arXiv:2303.03200v1 [cs.NE] 3 Mar 2023

向量遗传编程——优化片段以进行特征提取

Philipp Fleck[1, 2⋆[0000−0001−8290−6356]] Stephan Winkler[1, 2[0000−0002−5196−4294]] Michael Kommenda[1, 3[0000−003−2049−723X]] 和 Michael Affenzeller[1, 2[0000−0001−5692−5940]]

[1] 启发式与进化算法实验室（HEAL），上奥地利应用科学大学，软件园11号，4232 哈根贝格，奥地利[2] 符号人工智能研究所，约翰内斯·开普勒大学，阿尔滕贝格街69号，4040林茨，奥地利[3] Josef Ressel符号回归中心，上奥地利应用科学大学，软件园11号，4232 哈根贝格，奥地利

摘要。 向量遗传编程（Vec-GP）扩展了遗传编程，使向量能够与常规的标量特征一起作为输入特征，并通过逐分量应用算术运算，或使用某个聚合函数将向量聚合为标量来处理向量。Vec-GP还允许仅对向量的有限片段进行聚合，而不是对整个向量进行聚合，这带来了巨大的潜力，但也引入了遗传编程必须优化的新参数。本文将这一问题形式化为一个优化问题，用于分析优化聚合函数窗口的不同策略。文中介绍了多种策略，包括随机采样和引导采样；后者利用了近似梯度中的信息。这些策略可以作为一种简单的优化算法，而该算法本身又可以应用于遗传编程中的专用变异算子。结果表明，不同的随机采样策略并不会显著影响整体算法性能，而引导策略则容易陷入局部最优解。然而，结果还表明，发现更高效、能够超越本文所述策略的算法仍具有潜力。

关键词： 遗传编程 · 向量化 · 优化 · 梯度

1 引言

用于符号回归（SR）的向量遗传编程（Vec-GP）是遗传编程的一种扩展，它将输入特征空间扩展到向量，同时保留常规标量[1]。这使向量遗传编程能够直接处理时间序列等高维数据，无需事先进行特征工程来提取标量特征。相反，向量遗传编程模型本身会从向量中提取标量

⋆ 通讯作者
philipp.fleck@fh-hagenberg.at

值，例如通过应用算术平均值等聚合函数。与传统特征工程通常仅从原始向量特征中提取标量特征不同，向量遗传编程还可以先对向量执行算术运算，再进行聚合，从而从相互作用的向量中提取特征。图 1 展示了一个示例：在计算算术平均值以获得标量之前，先将温度和压力的向量变量相除。已有文献表明，在各种基准测试中，采用特征工程的向量遗传编程优于常规遗传编程[2]。

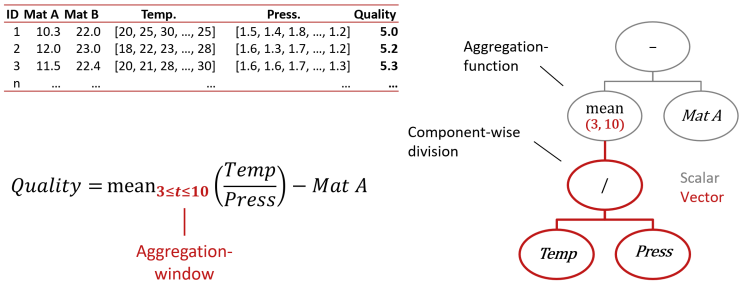


图1：一个示例数据集，其中包含标量和向量输入特征，用于预测一个标量目标值（质量）。图中还展示了一个遗传编程模型，该模型使用窗口聚合函数将向量聚合为标量。

如果向量表示时间序列，那么仅对某一特定时间范围进行聚合可能更有益。因此，可以进一步扩展向量遗传编程，使其不仅能够对整个向量进行聚合，还能对向量的子片段进行聚合，同时由遗传编程负责优化片段边界。例如，如图1所示，聚合可以仅针对某一特定时间范围进行，例如，在 $3 \le t \le 10$之间。

2 问题描述

使用带有聚合窗口的聚合函数时，遗传编程现在需要为遗传编程模型中的每个窗口聚合函数额外优化两个参数。遗传编程可以通过允许索引经由变异而改变来处理这一问题，但随机变异并不特别适合这项任务，尤其是在向量较长时。本文并不将变异作为遗传编程优化聚合窗口的唯一方式，而是迈出第一步，探讨其他更适合优化聚合窗口的策略，类似于采用高效的基于梯度的算法来优化固定遗传编程模型的连续数值参数[4]。

