题目

Multisurrogate-Assisted Ant Colony Optimization for Expensive Optimization Problems With Continuous and Categorical Variables

摘要

在昂贵优化问题（EOPs）的研究领域中，如何处理混合变量问题一直是一个挑战。本文聚焦于EOPs中的一个特殊类别——具有连续和分类变量的EOPs（EOPCCVs），并提出了一种新型的多代理辅助蚁群优化算法（MiSACO）。MiSACO通过结合径向基函数（RBF）和最小二乘提升树（LSBT）作为代理模型，设计了三种选择算子来从蚁群算法生成的后代解中筛选出有前景的解。为了克服传统RBF在处理混合变量时的局限性，我们引入了一种基于频率定义的距离计算公式，该公式能够更精细地反映类变量之间的差异，并为优化算法提供更丰富的信息，从而增强了算法在搜索空间中的探索能力。

关键词：昂贵优化；混合变量

1 引言

扩展优化问题是指具有耗时目标的优化问题功能和/或约束。EOPs可分为三类：1）仅包含连续变量的连续EOPs; 2）仅包含离散变量的组合EOPs;3）混合变量EOPs ，其中既包含连续变量，也包含离散变量。此外，根据离散变量的类型，混合变量EOPs可以分为不同的类型，如连续和整数变量EOPs ，连续和二元变量EOPs ，以及连续和分类变量EOPs（EOPCCV）。本文主要关注EOPCCV。

许多现实世界的应用程序可以建模为EOPCCV 。汽车侧车身的轻量化和耐撞性设计可以作为例子，如图1所示。通常，汽车的侧车身由许多部件组成，如B柱和侧门防撞梁。各部件的结构和材料对汽车的质量和耐撞性都有很大的影响。在设计每个零件的结构时，我们需要考虑其厚度，这是一个连续变量。此外，我们需要为每个零件从候选材料集中选择一种材料，这是一个分类变量。此外，耐撞性评估是基于有限元分析（FEA），这是一个耗时的过程。因此，它是EOPCCV。

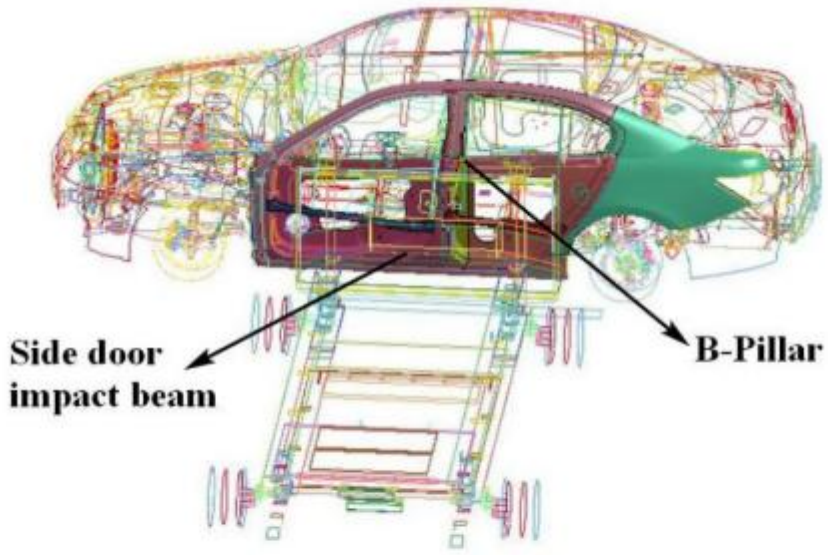

Fig. 1. Lightweight and crashworthiness design of the side body of an automobile [15].

