

面向多目标优化的多样性代理辅助进化算法*

孙哲人, 黄玉划, 陈志远

(南京航空航天大学 计算机科学与技术学院, 江苏 南京 211106)

通讯作者: 孙哲人, E-mail: szheren2k@163.com



摘要: 代理辅助进化算法(SAEA)是目前解决昂贵优化问题的一种有效途径.提出一种基于多样性的代理辅助进化算法(DSAEA)来解决昂贵多目标优化问题.DSAEA 采用 Kriging 模型近似每个目标来代替原目标函数进行评估,加速了进化算法的优化过程.其引入参考向量把问题分解为多个子问题,根据解与参考向量之间的角度大小建立它们的相关性,然后计算出最小相关解集.在此基础上,候选解生成算子和选择算子会趋向于保留多样性的解.另外,训练集 A 在每次迭代后会进行更新,根据多样性删除价值不大的样本以减少建模时间.实验部分对 DSAEA 与目前流行的代理辅助进化算法在大规模 2 目标和 3 目标优化问题上进行对比实验.每个算法在不同的测试问题上分别独立运行 30 次,并计算和统计反向迭代距离(IGD)、超体积(HV)和运行时间,最后使用秩和检验分析实验结果.结果表明:DSAEA 在多数实验测试问题上表现更好,因此具有有效性和可行性.

关键词: 代理模型;进化算法;多目标优化;昂贵问题;参考向量;模型管理;Kriging

中图法分类号: TP18

中文引用格式: 孙哲人,黄玉划,陈志远.面向多目标优化的多样性代理辅助进化算法.软件学报,2021,32(12):3814-3828.
<http://www.jos.org.cn/1000-9825/6109.htm>

英文引用格式: Sun ZR, Huang YH, Chen ZY. Diversity based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization problem. Ruan Jian Xue Bao/Journal of Software, 2021,32(12):3814-3828 (in Chinese). <http://www.jos.org.cn/1000-9825/6109.htm>

Diversity Based Surrogate-assisted Evolutionary Algorithm for Expensive Multi-objective Optimization Problem

SUN Zhe-Ren, HUANG Yu-Hua, CHEN Zhi-Yuan

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: The surrogate-assisted evolutionary algorithm (SAEA) is an effective way to solve expensive problems. This study proposed a diversity-based surrogate-assisted evolutionary algorithm (DSAEA) to solve the expensive multi-objective optimization problem. DSAEA approximates each objective with the Kriging model to replace the original objective function evaluation, accelerating the optimization process of the evolutionary algorithm. It decomposes the problem into several subproblems with the reference vectors. The correlation between the solution and the reference vector is established according to the angle between them. Then the minimum correlative solution set is computed. Based on it, the candidate producing operator and the selection operator tend to preserve the solutions of diversity. In addition, as the training set, Archive A is updated after each iteration, deleting the little value samples according to diversity to reduce the modeling time. In the experiment section, large scale 2- and 3-objective comparative experiments for DSAEA and several current popular SAEAs were done. Each algorithm on different test problems ran 30 times independently, and the inverted generational distance (IGD), hypervolume (HV), and running time were calculated and collected. At last, rank sum test was used to analyze the experimental results. The results show that DSAEA performs better on the most experimental test problems, therefore, it is effective and feasible.

* 基金项目: 江苏省科技支撑计划(BE2013879)

Foundation item: Science & Technology Support Plan of Jiangsu Province (BE2013879)

收稿时间: 2020-02-21; 修改时间: 2020-04-13, 2020-06-07; 采用时间: 2020-06-27

Key words: surrogate; evolutionary algorithm; multi-objective optimization; expensive problem; reference vector; model management; Kriging

多目标优化问题(multi-objective optimization problem,简称 MOP)^[1]是指包含两个或两个以上目标的优化问题,其目标之间往往相互冲突且难以相互比较.由于能够较好地权衡多目标优化问题中的多个目标,进化算法逐渐成为了很多领域中解决多目标优化问题的一种较为流行的工具^[2],如经济、金融、工程^[3-5].

在过去的 30 多年里,专家学者对多目标进化算法进行了大量的研究,提出了许多先进的算法,大体可以分为基于支配、基于分解和基于指标这 3 种^[6].基于支配的算法主要有 Zitzler 等人提出的 SPEA2^[7]和 Deb 等人提出的 NSGA-II^[8],基于分解的算法主要有 Deb 等人提出的 NSGA-III^[9]和 Zhang 等人提出的 MOEA/D^[10],基于指标的算法主要有 Zitzler 等人提出的 IBEA^[11]和 Beume 等人提出的 SMS-EMOA^[12].进化算法在解决多目标优化问题时,需要对解进行大量的真实评估,即,使用原始的目标函数评估解.然而,现实中存在很多问题需要计算机仿真、真实实验或数据驱动的方式来评估,因此其评估过程往往需要花费大量时间、人力、物力和财力,这类问题被称为昂贵问题^[13],如创伤系统优化^[14]等.对于昂贵问题,由于受到评估时间的限制,进化算法难以在短时间内给出较好的结果.而代理辅助进化算法可以通过廉价的模型评估代替昂贵的真实评估来加速进化算法,大大减少评估过程的代价,同时能够保证较好的优化结果.

目前,常用的代理模型主要有多项式响应模型、Kriging 模型、神经网络模型、径向基函数网络模型和支持向量回归模型,但在选择代理模型方面缺乏理论指导^[2].一般来说,在使用插值法的模型中,首选的是 Kriging 模型,因为 Kriging 模型在评估解的目标值的同时,可以给出解的不确定性,而不需要额外使用其他方法来计算不确定性.因为有此特性,Kriging 模型是代理辅助进化算法一个较为流行的选择,如 Knowles 提出的 ParEGO^[15],Ponweiser 等人提出的 SMS-EGO^[16],Zhang 等人提出的 MOEA/D-EGO^[17]和 Chugh 等人提出的 K-RVEA^[18]都使用了 Kriging 模型.

以上提到的算法是此领域较为流行且具有代表性的算法,它们选择解的时候更多关注解在模型评估下的收敛性.但很多情况下,在模型评估下的好解是伪好解,即在真实评估下并不是好解.对于 Kriging 模型来说,训练样本的分布对模型评估的准确度有很大的影响,训练样本在需要评估的解四周分布越广泛,则模型评估会相对越准确^[19].对于多目标优化问题,广泛分布在帕累托前沿(Pareto front,简称 PF)上的解,也会较为广泛分布在帕累托最优集合(Pareto set,简称 PS)上^[20].因此,如果能够在真实 PF 附近得到分布较为广泛的样本,Kriging 模型就可以较好地近似真实 PF 附近的局部空间,从而增加找到在真实 PF 附近的解的可能.这就需要更多地考虑多样性,而不只是收敛性.因此,本文提出了基于多样性的代理辅助进化算法(DSAEA)来解决昂贵多目标优化问题.

1 背景知识

1.1 多目标优化问题

多目标优化问题的一般形式如下:

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\}^T \\ \text{subject to } \mathbf{x} &\in R^d, \end{aligned}$$

其中, $\mathbf{F}(\mathbf{x})=\{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\}^T$ 为 m 个目标组成的目标向量, $\mathbf{x}=\{x_1, \dots, x_d\}^T$ 为 d 个决策变量组成的决策向量, R^d 为 d 维决策空间, R^m 为 m 维目标空间, \mathbf{F} 表示从决策空间映射到目标空间.当且仅当 $\forall i=\{1, \dots, m\}, f_i(\mathbf{x}^k) \leq f_i(\mathbf{x}^l)$ 和 $\exists j=\{1, \dots, m\}, f_j(\mathbf{x}^k) < f_j(\mathbf{x}^l)$, 则称 \mathbf{x}^k 支配 \mathbf{x}^l . 如果一个解在解集中没有解能够支配它,则称为非支配解;反之,则称为支配解.所有非支配解的决策向量构成的集合叫做帕累托最优集合 PS,对应的目标向量构成的集合叫做帕累托前沿 PF.

