加入代理辅助的DE进化算法实例应用

LJADE：一种用于光伏优化模型的动态代用代理辅助进化算法

摘要---昂贵的优化问题（EOP）广泛存在于各种重要的现实应用场景中。然而...

1.Introduction

当今，...

1. Background theory

2.1.新算法模型框架

2.2. JADE算法

2.3. 径向基函数

**Surrogate-assisted classification-alternative Evolutionary algorithm**

**(CAEA)**

引言背景

昂贵的优化问题 (EOP)是计算量大的优化问题，这意味着它们需要大量的计算资源和时间来解决。 这些问题通常需要花费昂贵的成本来计算函数/指标和约束的值，这使得使用传统优化技术找到最佳解决方案具有挑战性。EOP 在许多领域都很常见，包括工程、金融、机器学习和运筹学。 例如，在工程中，设计飞机和汽车等复杂系统可能会导致 EOP。

为了解决上述挑战，相关研究者提出使用低计算消耗的近似模型（或元启发式模型）来辅助EA进化，称为SAEAs\cite {jinEvolutionaryOptimizationApproximate}\cite {jinSurrogateassistedEvolutionaryComputation2011}\cite {liFastSurrogateassistedParticle2020 }\cite {shiSurveyFitnessApproximation2010}\cite {tenneComputationalIntelligenceExpensive2010a}。

最近，许多 SAEA 已成功应用于几个复杂的工程问题\cite {wang2019novel}\cite {wang2018global}。Tang 等人\cite {tang2013surrogate} 利用 PSO 的混合代理在弹簧和焊接梁优化中有效地实现全局优化 问题。 同时，Liu 等人引用 {liu2013gaussian} 将降维技术引入高斯过程代理辅助进化算法，以简化中维优化问题的复杂性。 Li et al.\cite {li2020surrogate} 在代理辅助多群优化算法 (SAMSO) 中使用了多个群，利用了基于教学学习的优化算法良好的全局可搜索性和 PSO 算法的快速收敛能力。 Sun et al.\cite {sun2017surrogate} 介绍了代理辅助协作群优化 (SA-COSO) 算法，该算法同时使用 CPSO（具有约束因子的 PSO）和 SL-PSO。 在 \cite {cai2019efficient} 中，作者提出了一种基于遗传算法的广义代理辅助进化算法 (GSGA)，该算法结合了利用代理模型的信任域局部搜索方法。 在 \cite {dong2020surrogate} 中，作者介绍了代理辅助灰狼优化 (SAGWO) 算法，该算法采用径向基函数作为代理模型。

图 1 显示了 SAEAs 的基本流程图。在 SAEA 中，替代模型通常基于一组具有代表性的适应度景观样本构建，这些样本可以通过随机生成或使用空间填充设计获得。 替代模型经过训练，可以根据其特征（例如设计变量或决策变量）预测新候选解决方案的适应度值。 一旦构建并验证了替代模型，它就可以通过以比原始适应度函数低得多的成本评估大量候选解决方案来指导 EA 的搜索过程。 随后，算法开始进入主循环。 在循环的每一代中，使用当前的代理模型来评估新个体的适应度值，而不是昂贵的适应度函数，并使用进化算法不断搜索问题的最优解。 同时，选择一些符合条件的个体作为填充样本，用于更新代理模型。 重复执行上述操作，直到满足终止条件。 \图[t!](topskip=0pt, botskip=0pt, midskip=0pt)[width=3.2 in]{SAEAs.png}