图 1. 汽车侧面车身的轻量化和耐撞性设计

2 相关工作

2.1 代理辅助进化算法

当前大多数SAEA关注连续EOP，其利用连续函数的替代模型，例如多项式回归模型，支持向量回归，径向基函数（RBF），人工神经网络和高斯过程（GP）例如，Liu等人。提出了一种GP辅助的EA来处理中等规模的EOP。Tian等人采用GP作为代理模型，并提出了一种多目标填充准则来处理高维EOP。Wang等人。提出了全局和局部代理辅助差分进化用于昂贵的约束优化问题。Sun等人。提出了一种代理辅助的协作群优化算法来处理高维EOP。Zhang等人。将MOEA/D与GP结合起来处理昂贵的多目标优化问题。Chugh等人。提出了一种代理辅助参考向量引导的EA来解决昂贵的多目标优化问题。由于连续函数的不同代理模型对于不同类型的问题景观具有不同的强度，因此已经提出了许多具有连续函数的多个或整体代理模型的SAEA 。例如，Lim等人。提出了一个代理辅助进化框架的推广。Le et al. 引入了一个进化框架，其中包含代理的可进化性学习。Lu等人。提出了一个具有分层代理的进化优化框架。Li等人。提出了一种代理辅助粒子群优化算法的集成来解决中等规模的EOP。Guo等人。开发了一个由异质系综代理模型辅助的多目标EA框架。与连续EOP相比，很少有人对组合EOP进行尝试。目前的研究表明，具有树结构的代理模型，如随机森林（RF）和最小二乘提升树（LSBT），更适合处理组合EOP。作为代表，Wang和Jin 开发了一种用于创伤系统中约束多目标组合优化的RF辅助EA。Sun等人。将RF纳入SAEA以设计卷积神经网络的架构。此外，一些研究人员将领域知识纳入SAEA以解决组合EOP，从而提高算法搜索能力。根据我们的调查，只有几篇论文研究EOPCCV。然而，在这些论文中提出的方法主要是扩展代理模型的连续功能。因此，他们解决EOPCCV的能力是有限的。

2.2 多代理辅助

对于SAEA而言，核心问题是如何合理地使用代理模型来指导优化过程。如在第I-B节中所讨论的，连续函数的代理模型擅长于求解连续EOP，而具有树结构的代理模型在组合EOP上表现良好。有人可能会说，EOPCCV可以通过使用连续函数的替代模型和具有树结构的替代模型分别处理连续变量和分类变量来解决。然而，这种方法是不合理的，因为连续变量和分类变量可能会相互作用。因此，它们不能单独优化。直观地说，连续变量和分类变量的数量对替代模型的性能有显著影响。对于EOPCCV，我们考虑以下三种情况：1）它的大部分变量是连续变量；2）它的大部分变量是分类变量；3）连续变量的数目与分类变量的数目相似。显然，对于第一种和第二种情况，连续函数的代理模型和具有树结构的代理模型分别是很好的选择。但是，对于第三种情况，这两种代理模型都是必要的。请注意，即使对于第一种和第二种情况，我们也不能只使用这两种替代模型中的一种，因为EOPCCV同时包含连续变量和分类变量。因此，在本文中，我们同时使用这两种代理模型。

直观地说，连续变量和分类变量的数量对替代模型的性能有显著影响。对于EOPCCV，我们考虑以下三种情况：1）它的大部分变量是连续变量；2）它的大部分变量是分类变量；3）连续变量的数目与分类变量的数目相似。显然，对于第一种和第二种情况，连续函数的代理模型和具有树结构的代理模型分别是很好的选择。然而，对于第三种情况，这两种代理模型都是必要的。请注意，即使对于第一种和第二种情况，我们也不能只使用这两种替代模型中的一种，因为EOPCCV同时包含连续变量和分类变量。因此，在本文中，我们同时使用这两种代理模型。

3 本文方法

3.1 动机

在本文中，RBF和LSBT被选为代理模型的连续函数和代理模型的树结构，分别。我们选择的原因有两个方面：1）RBF是一种广泛使用的连续函数代理模型。这是简单和容易训练。此外，作为一种基于核的模型，RBF具有通过重新定义两个分类向量之间的距离来处理分类变量的潜力；2）作为一种集成代理模型，LSBT具有良好的泛化能力。此外，LSBT有可能通过离散化决策来处理连续变量空间。因此，期望它们在求解EOPCCV问题时能够相互补充。