1.2 代理辅助进化算法

代理辅助进化算法中的评估方式分为两种.

- 一种为真实评估,即用原昂贵问题进行评估,正如引言部分提到的,用原昂贵问题进行评估一般需要仿真计算、真实实验、数据驱动的方式,需要巨大的代价.因此,真实评估需要被限制使用,不然优化时间会是难以接受的;
- 另一种为模型评估,即用训练的代理模型进行评估,这种方式相比真实评估只需要非常少的消耗.

本文所涉及到的代理辅助进化算法流程如图 1 所示,更多有关代理辅助进化算法的内容可以查阅文献 [2,13].图 1 中,初始化是在决策空间内随机采样一定数量的解,并进行真实评估.这些解作为训练集,用于建立代理模型.候选解生成算子是在模型评估下迭代优化产生候选解.选择算子是在候选解中选择部分候选解进行真实评估,并加入到训练集以更新代理模型.此时,若未达到真实评估上限,则继续迭代优化;否则输出真实评估过的解,即种群中的非支配解.其中,与一般进化算法的选择过程不同的是:由于受到真实评估次数的限制,同时,模型评估与真实评估的函数值存在一定的误差,代理辅助进化算法需要选择更具代表性的解^[17].即:在现有种群中对候选解的收敛性与多样性进行权衡,而不是直接按照适应度值进行选择.另外,新候选解的产生并不是把当前种群直接作为父代解,同时,真实评估的解较少,因此在代理辅助进化算法中,种群并没有明显的迭代更新过程,而是用于保存所有真实评估过的解.

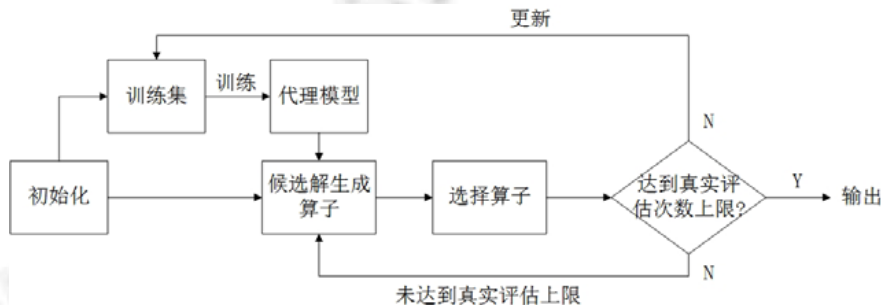


Fig.1 Flow diagram of the SAEA
图 1 代理辅助进化算法流程示意图

1.3 相关工作

目前有较多研究针对解决多目标昂贵问题的代理辅助进化算法,这里简要介绍较为流行的算法.

- Knowles 提出的 ParEGO^[15]拓展了 EGO^[19],并首次把 EGO 用于解决多目标优化问题.ParEGO 把问题均匀划分成多个子问题,即把多目标优化问题转化为单目标优化问题,每次随机选取一个子问题,把预期改善(EI)^[19]作为选择策略来选择进行真实评估的解.此外,ParEGO 还限制了训练集的大小以减少建模时间;
- Ponweiser 等人提出的 SMS-EGO^[16]使用下置信界(LCB)^[21]来划分解的支配关系,把得到的解分成非 ϵ 支配解、 ϵ 支配解、支配解:对于非 ϵ 支配解,SMS-EGO 计算其 S -metric 作为适应度;对于 ϵ 支配解和支配解,SMS-EGO 分配一个惩罚值作为适应度,距离非支配解越远惩罚越大.基于此选择策略来选择进行真实评估的解.虽然 SMS-EGO 表现不错,但是由于需要大量计算 S -metric,所以它的运行时间特别长;
- Zhang 等人提出的 MOEA/D-EGO^[17]先对进化算法得到解聚类,然后把 EI 作为选择策略,在每一簇中选择 一个解进行真实评估.此外,它采用模糊聚类的方法建立代理模型,在利用所有真实评估的解的同时减少了建模时间.与 ParEGO 相比,MOEA/D-EGO 运行速度相对更快,因为它每次批量选择解,而不只是选择单个解;
- Chugh 等人提出的 K-RVEA^[18]更多的贡献在模型管理策略上.它把本次得到的候选解与上一次得到的候选解分别分配到一组参考向量上:如果不活跃的参考向量在数量上的变化少于一定的数量,就采用不确定性作为选解依据;否则采用角度惩罚距离作为选解依据.此外,K-RVEA 在更新模型的时候会筛选解来限制训练集的大小,从而减少建模时间;

- Pan 等人提出的 CSEA^[22]采用前馈神经网络(FNNs)模型作为代理模型,其作用不同于以上几个算法,这里代理模型是作为分类器使用.迭代开始时,它把真实评估的解按比例分成训练集和测试集,并计算出分类器在测试集上非支配解和支配解的错误率.在选择过程中,分类器把进化算法得到的解分成非支配解和支配解两类;同时,根据分类器在测试集上的错误率与阈值对比来确定是选择非支配解还是选择支配解进行真实评估.

2 基于多样性的代理辅助进化算法(DSAEA)

DSAEA 算法框架如算法 1 所示.

- 步骤 1、步骤 2 是初始化部分,该部分先在目标空间内均匀生成参考向量,再随机采样初始解,并在真实评估后加入种群 P 和训练集 A ;
- 步骤 3~步骤 10 是算法的主要迭代过程,在真实评估次数达到上限后,算法便会终止并输出最终解,即种群 P 中的非支配解.其中:步骤 4 是用训练集 A 来训练代理模型;步骤 5 是候选解生成算子,它会在模型评估下迭代优化候选解,如算法 3;步骤 6 是选择算子,它是根据选择策略选择 μ 个候选解进行真实评估,如算法 4、算法 5;步骤 8 是训练集更新,如算法 6.训练集 A 会在每一次迭代后进行更新,以保证下一次迭代时训练出的模型有较好的精度.

算法 1. DSAEA 算法框架.

输入: N :参考向量的数量;

μ :选出解的数量;

FE_{max} :最大真实评估次数;

输出:种群 P 中的非支配解.

1. 均匀生成 N 个参考向量,记为 W ;
2. 采样 $11d-1$ 个初始解,真实评估后加入 P 和 A ; //初始化
3. **while** $FE_{max}>0$ **do**
4. 使用 A 训练模型 $model$;
5. $X=Candidates(A,W,model)$; //候选解生成算子,算法 3
6. $X=Selection(P,W,X,\mu)$; //选择算子,算法 4、算法 5
7. 真实评估 X 并添加到 P ;
8. $A=Update(P,W,X)$; //训练集更新,算法 6
9. $FE_{max}=FE_{max}-|X|$;
10. **end**

DSAEA 主要由初始化、代理模型构建、最小相关解集计算、候选解生成算子、选择算子、训练集更新这 6 个部分组成,下面分别进行详细说明.

2.1 初始化

初始化部分主要包括均匀生成参考向量和生成初始解.

- (1) 均匀生成参考向量:利用正则单形点阵设计方法^[23,24]在单位超平面上生成一组均匀分布的参考向量.