{SAEAs的基本流程图}

可以采用多种方法来构建代理模型\cite {jin2005comprehensive} 例如，径向基函数（RBF）模型\cite {sun2015two}，多项式响应面（PRS）\cite {wang2003adaptive}，支持向量回归（SVR） \cite {clarke2005analysis}，高斯过程（GP）\cite {chugh2016surrogate}，人工神经网络（ANN）\cite {jin2002framework}，kriging 模型\cite {cai2017metamodeling}，等等。SAEAs的优化性能也在很大程度上取决于优化器算法[12]的搜索能力。SAEAs采用不同的EAs时，SAEAs具有不同的特性，表现出不同的性能。通过集成不同的EAs的混合或集成技术[19,24,35-38]，有利于更好地利用不同的EAs的优势。Wansaseub 等人\引用 {wansaseub2017optimal} 介绍了 U 形挡板方管换热器 (USBSD-HX) 的代理辅助优化，使用多个代理模型，包括克里金模型、径向基函数插值、二次响应曲面模型 并支持目标函数逼近的向量回归。 Kudela 和 Matousek\cite {kuudela2023combining} 利用基于 Lipschitz 的代理模型以及标准径向基函数代理模型和局部搜索程序来开发基于差分进化的算法。根据一些研究\cite {diaz2017comparison}\cite {jin2001comparative} 的发现，观察到径向基函数 (RBF) 模型与其他模型相比表现出相对较好的性能。 它表明，对于那些希望在各个领域开发准确预测模型的人来说，RBF 模型可能是一个有前途的选择。

在评论论文\引用 {haftkaParallelSurrogateassistedGlobal2016} 中，首次强调了管理代理模型的重要性，以提高 SAEA 性能并防止进化算法被误导到错误的最优值。直到现在，基于个体，基于世代 , 和 mix-based是 SAEA 中广受欢迎且使用最多的技术。

在基于个人的模型管理中，挑战在于使用目标函数确定哪些个人应该重新评估。 一种有效的方法是选择具有最佳和最不确定适应值的个体\cite {jinComprehensiveSurveyFitness2005a} \cite {wangCommitteeBasedActiveLearning2017} 基于代理模型。 该策略可以帮助探索和利用搜索空间，同时还可以提高搜索过程的效率和有效性。

基于生成的方法 \cite {hskenStructureOptimizationNeural2005} 是一种优化算法，涉及评估候选解决方案群体的适应性。 在这种方法中，关键问题是确定合适的世代以使用真实目标函数评估整个种群。 仅在选定的世代中使用真实目标函数评估种群可以降低评估适应度函数的计算成本，但需要确定使用真实目标函数评估种群的频率。有两种主要方法可以确定 适当的频率参数。 一种方法是将频率参数基于替代模型的保真度。 这涉及使用替代模型的准确性度量（例如均方根误差）来确定何时使用真实目标函数评估总体。另一种方法是执行参数敏感性分析以确定每个输入参数的重要性 到健身功能。 这可以帮助识别哪些参数对适应度函数的影响最大，并且可以相应地调整使用真实目标函数评估种群的频率。

在本文中，我们采用了一种新颖的基于个人的模型管理方法。 具体来说，在通过进化算法优化生成的个体中，我们选择最好的个体和基于精英个体的中间个体。 通过比较这两个个体，我们选择最合适的个体来优化代理模型和更新数据集。

尽管有许多与 SAEA 相关的研究论文，但它们也存在一些需要解决的局限性和挑战。 例如，代理模型的准确性和泛化能力在很大程度上取决于训练样本的质量和代表性\cite {caiSurrogateguidedDifferentialEvolution2019}，训练样本的选择可能非常耗时且计算量大。 **Li 等人\cite{li2020fast} 提出了一种称为快速代理辅助 PSO (FSAPSO) 的方法，该方法利用全局 RBF 代理和新颖的填充准则来有效地解决中等规模问题。 CAL-SAPSO 由 Wang 等人提出。引用 {wang2017committee} 使用集成代理，而 Wang 等人。 \cite {wang2018offline} 提出了一种离线数据驱动的进化算法，利用选择性代理集合，以及由不同数据子集构建的一定数量的代理。** 此外，代理模型可能会在优化过程中引入额外的不确定性和偏差，这可能会影响最终解决方案的稳健性和可靠性。 需要进一步研究来解决 SAEA 的挑战和局限性，并开发更先进、更强大的替代模型，以处理更广泛的问题领域和应用。 因此，本文着重于通过提出一种称为代理辅助分类替代进化算法 (CAEA) 的新型 SAEA 来解决代价高昂的优化问题的挑战。 它包含一个具有出色自适应能力的模型辅助进化框架，并集成了 TLBO 和改进的新 JADE 算法，可以很好地平衡功能景观的开发和探索。为了评估我们提出的 CAEA 的有效性，我们对 35 个进行了综合实验 文献中广泛使用的基准问题。 实验结果表明，我们的 CAEA 在大多数具有不同维度的基准问题上优于几种最先进的竞争算法。