为聚焦于聚合窗口优化问题，我们采用一个简化的优化问题，省略所有与遗传编程相关的方面，只保留在某些固定数据上优化聚合窗口这一核心问题。

这个分段优化问题（SOP）是一个整数问题，旨在优化$\hat{a}, b \in \{i \in \mathbb{N} \mid i < N\}$所表示的起始索引和结束索引参数ˆ，同时使其最小化

$$\arg\min_{\hat{a},\hat{b}} \sum_{i=0}^{M} \Big(\underbrace{\mathrm{f}_{\mathrm{agg}}\big(\boldsymbol{v_i}[\hat{a} : \hat{b}]\big)}_{\text{optimized bounds}} - \underbrace{\mathrm{f}_{\mathrm{agg}}\big(\boldsymbol{v_i}[a : b]\big)}_{\text{known bounds}} \Big)^2, \tag{1}$$

其中，$\mathrm{f}_{\mathrm{agg}}$ 是一个固定的聚合函数，$\boldsymbol{v_i} \in \mathbb{R}^{M \times N}$ 是长度为 N的向量样本，M 表示样本数，a 和 b 表示已知的聚合索引，而$\boldsymbol{v_i}[a : b]$ 是所选向量 $\boldsymbol{v_i}$的子向量。本质上，SOP 的目标是找到使用自由索引ˆ$\hat{a}$ 和 b 进行聚合时，相较于使用已知索引 a 和 b进行聚合所产生的最小偏差。

3 方法

我们不再依赖遗传编程的变异来识别合适的聚合窗口，而是希望为遗传编程定义一种专用变异算子，以便高效优化聚合窗口的起始索引和结束索引。起始索引和结束索引既可以同时优化，即作为一个多维优化问题；也可以每次只优化一个维度，即作为一个单维优化问题。我们将在后文分析哪种形式能带来更好的结果。

观察图2中部分基准实例的适应度景观，我们可以清楚地看到，其中存在可用于高效优化的梯度信息。因此，我们首先通过在相邻索引（[5] ）上使用五点模板$h = 1$来近似梯度，然后利用该梯度引导搜索。本文提出的优化方法是一种包含两个步骤的迭代算法：首先，计算梯度并利用其引导搜索朝向有希望的区域。其次，从该有希望的区域中采样潜在解，对其进行评估，并选择其中最佳的一个用于下一次

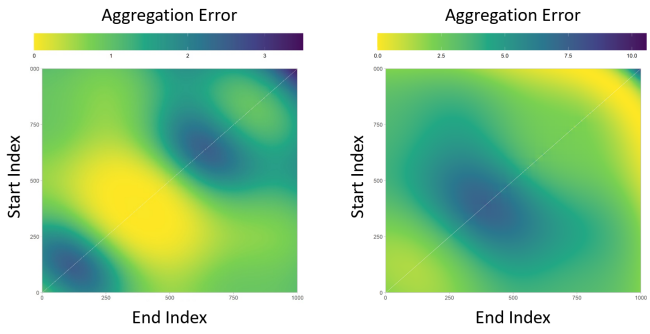


图 2：两个基准实例（x3 和 x6）的适应度景观，其中较低（较亮）的值表示搜索空间中更优的区域。

迭代。该过程持续进行，直至超过预定义的解评估总次数。

优化过程中的第一步引导负责选择一个有前景的区域，随后在第二步中从该区域抽取样本。我们定义了三种不同的引导策略，如图 3 所示：全空间表示完全不采用任何策略，即使用整个解空间。该策略可作为基线。方向 仅利用近似梯度来引导索引增大或减小，而不考虑梯度的幅值。 范围 利用近似梯度信息，通过设定的范围来界定一个有前景的区域；梯度的幅值也会影响该区域与当前索引之间的距离。



图3：在起始索引自由、结束索引固定的情况下，用于界定搜索空间的三种不同引导策略。

随机方向仅考察梯度方向，而引导范围则模仿传统的梯度下降算法，但作了两处修改，如图 4 所示。首先，我们将梯度步进取整到最接近的整数，以确保下一个索引仍然有效。此外，我们引入了一个新参数——搜索范围，用于定义下一个索引周围有多少样本被视为有前景，因而可以在之后被抽样。

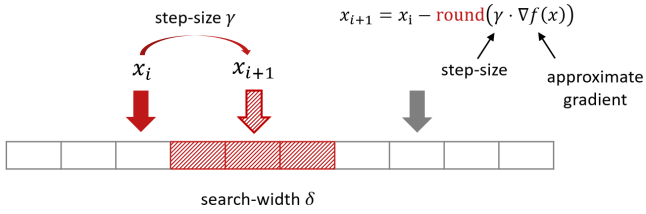


图4：对整数索引采用梯度下降步骤，并增加搜索范围参数。

定义搜索空间后，第二步负责从该搜索空间中抽取样本。本文考虑了三种

采样机制，如图 5 所示： 穷举采样 会从解空间中抽取所有样本。对于多维搜索空间，由于可能存在大量解，这通常不可行。不过，对于单维空间，这可能是可行的选择。 随机采样 会完全随机地抽取样本（不重复），样本数量由额外的样本量参数定义。 正交采样 会沿每个维度等间距地选取样本：首先在搜索空间的连续线上选取预先定义数量的等间距点，然后将其四舍五入到最近的索引，并删除重复项。