对于每个参考向量 $w^i = \{w_1^i, \dots, w_m^i\}$, $w^i \in W, i = 1, \dots, N$, 从下面集合中均匀取值:

$$\left\{ \frac{0}{H}, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H} \right\},$$

且满足:

$$\sum_{j=1}^m w_j^i = 1,$$

其中, H 用于控制参考向量生成的间隔, m 为目标数量.生成参考向量的数量为 $N = C_{H+m-1}^{m-1}$;

- (2) 生成初始解:利用拉丁超立方随机采样(LHS)^[25]方法对决策空间均匀采样初始解.如文献[15,17,18]所述,较为合适的随机采样次数为 $11d-1$,其中, d 为决策变量的数量.

2.2 代理模型构建

本文采用 Kriging 模型作为代理模型,即高斯过程来拟合目标函数,分别对每一个目标建模.一方面,Kriging 模型在评估目标值的同时可以给出该目标上的不确定性,不需要使用额外的方法评估解的不确定性;另一方面,Kriging 模型因其有效性而较为流行,很多研究都使用它,方便对比.下面对训练 Kriging 模型做一个简要说明,更多细节见文献[19,26].

为了建立一个廉价的 Kriging 模型,我们需要两点假设.

- 第一,假设对于任意 \mathbf{x} ,其目标函数值 y 都满足:

$$y = \mu + \varepsilon \quad (1)$$

其中, μ 是回归模型的预测值, $\varepsilon \sim N(0, \sigma^2)$;

- 第二,假设对于任意两个 \mathbf{x}^i 和 \mathbf{x}^j ,它们的高斯随机过程的协方差与它们之间的距离相关,表示如下:

$$c[\varepsilon(\mathbf{x}^i), \varepsilon(\mathbf{x}^j)] = \sigma^2 R(\mathbf{x}^i, \mathbf{x}^j) \quad (2)$$

$$R(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\sum_{k=1}^d \theta_k |x_k^i - x_k^j|^{p_k}\right) \quad (3)$$

其中, θ_k 和 p_k 为超参数, $p_k \in [1, 2]$.

在以上假设的基础上,对于 N_l 个训练样本,我们可以最大似然,公式(4),来估计参数 μ 和 σ^2 :

$$\frac{1}{(2\pi\sigma^2)^{N_l/2} \sqrt{\det(\mathbf{R})}} \exp\left[-\frac{(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right] \quad (4)$$

其中, \mathbf{y} 为训练样本的目标函数值, $\mathbf{1}$ 为长度为 N_l 的单位列向量, $\det(\mathbf{R})$ 为相关矩阵 \mathbf{R} 的行列式. \mathbf{R} 表示如下:

$$\mathbf{R} = \begin{bmatrix} R(\mathbf{x}^1, \mathbf{x}^2) & \cdots & R(\mathbf{x}^1, \mathbf{x}^{N_l}) \\ \vdots & \ddots & \vdots \\ R(\mathbf{x}^{N_l}, \mathbf{x}^1) & \cdots & R(\mathbf{x}^{N_l}, \mathbf{x}^{N_l}) \end{bmatrix} \quad (5)$$

为了最大化公式(4),可以得到 μ 和 σ^2 :

$$\mu = (\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \mathbf{R}^{-1} \mathbf{y} \quad (6)$$

$$\sigma^2 = \frac{1}{N_l} (\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu) \quad (7)$$

至此,对于输入 $\hat{\mathbf{x}}$,可以估计其近似目标值 \hat{y} 和方差 \hat{s}^2 ,即不确定性:

$$\hat{y} = \mu + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu) \quad (8)$$

$$\hat{s}^2 = \sigma^2 [1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + (\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1})^{-1} (1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2] \quad (9)$$

其中, \mathbf{r}^T 是样本与输入 $\bar{\mathbf{x}}$ 的相关矩阵:

$$\mathbf{r}^T = [R(\bar{\mathbf{x}}, \mathbf{x}^1), \dots, R(\bar{\mathbf{x}}, \mathbf{x}^{N_l})]^T \quad (10)$$

2.3 最小相关解集计算

因为候选解生成算子、选择算子和训练集更新都需要计算最小相关解集,算法 2 用于算法 3、算法 5 和算法 6 中,所以这里优先对其进行介绍.

把原问题分解为多个单目标子问题来求解,可以较好地权衡解的收敛性和多样性^[8,9].这里的计算最小相关解集也是基于分解的思想,它引入参考向量把原问题分解为多个单目标子问题,再求得每一个子问题最小相关解.所有参考向量的最小相关解组成的集合即为最小相关解集.其中的每一个最小相关解对应一个子问题的当前最优解,因而最小相关解集的大小与解的多样性有关.最小相关解集越大,则越多的子问题存在当前最优解,即这个集合的多样性越好;反之亦然.

DSAEA 是按照解与参考向量的角度来判定它们的相关性的,并在此基础上得到参考向量的最小相关解.

解 x 与参考向量 w 的角度 θ 的计算如下:

$$\theta = \arccos\left(\frac{\hat{y} \cdot w}{\|\hat{y}\| \cdot \|w\|}\right) \quad (11)$$

其中, \hat{y} 是目标函数值, $w \in W$ 为参考向量, $\|\cdot\|$ 是向量的模.

最小相关解集的具体算法见算法 2, 其中, $d(x, z^*)$ 为解 x 到理想点 z^* 的距离. 对于解集 X 中的每个解 x , 其相关参考向量是与 x 相差角度最小的参考向量. 对于一个参考向量 w , 其相关解为所有相关参考向量为 w 的解, 可能没有也可能很多个. 参考向量 w 的最小相关解为 w 的相关解中距离理想点 z^* 最近的一个. 所有参考向量的最小相关解组成的集合即为解集 X 的最小相关解集. 其中, $z^* = (\min(f_1), \dots, \min(f_m))$, $\min(f_i)$ 为种群 P 中的解在第 i 个目标上的最小值.

算法 2. 最小相关解集 MCS.

输入: X : 解集;

W : 参考向量集;

输出: C : 每个参考向量 w 的最小相关解的集合.

1. **for** 每个 $x \in X$ **do**
2. 找到与 x 角度相差最小参考向量 $w \in W$;
3. **if** $isempty(C_w) \parallel d(x, z^*) < d(C_w, z^*)$ **then**
4. 把 x 设置为 w 的最小相关解 $C_w = x$;
5. **end**
6. **end**

2.4 候选解生成算子

候选解生成算子是在模型评估下迭代优化候选解, 以得到在现有模型下表现较为优秀的候选解, 具体见算法 3. 为了充分利用现有条件, 候选解生成算子使用训练集 A 作为初始解, 而不是重新随机采样. 每次生成子代解之前, 如果 $|X|$ 小于 $|A|$, 就从训练集 A 中随机选择 $|A| - |X|$ 个解加入 X . 这样可以使 X 中解的数量不会太少, 以提高算法的探索能力和解的多样性, 从而更大机会搜索到更多有希望的候选解. 在得到初始父代解后, 候选解生成算子使用模拟二进制交叉(simulated binary crossover)^[27]和多项式突变(polynomial mutation)^[28]算子生成子代解, 并使用 Kriging 模型评估其函数值. 然后计算父代解和子代解并集的最小相关解集, 把最小相关解集中的解作为下一代父代解继续迭代优化. 当达到最大迭代次数时, 输出候选解集.

算法 3. 候选解生成算子 Candidates.

输入: A : 训练集 A ;

W : 参考向量集;

$model$: Kriging 模型;

w_{max} : 最大评估次数;

输出: X : 候选解集.