3.2 方法概述

MiSACO算法的流程从初始化开始，然后解存档（SA）和数据库（DB）被设置。接着，蚁群优化算法的混合变量版本（ACOMV）被用来生成候选解（Candidate OP）。这些候选解通过三种策略进行多代理选择（Xsel），包括基于径向基函数（RBF）的选择、基于最小二乘提升树（LSBT）的选择以及随机选择。如果满足条件（Nls > Nmin），则对当前最佳解的连续变量进行局部搜索，这涉及到构建一个RBF子模型（RBFsub）并使用序列二次规划（SQP）进行优化，以产生局部搜索解（Xls）。然后，将多代理选择得到的解（Xsel）和局部搜索得到的解（Xls）合并，形成新的解集合（X）。这个集合将被评估，并且数据库（DB）和解决方案存档（SA）将根据这些新解进行更新。如果函数评估次数（FEs）小于最大函数评估次数（MaxFEs），算法将继续迭代；否则，将输出最佳解（Xbest）作为最终结果。流程图如图2所示，该图展示了MiSACO算法如何通过结合代理辅助选择和局部搜索策略来解决具有连续和分类变量的昂贵优化问题（EOPCCVs）。

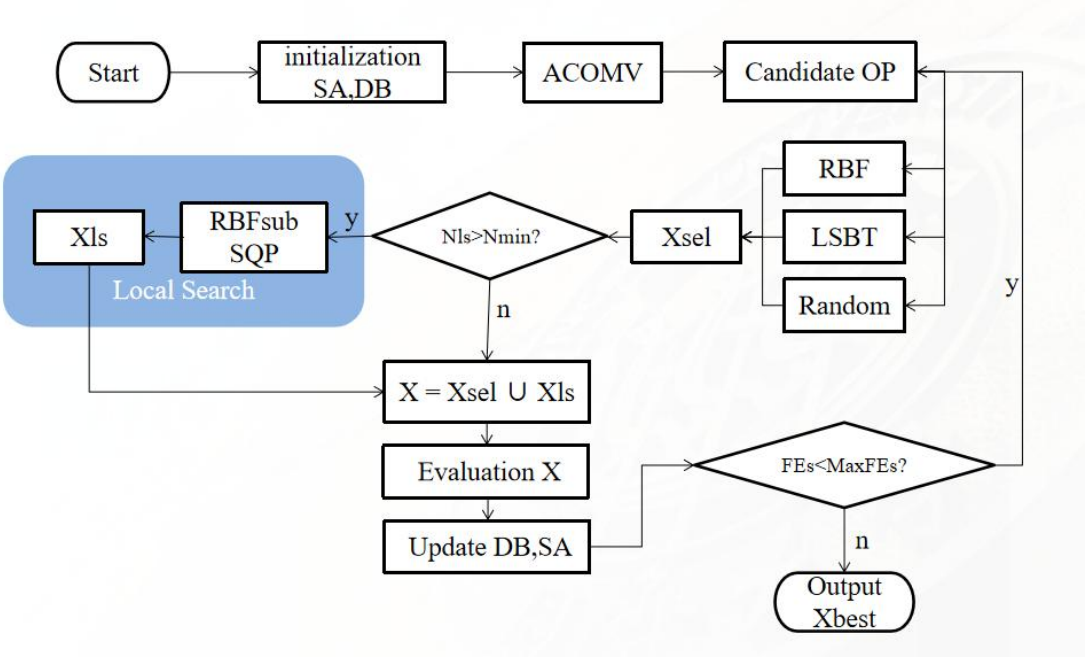

图 2. 方法示意图

4 复现细节

4.1 开源代码说明

开源代码中给出了方法的部分内容，但无法运行，复现工作根据论文中的描述以及给出的部分代码将原文实现。

4.2 复现代码展示

图3分别是对论文中提到的各个模块，基于论文给出的方法以及参数设置，进行复现

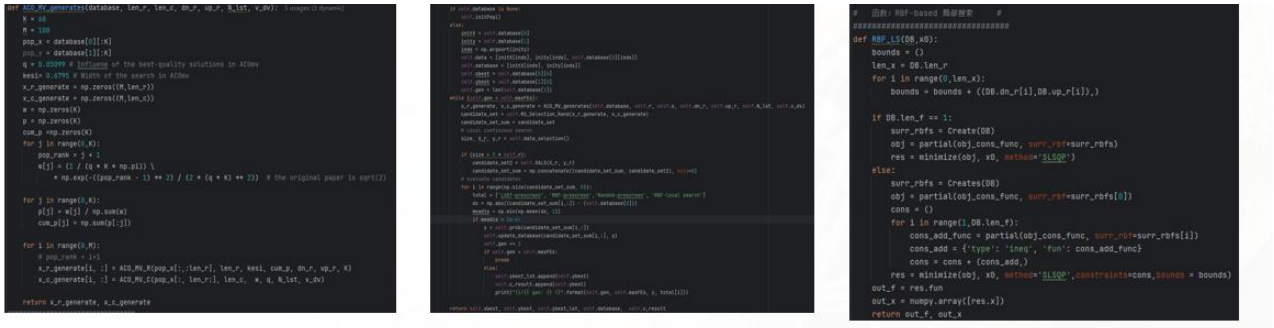

图 3. 代码展示

4.3 创新点

改进方法：基于历史数据，定义针对类变量的频率价值相关性度量。技术细节：解被分为不同的簇，计算在每个簇中，各类变量值出现的频率，基于出现频率度量不同类变量