与现有方法相比，这项工作的主要贡献可以总结如下。

开始{itemize}

\项目 [1)]

JADE算法通过在交叉概率中加入更多的个体信息来改进自适应控制参数，该方法具有适应度值无关的特点。

\项目 [2)]

在JADE的外部存档中，引入了基于时间的重启机制，有效引导种群进化方向。

\项目 [3)]

创新采用交替机制，将TLBO和改进的JADE两种强大的算法进行混合，并使用基于生成的模型管理，使这种混合策略能够充分利用双方的优势。

\项目 [4)]

使用一种新颖的基于个体的进化策略，结合域个体和最优个体来补充和优化数据集。

\项目 [5)]

在每一代中，仅使用当前种群附近的少量样本来构建具有出色拟合能力的模型。

\结束{逐项}

本文的其余部分组织如下。第 2 节提供了相关技术的总结，其中包括两种进化算法（TLBO 和 JADE），以及使用的代理模型 RBF。 在第 3 节中，我们概述了我们提议的 CAEA 的细节。 我们在第 4 节中进行了全面的实验来评估我们提出的 SABEA 的有效性。最后，在第 5 节中，我们总结了我们的结论并讨论了未来可能的研究方向。

1. 相关工作

在本章节中，将会简要回顾（在线的）SAEA框架，JADE算法，TLBO算法，以及基本的RBF模型。

2.1 SAEA框架

代理辅助优化方法广泛使用在昂贵优化问题中，当实际函数适应度计算成本极高时，代理辅助方法能够在一定置信度的基础上大幅减少实际评估次数，从而避免较高的计算负担或巨大的时间消耗，实现高效解决问题。图1显示了一个基本的SAEA框架。SAEA的整个过程可分为数据收集、代理建模、管理和优化三个部分。....展开具体说明

SAEA的性能取决于所采用的模型的可靠性（Lim et al. 2010）。请注意，样品的质量在替代品的可靠性中起着至关重要的作用。一定数量的收集数据被用来建立寻找最优解决方案的替代品。...一些现有的方法都是如何挑选数据来进行建模的 本文中，是由两种不同的挑选数据方式综合而成，最初生成初始种群之际使用当前全部个体当作数据集，并把通过此数据集生成的模型称为全局模型；进化过程中，将局部最优数据加入初始数据集，在此数据集中挑选较优个体数据。这两种挑选数据方式结合更有助于建立更加可靠的模型，避免由于数据选择不当造成的代理模型偏见。

离线优化和在线优化的主要区别（Wang et al. 2018）是可重复数据的可用性，如图1所示。对于离线优化，最佳的解决方案在每次迭代中都没有得到真实问题的验证。相反，对于在线优化，每次迭代都要验证最佳的解决方案，以指导粒子搜索。两者的优缺点对比 本文中使用在线优化，使得整体算法框架更加灵活、具有适应性，同时，提高模型的可靠性、保证代理评估值更加接近真实值。

2.2 JADE算法

JADE作为DE的一种高效且自适应能力极强的变体，得到了学术上广泛的认可和应用。就性能而言，相比于DE，其在保留优越勘探性能的基础上，具有更好的开发能力以及开发速度。

2.3TLBO算法

1. 改进的代理模型优化算法

CAEA的主要特点是融合了两个高效的元启发进化算法，在一定评判标准下两种算法交替进行，充分结合TLBO的探索能力和JADE的开发能力。另外，对于其中的元启发算法，创新性的加入了带有时间戳的外部存档，具有删除老旧个体的能力，进一步提升种群多样性。同时，提出了一种可供替代的策略，即基于fitness-value-independent的参数控制策略，使得在面对更加复杂的黑盒模型时，我们的CAEA将具有更强的鲁棒性。