1. $X = A$;
2. **while** $w_{max} > 0$ **do**
3. **if** $|X| < |A|$ **then**
4. 从 A 中随机选择 $|A| - |X|$ 个解添加到 X ;
5. **end**
6. 生成子代解 off , 并使用 $model$ 评估 off ;
7. $X = X \cup off$;
8. $X = MCS(X)$; // 计算 X 的最小相关解集, 见算法 2
9. $w_{max} = w_{max} - |off|$;

10. end

2.5 选择算子

选择算子是在候选解集中选择部分解进行真实评估,具体见算法 4.其先是找到每个解 $\mathbf{x} \in X$ 的邻居参考向量,然后根据是否存在活跃的邻居参考向量,把 \mathbf{x} 加入 X_a 或 X_{ia} .这里的活跃与否是看该参考向量在种群 P 中是否有相关解,即该子问题上是否存在已经真实评估的解.接下来,算法对 X_a 和 X_{ia} 进行筛选,删除一些希望渺茫的解,见算法 5.然后,对解集 X 使用 Kmeans 方法聚类,划分成 μ 个簇,并在每个簇中选择一个与理想点 \mathbf{z}^* 距离最小的解.与文献[17,18]中所说明的一致,在本文实验中, μ 设置为 5.

算法 4. 选择算子 Selection.

输入: P :种群 P ;

W :参考向量集;

X :候选解集;

μ :选出解的数量;

输出: X :候选解集.

1. **for** 每个 $\mathbf{x} \in X$ **do**
2. 设置与 \mathbf{x} 相差角度小于 θ_{th} 的参考向量为 \mathbf{x} 的邻居参考向量;
3. **if** \mathbf{x} 存在活跃的邻居参考向量 **then**
4. $X_a = X_a \cup \mathbf{x}$;
5. **else**
6. $X_{ia} = X_{ia} \cup \mathbf{x}$;
7. **end**
8. **end**
9. 对 X_a 和 X_{ia} 进行筛选; //算法 5;
10. $X = X_a \cup X_{ia}$;
11. 对 X 使用 Kmeans 方法聚类,划分成 μ 个簇;
12. 对于每个簇,保留距离 \mathbf{z}^* 最小的解;

算法 5. 筛选.

1. $X_r = MCS(P)$; //计算 P 的最小相关解集,见算法 2
2. 记 $d(X_r, \mathbf{z}^*)$ 的最大值为 d_{th} ;
3. 删除 X_{ia} 中 $d(\mathbf{x}, \mathbf{z}^*) > d_{th}$ 的 \mathbf{x} ;
4. **for** 每个 $\mathbf{x} \in X_a$ **do**
5. 把 X_r 中对应 \mathbf{x} 的邻居参考向量的解加入 X_r' ;
6. 升序排序 $d_a = d(X_r', \mathbf{z}^*)$;
7. $d_{th} = d_a^{[0.25 \times |d_a|]}$;
8. 若 $d(\mathbf{x}, \mathbf{z}^*) > d_{th}$,则在 X_a 中删除 \mathbf{x} ;
9. **end**

由于模型评估具有一定的误差,真实评估的函数值相对模型评估的函数值会发生一定程度的偏移.这里,我们根据解的不确定性来确定其邻居参考向量.算法 4 中的阈值 θ_{th} 的计算如公式(12):

$$\theta_{th} = \arccos \left(\frac{\hat{\mathbf{y}} \cdot \hat{\mathbf{y}}'}{\|\hat{\mathbf{y}}\| \cdot \|\hat{\mathbf{y}}'\|} \right) \quad (12)$$

$$\hat{\mathbf{y}}' = \hat{\mathbf{y}} + p \times (-\hat{s}_1, \hat{s}_2, \hat{s}_3, \dots, \hat{s}_m) \quad (13)$$

其中, $\hat{\mathbf{y}}'$ 是在置信范围内距离 $\hat{\mathbf{y}}$ 角度较大的一个解.

公式(13)中的 p 是一个系数,本文实验取 $p=2$. $(-\hat{s}_1, \hat{s}_2, \hat{s}_3, \dots, \hat{s}_m)$ 是所有目标上不确定性组成的向量,其中 f_1 对应的不确定性 \hat{s}_1 取负.

对 X_a 或 X_{ia} 的筛选见算法 5,其主要目的是根据现有真实评估过的解来删除一些希望渺茫的解,从而提高最终选出的解的质量.

其中, X_{ia} 为不存在活跃的邻居参考向量的解,即 $\mathbf{x} \in X_{ia}$ 周围不存在真实评估的解.一方面,考虑到 \mathbf{x} 周围没有真实评估的解,需要尽量保留 \mathbf{x} 以探索其附近;另一方面,保留过差的 \mathbf{x} 对模型和种群都没有任何改善,因此,这里用种群 P 中所有的解来计算一个大体范围来筛选 X_{ia} .这里把 $d(X_r, \mathbf{z}^*)$ 的最大值作为阈值 d_{th} 筛选解 X_{ia} ,删除 $d(\mathbf{x}, \mathbf{z}^*) > d_{th}$ 的 \mathbf{x} .其中, $d(X_r, \mathbf{z}^*)$ 为集合 X_r 中每一个解到理想点 \mathbf{z}^* 的距离.

X_a 为存在活跃的邻居参考向量的解,即 $\mathbf{x} \in X_a$ 周围存在真实评估的解.这里按照其邻居参考向量的最小相关解排序后的下四分位筛选 X_a ,阈值为

$$d_{th} = d_a^{\lceil 0.25 \times |d_a| \rceil}$$

下四分位是描述整体样本前一半的平均指标,即保留与邻居参考向量上的最小相关解相比较好的 \mathbf{x} ,这样对 \mathbf{x} 周围有较大的改善.

2.6 训练集更新

训练集 A 在每次迭代后都会进行更新,以保存多样性较好的解,以提高模型拟合的精度;同时,删除价值不大的解以减少建模时间.本文实验设置训练集 A 大小下限为 $11d-1$,更新过程见算法 6.

为了能够利用选择算子选出的解,步骤 1 从解集 X 中选出一个 $d(\mathbf{x}, \mathbf{z}^*)$ 最小的解加入训练集 A .步骤 2 计算所有参考向量在种群 P 中的最小相关解集,并解加入训练集 A .这样就可以得到多样性较好的训练集,并且随着算法的迭代,样本会越来越靠近真实 PF,从而帮助 Kriging 模型较好地拟合真实 PF 附近的空间.步骤 4、步骤 5 从相关解最少的参考向量的相关解中选出最好的解加入训练集 A .这是为了补充训练样本,防止训练样本过少而导致模型准确度下降.

算法 6. 训练集更新 Update.

输入: P : 种群 P ;

W : 参考向量集;

X : 解集;

输出: A : 训练集 A .

1. 选出 $d(\mathbf{x}, \mathbf{z}^*)$ 最小的 $\mathbf{x} \in X$ 加入 A ;
2. $X_r = MCS(P)$; // 计算 P 的最小相关解集,见算法 2;
3. 把 X_r 加入 A ;
4. $P' = P - A$;
5. **while** $|A| < 11d - 1$ **do**
6. 找到在 P' 上相关解最少的 w ;
7. 把 w 的最小的相关解加入 A ;
8. 在 P' 中删除此解;
9. **end**

3 实验

由于使用原昂贵问题评估需要花费大量的时间,若采用真实的昂贵问题,那么对比实验将会花费巨量的时间,实验的时间花费将会令人难以接受.因此,我们限制测试问题的真实评估次数来模拟解决昂贵多目标优化问题的场景,即测试问题只会被有限次地用于评估解.

