
	50	3.92E+00(3.26E-01)	<b>3.52E+00(3.47E-01)</b>	5.00E+00(4.61E-01)	5.07E+00(6.66E-01)
	100	8.25E+00(2.45E-01)	<b>7.79E+00(4.52E-01)</b>	1.61E+01(5.07E-01)	1.61E+01(5.07E-01)
F5	10	6.76E+01(1.08E+01)	<b>8.40E+01(1.27E+01)</b>	<b>6.09E+01(1.83E+01)</b>	6.19E+01(1.79E+01)
	30	1.73E+02(2.13E+01)	1.77E+02(4.95E+01)	2.08E+02(2.87E+01)	1.98E+02(2.08E+01)
	50	3.92E+02(3.80E+01)	4.01E+02(2.53E+01)	4.35E+02(2.38E+01)	4.48E+02(2.26E+01)
	100	8.69E+02(4.32E+01)	<b>8.66E+02(3.63E+01)</b>	1.02E+03(6.44E+01)	1.02E+03(6.44E+01)
+/-/~		12/0/8		6/6/8	5/5/10

## 6 总结与展望

本次复现工作复现的论文提出了一种代理辅助的多种群粒子群优化器(SA-MPSO)，该算法突破了代理辅助优化技术的局限，可求解维数为100维的高维昂贵优化问题；在5个10维至100维的基准测试问题(Ellipsoid, Rosenbrock, Griewank, Ackley, Rastrigin)上的实验结果表明，与另外两种使用代理辅助进化算法相比，SA-MPSO算法具有更好的效果。

本次复现工作从代理模型、探索能力、初始化方法三个方面对原算法实现了改进，但由于时间和精力有限，未能实现代理模型集成方面的改进；并且在实验中还体现出——训练代理模型的计算时间随着问题维数的增加而增加这一问题，代理模型集成或改进代理模型管理策略或许会对提高算法效率有所帮助。

## 参考文献

- [1] Y. Tang, J. Chen, and J. Wei, “A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions,” *Eng. Optim.*, vol. 45, no. 5, pp. 557–576, 2013.
- [2] L. Vincenzi and P. Gambarelli, “A proper infill sampling strategy for improving the speed performance of a surrogate-assisted evolutionary algorithm,” *Comput. Struct.*, vol. 178, pp. 58–70, Jan. 2017.
- [3] R. G. Regis, “Particle swarm with radial basis function surrogates for expensive black-box optimization,” *J. Comput. Sci.*, vol. 5, no. 1, pp. 12–23, 2014.
- [4] B. Liu, Q. Zhang, and G. G. E. Gielen, “A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.
- [5] H. Wang, Y. Jin, and J. Doherty, “Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems,” *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.
- [6] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, “Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.
- [7] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, “Surrogate-assisted hierarchical particle swarm optimization,” *Inf. Sci.*, vols. 454–455, pp. 59–72, Jul. 2018.
- [8] Ma H, Shen S, Yu M, et al. Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey[J]. *Swarm and evolutionary computation*, 2019, 44: 365-387.
- [9] Frey B J, Dueck D. Clustering by passing messages between data points[J]. *science*, 2007, 315(5814): 972-976.
- [10] Er M J, Wu S, Lu J, et al. Face recognition with radial basis function (RBF) neural networks[J]. *IEEE transactions on neural networks*, 2002, 13(3): 697-710.