3.1 CAEA框架

在本节中，详细介绍了一种高效的代理辅助分类交替进化算法（CAEA），图1展示了所提出CAEA的总体框架。

为了保持种群多样性，避免陷入局部最优，本文提出的新算法融合了TLBO算法和改进的JADE算法，具有明确的交替规则。 它可以平衡 TLBO 出色的探索能力和 JADE 出色的开发能力来解决昂贵的优化问题。

在图1中，蓝色线条表示数据流，而黑色线条则表示算法流。Database中包含收集到的数据样本以及他们对应的真实适应度评估值。CAEA算法主要由四个部分组成：种群初始化、构建或更新RBF代理模型、通过混合TLBO和JADE使种群进化、以及为真实适应度函数选择个体。算法1给出了CAEA的伪代码。另外，CAEA包含基于代数和个体的进化控制，它在每K代的进化之后挑选出一个最合适的个体对其进行真实适应度评估，该个体数据将被存入数据库中用来优化代理模型构建过程。

种群初始化：采用拉丁超立方采样（LHS）在全局搜索空间内初始化样本点，并使用昂贵的真实适应度函数评估它们的适应度值。初始样本及其适应度值都存储在数据库中，如图1的①所示。初始样本数量与种群大小相同，具体如下。对于10、20和30维问题，种群大小根据[26]设置为5 × D。对于50和100维问题，种群大小根据[16]设置为100 + D/10，根据[16]。

种群进化：进化过程根据使用的元启发算法分成两类：TLBO、改进的JADE，随机选择一个元启发算法开始，然后进化K代。当找不到更好的候选解决方案时，这两种算法交替执行。本文优化过的JADE算法具有较强的局部搜索能力，可以集中开发搜索有前景的区域。而，TLBO算法具有较好的全局搜索能力，因为由TLBO更新的每个个体都可以由任何其他个体引导，可以专注于探索整个搜索空间。通过集成TLBO和JADE，该算法可以结合两种算法的优势，平衡探索和开发能力。值得注意，在进化的K代过程中，利用局部RBF替代模型估计新生成种群的适应度值。

RBF代理模型：使用算法2建立或更新本地RBF代理模型。代理模型的预测能力对该算法的优化性能有很大的影响。因此，建立一个高精度的替代模型是很有必要的，其不仅取决于所选择的训练数据，还取决于训练数据的大小。在本文中，我们选择当前种群的邻居来构建局部RBF代理模型，如图1中的②所示。由于邻居模型更接近种群，因此用邻居样本训练的局部RBF代理模型具有较高的精度。训练数据的大小不仅影响训练时间，而且还影响替代模型的准确性。对于10、20和30维问题，个体的邻居大小设置为5\*D。对于50和100维，个体的邻居大小设置为D，比Yu et al. [16]使用的大小要小。请注意，本地RBF代理模型将在每一代中构建或更新。

选择个体：应用一种新颖的筛选准则来选择一个有前途的个体，并用于生成新的真实适应度函数评估值，具体实现细节将展示在算法3中。EAs通常需要大量的FEs才能获得令人满意的解，但对于计算成本昂贵的问题来说这将会非常耗时。因此，在可用资源有限的情况下，为这些问题挑选出合适的个体是很重要的。如果只选择种群中表现最好的个体进行真实的FEs，种群容易陷入局部最优[18]。本文提出了一种新颖的筛选策略，该策略结合种群的最优个体和随机中间区域个体信息（随机中间区域个体是基于多个精英个体信息得出的）来帮助SAHO摆脱局部最优。具体来说，我们首先根据适应度值对种群中的个体进行排序。接着，随机生成一个整数r（不大于ps），计算排名为前r个的精英个体所处位置的平均值。接下来，我们用局部RBF模型估计随机中间区域个体的适应度值，并将其适应度值与当前种群中最佳个体的适应度值进行比较。如果随机均值个体的适合度值比最佳个体的更好，则选择随机中间区域个体进行真实FEs。否则，最佳个体被选择为真正的FEs。最后，将所选个体及其适应度值存储到数据库中，如图1中的③所示。