本文实验通过在大规模 2 目标和 3 目标优化问题上的对比实验来证明 DSAEA 的有效性.目前提出的基于 Kriging 模型且性能较好的代理辅助进化算法有 ParEGO, MOEA/D-EGO, K-RVEA, SMS-EGO.其中,由于 SMS-

EGO 需要计算 S -metric 来选解,所以运行速度非常慢,甚至需要几个小时才能运行一次.所以,我们使用另外几个算法作为对比算法.

3.1 测试问题集

实验选用 ZDT^[29]和 DTLZ^[30]测试问题作为基准测试问题.其中,选用 ZDT1-4 和 ZDT6 作为 2 目标基准测试问题,选用 DTLZ1-7 作为 3 目标基准测试问题.

3.2 度量指标

实验选用反向迭代距离(inverted generational distance,简称 IGD)^[9]和超体积(hypervolume,简称 HV)^[31]作为度量指标.

(1) IGD 计算公式如下:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|},$$

其中, P 是对 PF 均匀采样后得到的目标向量集合, P^* 是算法计算得到的目标向量集合, $d(v, P)$ 是一个目标向量 $v \in P^*$ 到 P 中最近的向量的距离.本文实验在 PF 上均匀取 10 000 个点作为参考点来计算 IGD.

(2) HV 计算公式如下:

$$HV = volume\left(\bigcup_{v \in P^*} v\right),$$

其中, P^* 是算法计算得到的目标向量集合, P' 是 P^* 中非支配解的集合, $volume$ 是计算向量到参考点的超体积.本文实验取点 $1.1 \times (\max(f_1), \dots, \max(f_m)), f_1, \dots, f_m \in PF$ 作为参考点来计算 HV.

3.3 参数设置

(1) 一般设置

- 决策变量数量:ZDT 设置为 12,DTLZ 设置为 10;
- 目标变量数量:ZDT 设置为 2,DTLZ 设置为 3;
- 最大真实评估次数:ZDT 设置为 200 次,DTLZ 设置为 300 次;
- 初始采样数:利用 LHS 方法采样 $11d-1$ 个解,ZDT 问题采样 131 个解,DTLZ 问题采样 109 个解;
- 交叉算子:模拟二进制交叉(simulated binary crossover)^[27],设置概率为 1.0,分布指数为 20;
- 变异算子:多项式突变(polynomial mutation)^[28],设置概率为 $1/d$,分布指数为 20;
- 候选解生成算子的最大评估次数: $20 \times (11d-1)$ 次模型评估;
- 参考向量数量:2 目标 300 个($H=299$),3 目标 595 个($H=33$).
- 独立运行次数:每个算法对每个测试问题独立运行 30 次.

(2) 其他设置

DSAEA,MOEA/D-EGO 和 K-RVEA 中每次迭代选出的解的数量设置为 5,即 $\mu=5$.其他未说明的参数设置与其论文中的一致.

3.4 实验结果

IGD 结果见表 1,HV 结果见表 2,算法运行时间如表 3.数字表示该指标的平均值,括号内的数字表示该指标的标准差,加粗表示在此测试问题上该项平均值为最优值.另外,我们使用秩和检验在显著性水平为 0.05 下对 30 次独立运行结果进行分析.“+”表示此算法在此测试问题上与 DSAEA 相比在该指标上表现更好,“-”表示此算法在此测试问题上与 DSAEA 相比在该指标上表现更差,“ \approx ”表示此算法在此测试问题上与 DSAEA 相比在该指标上的表现区别不大.每个表格的最后一行还给出了在该指标上的+/-/ \approx 结果的汇总.由于部分测试问题没有得到较好的收敛,甚至距离 PF 非常远,如 ZDT4,ZDT6,DTLZ1,DTLZ3,DTLZ6.在没有很好的收敛的情况下,对比多样性没有很大的意义,所以这里不讨论以上几个测试问题的 HV 指标.

Table 1 Results of IGD

表 1 反向迭代距离结果

Problem	DSAEA	K-RVEA	MOEA/D-EGO	ParEGO
ZDT1	6.2887e-2(2.73e-2)	8.2035e-1(1.15e-1)-	1.7097e+0(3.74e+0)-	4.7243e-1(2.08e-1)-
ZDT2	1.1052e-1(1.07e-1)	7.4677e-1(1.37e-1)-	1.3118e+0(1.55e+0)-	5.9465e-1(1.43e-1)-
ZDT3	1.9822e-1(1.88e-1)	8.1794e-1(1.59e-1)-	1.4819e+0(1.73e+0)-	5.6781e-1(1.61e-1)-
ZDT4	6.3493e+1(1.78e+1)	7.9403e+1(2.22e+1)-	1.0688e+2(1.28e+1)-	8.5326e+1(1.37e+1)-
ZDT6	3.2883e+0(8.31e-1)	5.4347e+0(2.27e+0)-	6.2323e+0(1.99e+0)-	6.4443e+0(8.96e-1)-
DTLZ1	5.4726e+1(1.71e+1)	7.6495e+1(1.87e+1)-	8.7589e+1(1.64e+1)-	8.9775e+1(2.24e+1)-
DTLZ2	1.4977e-1(3.47e-2)	1.8420e-1(1.88e-2)-	3.3203e-1(2.55e-2)-	2.6802e-1(2.14e-2)-
DTLZ3	1.3896e+2(2.69e+1)	2.1205e+2(7.18e+1)-	2.1185e+2(4.05e+1)-	2.6622e+2(6.05e+1)-
DTLZ4	3.3670e-1(1.01e-1)	3.6212e-1(9.93e-2)≈	6.4259e-1(7.17e-2)-	4.1291e-1(1.12e-1)-
DTLZ5	2.5984e-2(6.28e-3)	7.5002e-2(1.28e-2)-	2.5429e-1(3.03e-2)-	1.7326e-1(3.06e-2)-
DTLZ6	2.7593e+0(4.34e-1)	3.8069e+0(4.72e-1)-	1.8576e+0(5.81e-1)+	4.1258e+0(4.68e-1)-
DTLZ7	1.7815e-1(3.03e-2)	1.0011e+0(1.10e-1)-	2.3411e-1(9.23e-2)-	3.3278e-1(6.28e-2)-
+/-/≈	-	0/11/1	1/11/0	0/12/0

Table 2 Results of HV

表 2 超体积结果

Problem	DSAEA	K-RVEA	MOEA/D-EGO	ParEGO
ZDT1	6.4567e-1(1.48e-2)	8.9466e-2(7.60e-2)-	3.4343e-1(2.33e-1)-	2.0311e-1(1.29e-1)-
ZDT2	3.2907e-1(8.19e-2)	4.9670e-3(1.05e-2)-	9.4252e-2(1.02e-1)-	4.0968e-2(3.81e-2)-
ZDT3	5.2323e-1(1.39e-1)	1.0257e-1(1.19e-1)-	2.4279e-1(2.35e-1)-	1.9277e-1(1.16e-1)-
DTLZ2	4.3094e-1(4.84e-2)	3.5085e-1(2.65e-2)-	1.3774e-1(4.64e-2)-	1.8637e-1(3.65e-2)-
DTLZ4	2.0978e-1(1.07e-1)	1.2883e-1(1.17e-1)-	8.5741e-3(1.49e-2)-	2.1018e-1(6.82e-2)≈
DTLZ5	1.7894e-1(6.78e-3)	1.4107e-1(1.17e-2)-	2.5012e-2(2.17e-2)-	5.9209e-2(2.35e-2)-
DTLZ7	2.2066e-1(1.15e-2)	1.5166e-1(1.28e-2)-	2.1222e-1(1.71e-2)-	1.5591e-1(2.40e-2)-
+/-/≈	-	0/7/0	0/7/0	0/6/1