值之间的相关性。优点：更精细地反映不同类变量值之间的差异，提升代理模型精度可以为优化算法提供更丰富的信息，帮助算法在搜索空间中更有效地探索。计算公式以及示意图如图4所示：

$$d(\mathbf{x}_i,\mathbf{x}_j)=\frac{1}{d_1+d_2}\left(\sum_{d=1}^{d_1}\frac{|x_{i,d}^{co}-x_{j,d}^{co}|}{x_{u,d}^{co}-x_{l,d}^{co}}+\sum_{d=1}^{d_2}\frac{d(x_{i,d}^{ca},x_{j,d}^{ca})}{U_d-L_d}\right)$$

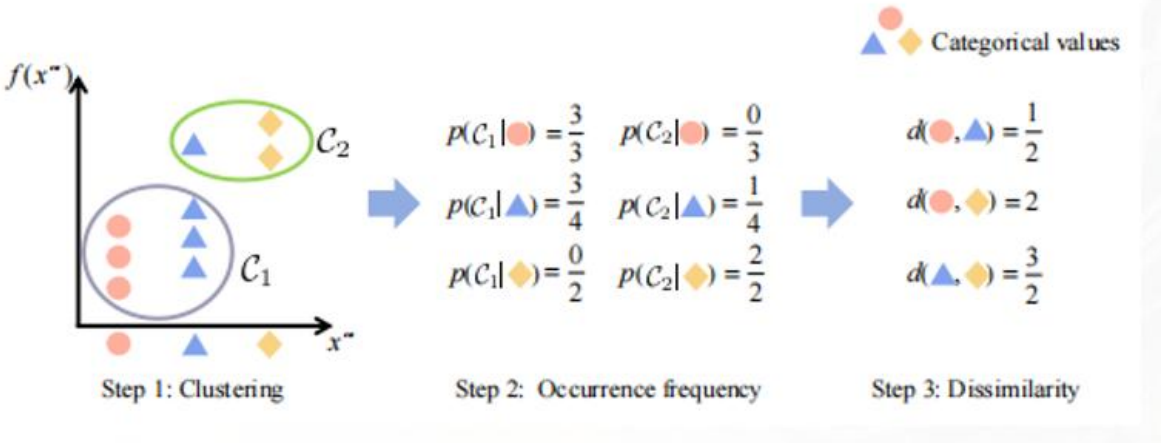

图 4. 创新公式及示意图

在图中，我们有三个不同的形状：圆形、三角形和菱形，它们代表类别变量的不同值。我们的目标是衡量这些不同值之间的不相似度。首先，我们将这些形状根据它们的特性分成两个组，即C1和C2。出现频率：然后，我们计算每个形状在每个组中出现的次数。例如，在C1组中，圆形出现了3次，三角形出现了1次，菱形没有出现。在C2组中，圆形没有出现，三角形出现了2次，菱形出现了1次。最后，我们根据这些出现频率来衡量不同形状之间的不相似度。我们通过比较每个形状在两个组中的出现频率来得出这个度量。例如，圆形和三角形在C1中的不相似度是1/2，因为圆形出现了3次而三角形出现了1次，我们取这个比例的倒数作为不相似度。在C2中，圆形和三角形的不相似度是3/2，因为三角形出现了2次而圆形没有出现，我们同样取这个比例的倒数作为不相似度。

通过这种方式，我们可以量化不同类别变量值之间的差异，这有助于优化算法更有效地在搜索空间中探索。这种方法的优势在于它能够提供更精细的度量，从而提升代理模型的精度，并为算法提供更丰富的信息。