当实际FEs的最大数量耗尽时，此过程将停止。

将整体种群随机分为两个大小相等的子群。一个子群随着TLBO算法的进化，专注于探索具有较强全局搜索能力的TLBO。另一个子群采用DE算法具有“DE/best/1/1”突变方案的DE算法进行发展，使其具有较强的局部搜索能力。

* 使用当前邻居建立RBF局部代理模型，且在每一代都进行更新；
* 一个新的预测准则：（当资源有限是挑选出一个合适个体进行真是适应度评估是很重要的）This paper proposed a new prescreening criterion which uses the best and the top collection information of the population to help the SAHO escape from the local optimum.刚开始是根据fitness values挑选个体，后来使用随机的整数r，通过计算1~r中头部子种群的随机中值个体的代理适应度值，并将其与当前最好随即个体的适应度值进行比较。以此来挑选个体进行真实值评估，并将算出的适应度值存入DB中。
* 采用了两种新颖的重启策略：

①一般来说选择NP个顶级样本，但是当种群没有得到改善时，随机选择NP个样本重新形成进化种群。这样可以避免陷入局部最优，充分利用评估数据提供的准确信息，以避免DE和TLBO被近似模型所带来的偏差所误导，并加速搜索到有希望的区域。Xxxxxxxx

②时间戳机制；M7算法对应在变异方式中，随机个体来源于集合A中，A用于储存进化过程中形成的精英个体。当A中的精英个体，长时间没有进一步改进时，该精英个体将会从集合A中被剔除。

* 创新了一种新颖的基于过程适应度函数值DIS计算结果的方式来储存和更新Cr值，不同于...基于specifific fitness value的方式，...在无需了解过程fitness value的情况下，充分利用进化过程中自然生成的函数信息来优化Cr取值，实现不基于fitness value的自适应参数优化，从而引导种群收敛，加快收敛速度。

本节详细介绍了一种有效的替代辅助双种群交叉（SATC,Substitution-assisted two-population crossover）优化算法。

在SAHO中包含了基于世代的进化控制和基于个体的进化控制。:Both the generation- and individual-based evolution controls are incorporated in SAHO.

当找不到更好的候选解决方案时，这两种算法交替执行。使用“DE/best/1”突变策略的DE可以搜索出最好的个体。因此，它具有较强的局部搜索能力，可以集中开发搜索有前景的区域。然而，当使用DE算法来解决多模态问题时，这可能会导致过早的收敛。:These two algorithms are performed alternately when no better candidate solution can be found. The DE using ‘‘DE/best/1” mutation strategy can search around the best individual. Hence, it has strong local search ability and it can concentrate on exploitation to search the promising region. However, this may lead to premature convergence when using the DE algorithm to solve multimodal problems.

TLBO算法具有更好的全局搜索能力，因为由TLBO更新的每个个体都可以由任何其他个体指导。因此，它可以专注于探索来探索整个搜索空间。通过集成TLBO和DE，该算法可以结合两种算法的优势，平衡探索和开发能力。在进化开始时，从当前DB中挑选出顶部ps最佳数据样本，形成SAHO的进化种群，如图1的②所示。这种方法被称为最优（顶级）重启策略[16]。:the TLBO algorithm has better global search ability as each individual updated

by the TLBO can be guided by any other individuals. Thus, it can focus on exploration to explore the entire search space. By

integrating both TLBO and DE, the proposed algorithm can combine the advantage of two algorithms to balance the explo

ration and exploitation capabilities. At the beginning of evolution, the top ps best data samples are picked out from the cur

rent DB to form the evolutionary population of SAHO as shown in ② of Fig. 1. This method is called optimal (top-ranked)

restart strategy [16].