Table 3 Results of running time

表 3 运行时间结果

Problem	DSAEA	K-RVEA	MOEA/D-EGO	ParEGO
ZDT1	2.7213e+1(2.24e-1)	2.6465e+1(3.81e-1)+	3.8747e+1(3.09e-1)-	5.3115e+1(4.98e-1)-
ZDT2	2.6538e+1(3.10e-1)	2.5988e+1(4.28e-1)+	3.8776e+1(3.57e-1)-	5.2682e+1(6.76e-1)-
ZDT3	2.6967e+1(3.53e-1)	2.6303e+1(3.22e-1)+	3.8663e+1(3.22e-1)-	5.3362e+1(5.72e-1)-
ZDT4	2.7191e+1(2.32e+0)	2.6788e+1(6.96e-1)≈	3.6354e+1(1.36e+0)-	5.3146e+1(7.36e-1)-
ZDT6	2.7832e+1(5.32e-1)	2.7018e+1(4.51e-1)+	3.9848e+1(8.02e-1)-	5.3760e+1(8.56e-1)-
DTLZ1	9.3734e+1(2.42e+0)	8.3035e+1(8.01e-1)+	1.5175e+2(2.18e+0)-	1.1071e+2(1.99e+0)-
DTLZ2	1.0472e+2(2.79e+0)	8.2676e+1(8.58e-1)+	1.4879e+2(4.22e+0)-	1.1163e+2(1.56e+0)-
DTLZ3	9.3374e+1(2.28e+0)	8.2940e+1(9.74e-1)+	1.5196e+2(2.16e+0)-	1.1049e+2(1.57e+0)-
DTLZ4	8.4365e+1(2.40e+0)	8.3634e+1(1.37e+0)≈	1.4511e+2(4.16e+0)-	1.0792e+2(2.04e+0)-
DTLZ5	8.4555e+1(1.11e+0)	8.2550e+1(8.31e-1)+	1.4961e+2(4.99e+0)-	1.1101e+2(1.15e+0)-
DTLZ6	1.0086e+2(3.40e+0)	8.3866e+1(9.85e-1)+	1.5024e+2(3.75e+0)-	1.1151e+2(1.08e+0)-
DTLZ7	7.9500e+1(8.09e-1)	8.0235e+1(1.00e+0)-	1.4154e+2(1.46e+0)-	1.1207e+2(1.54e+0)-
+/-/≈	-	9/1/2	0/12/0	0/12/0

图 2、图 3 给出了 DSAEA 与对比算法在 ZDT,DTLZ 问题上的收敛曲线,横坐标为真实评估次数,纵坐标为 IGD 指标。

对于 ZDT 测试问题:

- IGD 和 HV 结果表明:在实验设置一致的情况下,DSAEA 明显表现更好;
- 而收敛曲线表明:DSAEA 的收敛效果在大多数情况下要好于对比算法,只有在 ZDT1-3 问题上,DSAEA 的收敛效果在 170 次真实评估前不如 ParEGO。

由于 ZDT1-3 问题较为简单,在最初阶段,搜索到比现有种群更优的解很容易,如果算法根据收敛性选解,则收敛速度会很快.但在靠近真实 PF 的区域内,搜索到比现有种群更优的解相对不易,需要模型能够较好地拟合真实 PF 附近的区域.不同于 ParEGO,DSAEA 是基于多样性选解的,它更倾向于增加解集的多样性,使代理模型可以较好地拟合目前种群最优解附近的区域.因此,DSAEA 的收敛曲线前 30 次的收敛效果不如 ParEGO,而之后的

收敛效果要优于它。

对于 DTLZ 测试问题,IGD,HV 结果和收敛曲线表明:在实验设置一致的情况下,DSAEA 在大多数的测试问题上明显表现更好;只有在 DTLZ4 测试问题上,DSAEA 与 K-RVEA 的效果相当;在 DTLZ6 问题上,DSAEA 的表现不如 MOEA/D-EGO。