5 实验结果分析

针对15个人工测试问题，改进前后的数据结果如表1所示：

表 1 改进前后数据对比

Comparison of Algorithms on 15 Problems

Problem	Before	After
F1	6.41e−02 ± 5.47e−02	4.33e−01 ± 1.25e+00 +
F2	3.29e+01 ± 1.18e+01	3.69e+01 ± 1.34e+01 =
F3	1.56e−01 ± 6.96e−02	6.03e−01 ± 6.68e−01 +
F4	6.07e+00 ± 2.89e+00	1.91e−03 ± 2.43e−03 −
F5	2.14e+00 ± 5.48e−01	7.40e−01 ± 2.28e−01 −
F11	1.27e+01 ± 1.68e+01	2.06e+01 ± 2.82e+01 =
F12	5.89e+01 ± 1.12e+01	5.51e+01 ± 1.39e+01 =
F13	1.02e+00 ± 1.24e+00	5.12e−01 ± 1.04e+00 =
F14	5.89e−01 ± 4.91e−01	5.04e−01 ± 6.09e−01 =
F15	9.66e−01 ± 3.16e−01	7.55e−01 ± 4.60e−01 =
F21	5.99e+01 ± 4.15e+01	2.38e+01 ± 2.35e+01 −
F22	6.51e+01 ± 6.87e+00	5.83e+01 ± 1.26e+01 =
F23	2.58e+00 ± 1.93e+00	5.22e−01 ± 1.03e+00 −
F24	3.01e+00 ± 2.06e+00	3.87e−01 ± 4.19e−01 −
F25	1.66e+00 ± 8.29e−01	8.58e−01 ± 2.34e−01 −
+/=/−		2/7/6

在这张图中，我们比较了15个问题在算法改进前后的性能。每个问题的性能通过平均值±标准差来表示，这反映了算法在多次运行中的表现及其稳定性。我们关注的是最小化问题，也就是说，我们希望找到最小的平均值，这代表了算法在解决特定问题时的最佳性能。

从图中可以看出：对于问题F4、F5、F21、F23、F24、F25，改进后的算法（After）在平均值上显示出了更好的性能，即平均值更低，这表明这些情况下改进后的算法更有效地最小化了问题。

对于问题F2、F11、F12、F13、F14、F15、F22，改进后的算法在平均值上与改进前相近（标记为=），这可能意味着改进对这些问题的影响不大，或者改进前后算法性能相当。

总的来说，改进后的算法在大多数问题上都有所提升，特别是在F1、F4和F5上，显示出了显著的改进。这表明算法的改进对于解决这些最小化问题是有效的。

下面图5展示了修改前后收敛曲线的对比，这张图通过九个子图展示了算法修改前后在多个测试问题上的收敛曲线对比，每个子图对应一个问题，并且每条曲线代表算法在不同时间点的函数误差。从图中可以看出，修改后的算法（红色曲线）在大多数情况下都显示出了更好的收敛性能，尤其是在函数评估次数较少的初期阶段，红色曲线通常低于蓝色曲线，这意味着修改后的算法能够更快地减少函数误差，接近最优解。然而，在一些问题上，如F12和F13，两条曲线非常接近，这表明算法的修改对这些问题的性能提升可能并不显著。而在F21和F24等子图中，红色曲线在后期低于蓝色曲线，这可能意味着修改后的算法在这些问题上需要更多的评估次数才能显现出其优势。总体而言，这些收敛曲线直观地反映了算法修改对于提高性能的有效性，尤其是在大多数测试问题上，修改后的算法都表现出了