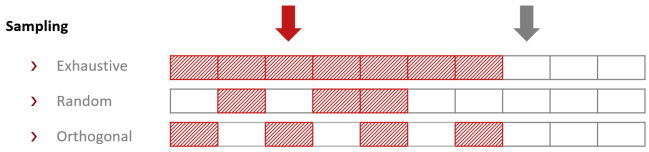


图 5：给定搜索空间的不同采样策略。

4 实验设置

为了确定上一节中定义的搜索策略的优良参数，我们在多个基准实例上对参数进行了网格搜索。不过，由于运行时间不可行，某些组合被跳过，例如在多维搜索空间中进行穷举采样。每种参数组合重复运行 20 次，并允许优化算法的每次执行最多进行 100,000 次样本评估；这意味着迭代总次数会因样本规模等因素而有所不同。

生成的基准实例具有不同特征，以呈现不同的挑战和难度。例如，如图 6 所示，第二个实例整体呈现递增趋势，而噪声保持恒定。然而，最右侧的基准实例不仅斜率变化更大，噪声也随时间增加。基准实例通过多个步骤创建：首先生成随机控制值，再将其用作其他分布的参数，最后利用这些分布生成向量。例如，最右侧的基准实例基于控制值 $m = \mathcal{U}(2,5) + \mathcal{U}(0.2, 4)/80 \cdot t$ 和 $s = \mathcal{U}(0, 0.001) + \mathcal{U}(0.002, 0.01) \cdot t$，其中 $\mathcal{U}$ 表示均匀随机变量，t 表示时间或索引。随后，使用正态分布 $\mathcal{N}(m, s^2)$ 以及先前定义的控制值生成最终向量。因此，对于这个基准实例，均值和噪声都会随着更大的 t 而增加。由于分布的参数会随 t 变化，因此每个 t 都会独立创建用于抽样的分布。基准实例的完整列表及其定义见在线补充材料：https://dev.heuristiclab.com/wiki/AdditionalMaterial。本文展示的结果使用长度为 1,000 的向量。

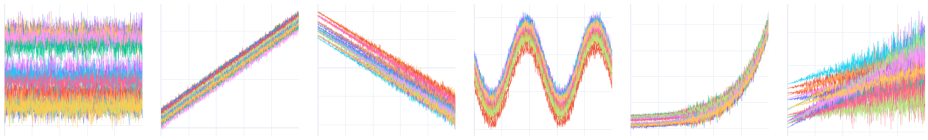


图 6：一些示例基准实例，其中每条线代表数据集中的一行。

5 结果

为了比较不同的优化策略，我们分析运行长度分布，其定义见黑盒优化基准测试（BBOB）[3]。这样不仅可以评估算法找到优质解的能力，还能展示算法的收敛速度。这一点至关重要，因为分段优化问题的优良优化策略将用作遗传编程（GP）中的变异算子，因此，在遗传编程的应用情境下，较长的运行时间会被放大。

除所呈现的结果外，我们还分析了其他参数，例如随机采样与正交采样的影响，但二者未显示出显著差异，因此本文不再进一步讨论。我们首先考察将优化问题作为单维或多维问题处理时哪种方式表现更好，如图 7所示。总体而言，结果表明，同时优化所有维度的表现略好；然而，对于某些问题实例（如 x1），使用单个维度能够获得更好的结果。我们预料到了这种行为，因为这表明部分基准实例足够简单，可以进行分离，并逐维完成优化。

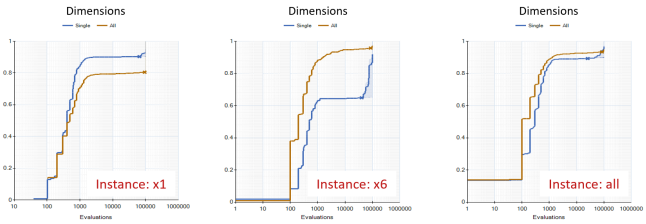


图 7：单维与多维对算法性能的影响。

接下来，我们分析定向搜索空间缩减，比较随机地将搜索空间限制在某个方向，以及利用梯度引导方向这两种方法，如图 8 所示。结果表明，效果取决于具体的问题实例。对于问题实例 x5，使用引导方向似乎会减缓收敛；总体而言，其优势并不明确。我们推测，始终利用梯度朝最佳选项移动，会迫使算法陷入局部最优解，且无法逃脱。