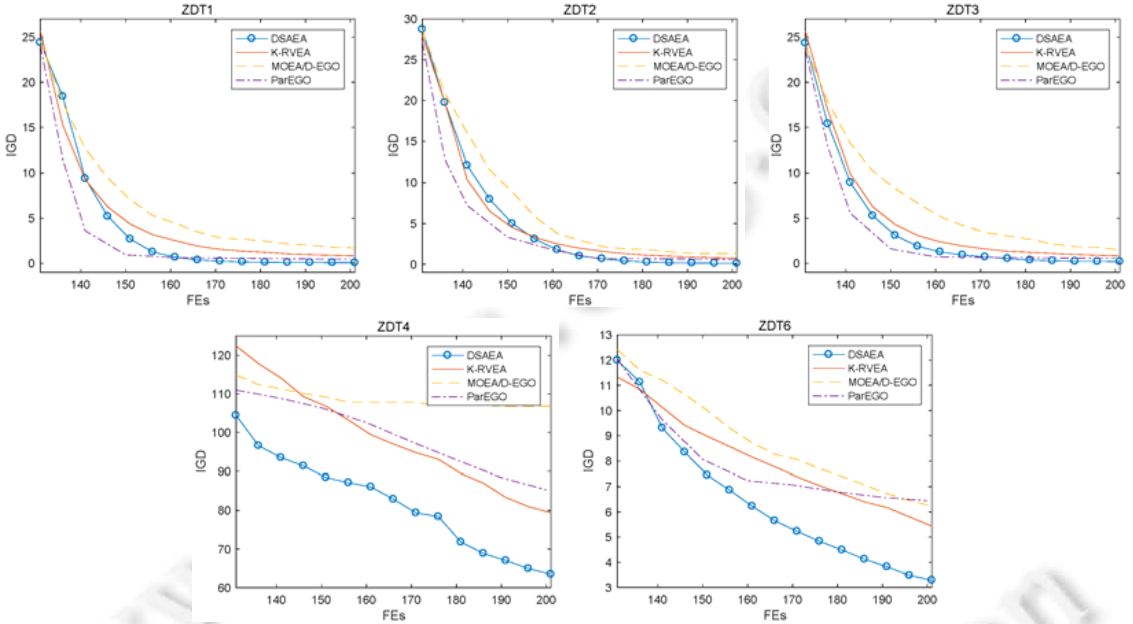


Fig.2 Convergence curve of the algorithms on the ZDT problems

图 2 算法在 ZDT 问题上的收敛曲线

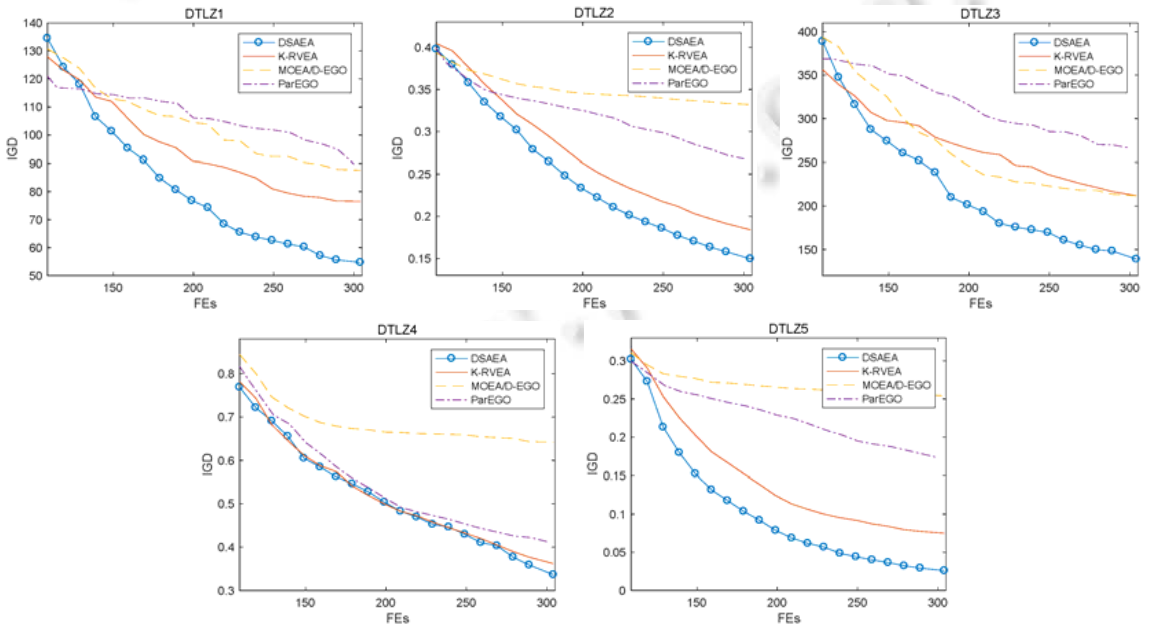


Fig.3 Convergence curve of the algorithms on the DTLZ problems

图 3 算法在 DTLZ 问题上的收敛曲线

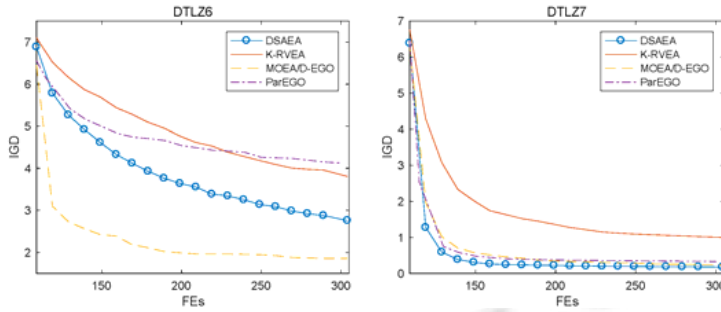


Fig.3 Convergence curve of the algorithms on the DTLZ problems (Continued)

图3 算法在 DTLZ 问题上的收敛曲线(续)

图 4 为 DSAEA 和 MOEA/D-EGO 的 30 次独立运行中,最接近 IGD 平均值的实验结果.可以看到:DSAEA 大部分的解都集中在内部 3 个参考向量附近,且有较多解向这真实 PF 靠拢;而 MOEA/D-EGO 大部分的解都集中在边界的 3 个参考向量附近,其中只有几个非常好的解,其余大部分解距离真实 PF 非常远.

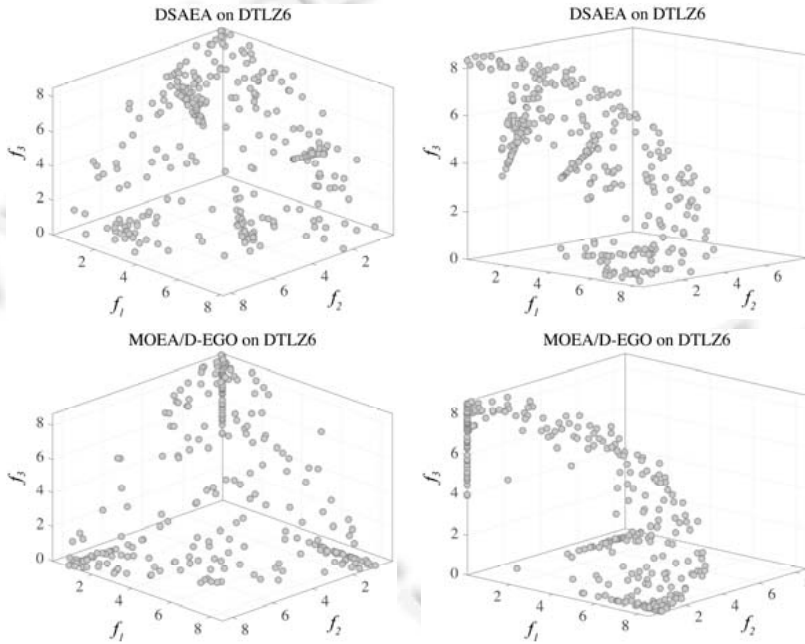


Fig.4 On the DTLZ6 problem, the function evaluated solutions of DSAEA and MOEA/D-EGO are observed from different angles

图 4 在 DTLZ6 测试问题上,从不同角度观察 DSAEA 和 MOEA/D-EGO 所有真实评估的解

在本实验中,DLTZ6 问题在决策空间内存在 4 个不连续的 PS 区域,需要算法具有更高的探索能力.DSAEA 是基于多样性的,在模型评估下,很好的解可能会被淘汰,转而保留增加多样性的解,这样可能会错过部分真实评估下很好的解.即便如此,DSAEA 的探索的大体趋势仍然要优于 MOEA/D-EGO.

从收敛性和多样性上来看,DSAEA 要比目前较为流行的算法表现更好.

在运行时间方面,相对 MOEA/D-EGO 和 ParEGO,DSAEA 运行速度更快.相对 K-RVEA,DSAEA 在大多数问题上运行速度相差不是太多,差异都在 3%以下.其中,差异较大的问题为 DTLZ1,DTLZ3,DTLZ2,DTLZ6,相比 K-RVEA,DSAEA 在前两个问题上要慢 12%左右,在后两个问题上要慢 23%左右.对于以上几个问题,因为其 PS

分布在决策空间中范围较大,很容易搜索到不同方向的解,即在不同参考向量上的解,这就使训练集 A 的大小超过 $11d-1$,增加了一定的建模时间.

从运行时间上来看,DSAEA 运行速度不亚于目前较为流行的算法.

由此可见,相比对比算法,DSAEA 在限制评估次数的情况下可以更加有效地解决实验选用的测试问题,主要分析有以下几点.

- 首先,DSAEA 引入参考向量的最小相关解来保持多样性.DSAEA 中,候选解生成算子不是按照支配关系迭代优化候选解,而是按照多样性,即会出现淘汰非支配解但保留支配解的情况.例如, x_1 支配 x_2 ,但 x_1 所在参考向量已存在最小相关解且优于 x_1 ,而 x_2 所在参考向量不存在相关解,则 x_1 会被淘汰,而 x_2 会被保留.值得一提的是:由于这里是采用模型评估,而模型评估总是存在一定的误差,所以在模型评估下的非支配解在真实评估下可能是支配解;反之亦然.此外,按照多样性保留解可以增加探索范围,避免陷入探索单一方向.因此,DSAEA 中候选解生成算子更有可能找到更多有希望的解,即,候选解生成算子迭代优化的目标是在模型评估下表现优秀的解以及未探索区域内的解;
- 其次,DSAEA 中的选择算子把解按照周围是否存在活跃参考向量进行划分筛选.一方面,DSAEA 会保留相比周围参考向量的最小相关解更好的解,以保留在真实评估下可能比现有解更好的解,从而提高当前解的收敛性;另一方面,DSAEA 会保留周围没有活跃参考向量的解,以对此未探索区域进行探索,从而提高当前解的多样性.在筛选后,DSAEA 对保留的解进行聚类,选出 μ 个代表解进行真实评估,进行批量的真实评估大大提升算法运行速度.这样,DSAEA 的选择算子对最终解的收敛性和多样性会有所帮助;
- 最后,由于测试问题决策变量数量较大,属于大规模问题,而模型精度有限,所以每次迭代后对训练集的更新是十分必要的.算法 6 先对所有真实评估过的解,即种群 P ,计算其最小相关解加入训练集.若此时的训练集大小仍小于 $11d-1$,则在剩下的解中按照参考向量上的解的密度继续选择,优先选择密度小的参考向量上的最小相关解加入训练集.这样做可以得到在目标空间内分布较为广泛且多样性较好的训练集,在此基础上,训练的模型会较好地拟合较大的范围,并且随着迭代次数的增加,训练集会更加靠近真实 PF 区域,模型也就可以更好地拟合真实 PF 附近区域.也就是说,随着迭代的进行,不仅进化算法会朝着真实 PF 附近区域搜索,而且模型也会朝着真实 PF 附近区域靠拢.