更快的收敛速度和更小的函数误差

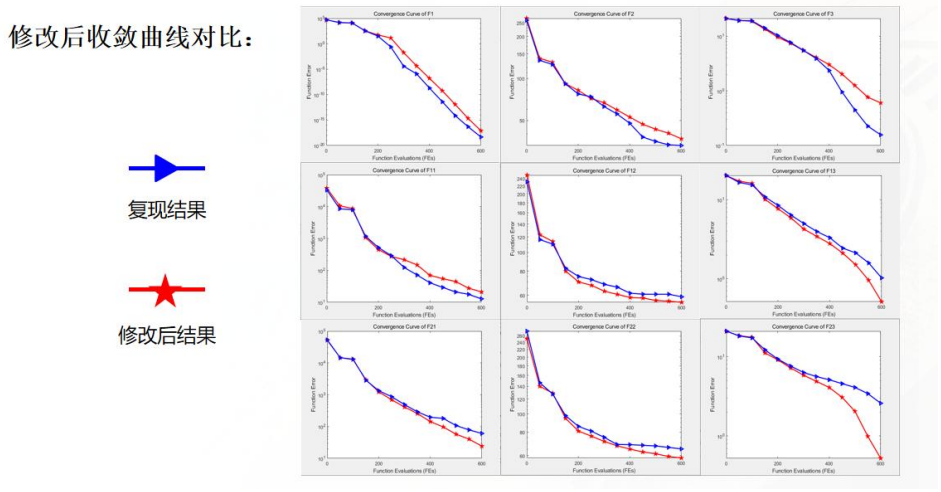

图 5.　修改前后收敛曲线对比

我们人工测试问题中解是10维的，第一排中，连续变量占8个，类变量占2个。第二排为类变量8个，连续变量2个。第三排为各占一半，可以看到修改后的结果在第一类问题中并没有很好的表现。在第二排中时好时坏。第三排则基本比修改前表现更佳。

导致这种情况的原因可能有这些。一是我们的频率价值度量与RBF结合时，模型的复杂度增加。因为我们只改进了RBF中，对类变量的处理，当类变量的占比不大时，频率价值度量对解的影响较小，且模型的复杂度增加，这可能是因为模型在小规模的类变量影响下容易受到噪声干扰，也就是第一排的收敛曲线中并没有改进甚至还不如原数据的原因。当类变量占比较大时，虽然有时好有时坏，这可能是这种度量方式有所奏效，但可能导致过拟合的情况，对数据训练模型更准确，但是在测试数据上泛化能力较差，表现会显得不稳定。类变量和连续变量的占比接近时（如第三排），模型能够更好地平衡两种变量的影响。频率价值度量能够为模型提供更丰富的类别相关信息，同时又不会导致过度复杂化，因此在这种情况下，改进后的方法展现了更好的性能。

没有较大提升的原因还可能是参数的选择不当，如RBF核函数的参数sigma，小的 sigma 值会让 RBF 核更加"局部化"，即每个训练样本的影响范围较小，可能导致过拟合；大的 sigma 值则使得每个训练样本的影响范围较大，数据点之间的相似度评价范围更广，可能导致欠拟合。

6　总结与展望

在本次算法改进过程中，由于时间限制，我们未能对原始文献中使用的特定参数进行调整。特别是在处理类别变量占比较小的问题时，改进后的频率价值度量方法增加了径向基函数（RBF）的复杂性。然而，如果RBF中的sigma参数设置过小，模型可能缺乏足够的灵活性来捕捉类别变量的模式，这可能是导致改进效果未能超越原始方法的原因之一。

此外，我们尚未对数据分簇时簇的数量等其他参数进行详尽的实验对比，以评估其对模型性能的影响。这也可能是实验结果未达到预期效果的原因之一。本次改进仅针对RBF中

的距离度量进行了优化，而从原始框架来看，有前景的解是通过三种算子选出的，其中包括随机选择算子。这种随机选择可能未能充分平衡探索与开发的需求，而仅仅是随机选取一个解，这在昂贵的优化问题中可能会浪费宝贵的评估机会。同时，算法在进化过程中对类别变量的处理是否合理，以及算法的整体框架是否能够根据这种新的度量进行相应的调整，都可能影响算法的最终表现。

针对上述问题，未来的工作应包括以下几个方面：首先，通过交叉验证和充分的实验来解决模型对不同数据集的泛化能力问题。其次，利用贝叶斯优化等方法来选择更优的超参数。再次，探索更有效的方法来提升对类别变量相关性的捕捉能力。最后，研究更优的类别变量距离度量方式，例如，能否通过某种方法学习将类别变量转化为实数空间中的向量表示，同时确保在问题中距离相近的类别变量对应的目标函数值也相近，以便更好地捕捉类别变量的语义信息和相似性。这些是我们需要继续深入研究的方向。

参考文献

[1] J. Liu, Y. Wang, G. Sun, and T. Pang, “Multisurrogate-assisted ant colony optimization for expensive optimization problems with continuous and categorical variables,” IEEE Transactions on Cybernetics, vol. 52, no. 11, pp. 11 348–11 361, 2021.