CUT收敛图像:

\begin{figure\*}[!htbp]

\centering

\subfigure[$f\_1$]{

\label{Fig.sub.1}

\includegraphics[width=0.35\textwidth]{f1.png}}

\subfigure[$f\_2$]{

\label{Fig.sub.2}

\includegraphics[width=0.35\textwidth]{f2.png}}

\subfigure[$f\_3$]{

\label{Fig.sub.3}

\includegraphics[width=0.35\textwidth]{f3.png}}

\subfigure[$f\_4$]{

\label{Fig.sub.4}

\includegraphics[width=0.35\textwidth]{f4.png}}

\subfigure[$f\_5$]{

\label{Fig.sub.5}

\includegraphics[width=0.35\textwidth]{f5.png}}

\subfigure[$f\_6$]{

\label{Fig.sub.6}

\includegraphics[width=0.35\textwidth]{f6.png}}

\subfigure[$f\_7$]{

\label{Fig.sub.7}

\includegraphics[width=0.35\textwidth]{f7.png}}

\subfigure[$f\_8$]{

\label{Fig.sub.8}

\includegraphics[width=0.35\textwidth]{f8.png}}

\caption{Here presents the convergence speed comparison by employing the median value of 51 runs obtained by each algorithm on $30D$ optimization. There are total 28 comparison figures and the first 8 figures are presented here. }

\end{figure\*}

\begin{figure\*}[!htbp]

\centering

\subfigure[$f\_9$]{

\label{Fig.sub.9}

\includegraphics[width=0.35\textwidth]{f9.png}}

\subfigure[$f\_{10}$]{

\label{Fig.sub.10}

\includegraphics[width=0.35\textwidth]{f10.png}}

\subfigure[$f\_{11}$]{

\label{Fig.sub.11}

\includegraphics[width=0.35\textwidth]{f11.png}}

\subfigure[$f\_{12}$]{

\label{Fig.sub.12}

\includegraphics[width=0.35\textwidth]{f12.png}}

\subfigure[$f\_{13}$]{

\label{Fig.sub.13}

\includegraphics[width=0.35\textwidth]{f13.png}}

\subfigure[$f\_{14}$]{

\label{Fig.sub.14}

\includegraphics[width=0.35\textwidth]{f14.png}}

\subfigure[$f\_{15}$]{

\label{Fig.sub.15}

\includegraphics[width=0.35\textwidth]{f15.png}}

\subfigure[$f\_{16}$]{

\label{Fig.sub.16}

\includegraphics[width=0.35\textwidth]{f16.png}}

\caption{As a continued part from Fig.1, comparisons on benchmarks $f\_9$-$f\_{16}$.}

\end{figure\*}

\begin{figure\*}[!htbp]

\centering

\subfigure[$f\_{17}$]{

\label{Fig.sub.17}

\includegraphics[width=0.35\textwidth]{f17.png}}

\subfigure[$f\_{18}$]{

\label{Fig.sub.18}

\includegraphics[width=0.35\textwidth]{f18.png}}

\subfigure[$f\_{19}$]{

\label{Fig.sub.19}

\includegraphics[width=0.35\textwidth]{f19.png}}

\subfigure[$f\_{20}$]{

\label{Fig.sub.20}

\includegraphics[width=0.35\textwidth]{f20.png}}

\subfigure[$f\_{21}$]{

\label{Fig.sub.21}

\includegraphics[width=0.35\textwidth]{f21.png}}

\subfigure[$f\_{22}$]{

\label{Fig.sub.22}

\includegraphics[width=0.35\textwidth]{f22.png}}

\subfigure[$f\_{23}$]{

\label{Fig.sub.23}

\includegraphics[width=0.35\textwidth]{f23.png}}

\subfigure[$f\_{24}$]{

\label{Fig.sub.24}

\includegraphics[width=0.35\textwidth]{f24.png}}

\caption{As a continued part from Fig.2, comparisons on benchmarks $f\_{17}$-$f\_{24}$.}

\end{figure\*}

\begin{figure\*}[!htbp]

\centering

\subfigure[$f\_{25}$]{

\label{Fig.sub.25}

\includegraphics[width=0.35\textwidth]{f25.png}}

\subfigure[$f\_{26}$]{

\label{Fig.sub.26}

\includegraphics[width=0.35\textwidth]{f26.png}}

\subfigure[$f\_{27}$]{

\label{Fig.sub.27}

\includegraphics[width=0.35\textwidth]{f27.png}}

\subfigure[$f\_{28}$]{

\label{Fig.sub.28}

\includegraphics[width=0.35\textwidth]{f28.png}}

\caption{As a continued part from Fig.3, the last comparisons are figured out here for analysis.}

\end{figure\*}