4 总 结

本文提出了基于多样性的代理辅助进化算法(DSAEA)来解决昂贵多目标优化问题.DSAEA 采用 Kriging 模型近似每个目标.候选解生成算子分配解到对应的参考向量来计算最小相关解集,以达到迭代优化候选解的目的.选择算子筛选出无活跃邻居参考向量的解,以及表现比大多数邻居参考向量的相关解更好的解.然后对筛选出的解 K -means 聚类,从每个簇中选择最好的解进行真实评估.另外,DSAEA 使用一个大小下限为 $11d-1$ 的训练集 A 作为代理模型的训练集,删除价值不大的样本以降低建模时间.实验部分选用大规模 ZDT,DTLZ 测试问题,与目前流行的 MOEA/D-EGO,K-RVEA 和 ParEGO 算法进行对比实验.结果表明:DSAEA 在大多数选用的测试问题上,要比以上几个算法表现更好.因此,本文的方法是有效可行的.

在实验中,对于大规模问题来说,决策变量的数量设置较小,但仍有部分测试问题没有很好地收敛.当决策变量的量变得更大时,搜索和建模压力会随之上升,这对进化算法的选择和建模方法是一个巨大的挑战.另外,由于随着目标数量的增加,需要更多的解来近似表示 PF,而昂贵问题中真实评估次数十分有限,所以如何用有限的点来有效地表示 PF,也是一个严峻的挑战.

References:

- [1] Gong MG, Jiao LC, Yang DD, *et al.* Research on evolutionary multi-objective optimization algorithms. *Ruan Jian Xue Bao/ Journal of Software*, 2009,20(2):271–289 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]
- [2] Jin Y, Wang H, Chugh T, *et al.* Data-driven evolutionary optimization: An overview and case studies. *IEEE Trans. on Evolutionary Computation*, 2019,23(3):442–458.
- [3] Tapia MGC, Coello CAC. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In: *Proc. of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*. Singapore: IEEE, 2007. 532–539. [doi: 10.1109/cec.2007.4424516]
- [4] Arias-Montano A, Coello CAC, Mezura-Montes E. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Trans. on Evolutionary Computation*, 2012,16(5):662–694. [doi: 10.1109/tevc.2011.2169968]
- [5] Ponsich A, Jaimes AL, Coello CAC. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Trans. on Evolutionary Computation*, 2013,17(3):321–344. [doi: 10.1109/TEVC.2012.2196800]
- [6] Zhang J, Xing L. A survey of multiobjective evolutionary algorithms. In: *Proc. of the 2017 IEEE Int'l Conf. on Computational Science and Engineering (CSE) and IEEE Int'l Conf. on Embedded and Ubiquitous Computing (EUC)*. Guangzhou: IEEE, 2017. 93–100. [doi: 10.1109/CSE-EUC.2017.27]
- [7] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. *TIK-Report*, 2001,103:1–21. [doi: 10.3929/ethz-a-004284029]
- [8] Deb K, Pratap A, Agarwal S, *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002,6(2):182–197.
- [9] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Trans. on Evolutionary Computation*, 2014,18(4):577–601.
- [10] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation*, 2008,11(6):712–731.
- [11] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Yao X, *et al.* ed. *Proc. of the Parallel Problem Solving from Nature—PPSN VIII (PPSN 2004)*. LNCS 3242, Berlin, Heidelberg: Springer-Verlag, 2004. 832–842.
- [12] Beume N, Naujoks B, Emmerich M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 2007,181(3):1653–1669.
- [13] Jin Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 2011,1(2):61–70.
- [14] Wang H, Jin Y, Janson JO. Data-driven surrogate-assisted multi-objective evolutionary optimization of a trauma system. *IEEE Trans. on Evolutionary Computation*, 2016,20(6):939–952.
- [15] Knowles J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. on Evolutionary Computation*, 2006,10(1):50–66.
- [16] Ponweiser W, Wagner T, Biermann D, *et al.* Multiobjective optimization on a limited budget of evaluations using model-assisted *s-metric* selection. In: Rudolph G, Jansen T, Beume N, Lucas S, Poloni C, eds. *Proc. of the Parallel Problem Solving from Nature—PPSN X (PPSN 2008)*. LNCS 5199, Berlin, Heidelberg: Springer-Verlag, 2008. 784–794.
- [17] Zhang Q, Liu W, Tsang E, *et al.* Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans. on Evolutionary Computation*, 2010,14(3):456–474.
- [18] Chugh T, Jin Y, Miettinen K, *et al.* A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2016,22(1):129–142.
- [19] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998,13(4):455–492.
- [20] Zhang Q, Zhou A, Jin Y. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. on Evolutionary Computation*, 2008,12(1):41–63.

- [21] Liu B, Zhang Q, Gielen GGE. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. on Evolutionary Computation*, 2014,18(2):180–192.
- [22] Pan L, He C, Tian Y, *et al.* A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2019,23(1):74–88.
- [23] Das I, Dennis JE. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 1998,8(3):631–657.
- [24] Morris, Max D. The design and analysis of computer experiments. *Journal of the American Statistical Association*, 2004,99(468):1203–1204.
- [25] Mckay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 2000,42(1):55–61.
- [26] Rasmussen CE. Gaussian processes in machine learning. In: Bousquet O, von Luxburg U, Rätsch G, eds. *Proc. of the Advanced Lectures on Machine Learning (ML 2003)*. LNCS 3176, Berlin, Heidelberg: Springer-Verlag, 2004. 64–71.
- [27] Agrawal RB, Deb K, Agrawal RB. Simulated binary crossover for continuous search space. *Complex Systems*, 2000,9(3):115–148.
- [28] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 1996, 26(4):30–45.
- [29] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2000,8(2):173–195.
- [30] Deb K, Thiele L, Laumanns M, *et al.* Scalable multi-objective optimization test problems. In: *Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002)*. IEEE, 2002. 825–830. [doi: 10.1109/CEC.2002.1007032]
- [31] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. on Evolutionary Computation*, 1999,3(4):257–271.

附中参考文献:

- [1] 公茂果,焦李成,杨咚咚,等.进化多目标优化算法研究.软件学报,2009,20(2):271–289. <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]



孙哲人(1992—),男,硕士生,CCF 专业会员,主要研究领域为智能计算.



陈志远(1996—),男,硕士生,主要研究领域为智能计算,RNA 二级机构预测.



黄玉划(1975—),男,博士,副教授,CCF 专业会员,主要研究领域为智能计算,信息安全.