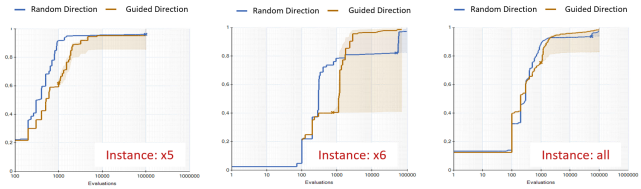
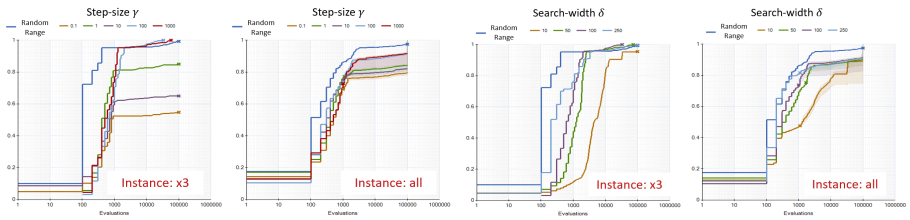


图 8：随机采样方向与引导方向的性能比较。

接下来，我们分析将搜索空间限制在特定范围内的效果。由于范围策略还有两个额外参数，即步长和搜索宽度，我们将随机采样版本与不同参数设置进行比较，结果均见图 9。步长控制从当前索引起跳的距离，因此预计较小的步长会导致较差的结果，因为跳跃距离过小会使搜索效率低下，甚至由于梯度步进中的舍入而完全停滞。实例 x3 上步长的影响表明这一假设成立：较小的步长表现不佳，如图 9a 所示。另一方面，较大的步长在实例 x3 上往往表现更好，甚至可能比随机范围策略更快地找到最佳解；然而，并非所有实例都如此。除了步长之外，搜索宽度还决定了搜索空间围绕下一个索引延伸多远。图9b 中的结果表明，较小的搜索宽度通常会使算法收敛得更慢。对于步长较小的组合，这一点尤其明显，因为当梯度步进舍入为零时，搜索范围是改变索引的唯一机制。与引导方向类似，我们推测随机方向优于引导方向，是因为算法会陷入局部最优；在一定程度上，引导范围也可能存在同样的问题。较大的搜索宽度或许有助于摆脱局部最优，但结果仍表明，在大多数情况下，随机范围是更好的选择。



(a) 步长的影响。 (b) 搜索宽度的影响。

图 9：随机采样与引导式搜索范围缩减的性能比较。

6 结论

上一节给出的结果表明，简单的随机采样通常优于使用引导方向或引导范围等引导策略。我们认为，这主要是因为引导策略的行为往往类似于贪心算法，因此很快就会陷入局部最优解。然而，在遗传编程（GP）中应用本文所述方法时，初步结果表明，引导策略往往优于相应的随机策略。我们推测，遗传编程可以通过常规的交叉和变异操作轻松摆脱局部最优解，因此陷入局部最优解并非那么关键。

尽管目前的结果表明，基于梯度的策略效果不佳，但我们仍然认为，近似梯度的信息极为有用，因为基准实例的大多数适应度景观都清楚地表明存在可用于快速优化的梯度。因此，我们将继续探索如何利用梯度，为分段优化问题设计快速优化算法。

致谢 本研究在奥地利应用科学大学 OO #875441 的博士论文项目中开展，项目为面向符号回归与时间序列分类的向量式遗传编 ¨程（*Sym- RegZeit*），由奥地利研究促进局 FFG 资助。作者还衷心感谢克里斯蒂安·多普勒研究协会以及数字与经济事务联邦部在约瑟夫·赖塞尔符号回归中心项目中提供的支持。

参考文献

1. Azzali, I., Vanneschi, L., Silva, S., Bakurov, I., Giacobini, M.：遗传编程的向量方法。EuroGP，第 213–227 页（2019） 2. Fleck, P., Winkler, S., Kommenda, M., Affenzeller, M.：用于符号回归的基于语法的向量式遗传编程。载于：Banzhaf, W., Trujillo, L., Winkler, S., Worzel, B.（编）遗传编程理论与实践 XVIII。施普林格·自然新加坡有限公司（2021） 3. Hansen, N., Auger, A., Ros, R., Finck, S., Po˘s´ık, P.：比较黑箱优化基准测试 bbob-2009 中 31 种算法的结果。载于：第 12 届遗传与进化计算年度会议附属会议论文集，第 1689–1696 页（2010） 4. Kommenda, M., Burlacu, B., Kronberger, G., Affenzeller, M.：使用非线性最小二乘进行符号回归的参数识别。《遗传编程与可进化机器》**21**(3)，471–501（2020） 5. Sauer, T.：金融领域随机微分方程的数值求解。载于：《计算金融手册》，第 529–550 页。Springer（2012）