arXiv:2301.09876v1 [cs.NE] 24 Jan 2023

利用知识图谱预测模块化优化算法的性能

Ana Kostovska[1,2][0000−0002−5983−7169], Diederick Vermetten[4][0000−0003−3040−7162], Sašo Džeroski[1,2][0000−0003−2363−712X], Panče Panov[1,2][0000−0002−7685−9140], Tome Eftimov[1][0000−0001−7330−1902], Carola Doerr[3][0000−0002−4981−3227]

[1] Jožef Stefan Institute，卢布尔雅那，斯洛文尼亚[2] Jožef Stefan International Postgraduate School，卢布尔雅那，斯洛文尼亚[3] 索邦大学，法国国家科学研究中心，LIP6，巴黎，法国[4] LIACS，莱顿大学，莱顿，荷兰

摘要。 经验数据在进化计算研究中发挥着重要作用。为了更好地利用现有数据，文献中提出了本体，以结构化的方式组织数据存储。然而，这些形式化方法在捕获领域知识方面的全部潜力仍有待展现。在这项工作中，我们评估了一个基于近期提出的 OPTION 本体扩展构建的性能预测模型。具体而言，我们首先扩展 OPTION 本体，加入表示模块化黑箱优化算法所需的词汇。随后，我们利用扩展后的 OPTION 本体，为两个模块化算法框架 modCMA 和 modDE 创建知识图谱，其中包含针对 24 个无噪声 BBOB 基准函数的固定预算性能数据。我们采用基于知识图谱嵌入的方法构建性能预测模型。通过若干不同的评估场景，我们证明，三元组分类这一知识图谱领域相当标准的预测建模任务，能够正确预测给定算法实例能否在给定问题实例上达到某一目标精度。这种方法需要对算法和问题进行特征表示。尽管问题的特征表示已发展得相当完善，我们仍希望本研究能促使学界合作开发适当的算法表示。

关键词： 算法性能预测 · 知识图谱补全 · 进化计算 · 黑箱优化

1 引言

可复现性正逐渐成为计算机科学许多领域的常态 [12]。在黑箱优化领域，这意味着许多研究人员不仅公开代码，还提供大量基准数据。尽管数据日益广泛地开放有益于整个社区，但用于组织和解读数据的工具尚未得到广泛采用。

随着基准测试环境之间的互操作性不断提升，性能数据变得更容易使用；然而，收集这些数据的算法相关信息却不那么容易获取。这在一定程度上是因为描述优化启发式方法本身就很复杂。即便是在同一个算法家族中，算子选择、参数调整策略和超参数设置的差异，也可能导致算法行为大相径庭。如果能将这些设计决策与相应的性能数据一并存储，就有望从每天产生的大量数据中提取知识。实现这一目标的一种途径是使用本体。借助本体组织的数据随后可用于各种预测研究，例如算法性能预测。

在计算机科学领域，本体是“对给定领域中可能存在的概念及其关系进行明确形式化的描述” [5]。本体是通用的数据模型，即只对具有某些共同属性的一般类型事物进行建模，而不包含该领域中特定个体的信息。另一方面，将特定个体的数据存储在有向标记图中，且其中的标签具有源自某一本体的明确定义含义，这类数据通常称为**知识图谱（KG）**。**知识图谱嵌入（KGEs）**是知识图谱中实体和关系的低维特征表示。它们为整个 KG 提供可泛化的上下文，可用于知识图谱补全、三元组分类、链接预测和节点分类等任务 [2,20]。

人们已开展多项工作，对黑箱优化领域知识的不同方面进行概念化，例如进化计算本体 [21]、多样性导向优化本体 [1]、基于偏好的多目标本体 [11],以及语义多准则决策本体 [13]。上述本体高度侧重具体细节，因此形成的算法分类仅支持用户查询高层关系。例如，搜索能够解决某一特定类别问题的算法，以及搜索已应用于特定工程问题的算法。优化算法基准测试本体（OPTION） [9] 对优化算法基准测试相关知识进行了形式化，重点是对性能和问题景观领域的数据进行形式化表示，但目前缺少对优化算法的描述符。

尽管人们在黑箱优化中对知识进行概念化做出了种种努力，但这些研究的主要目标仍局限于领域知识的表示与组织。换言之，据我们所知，已有的表示从未被用于检验其在预测研究中的表现。在本研究中，我们提出一种新方法，利用并评估以本体和知识图谱形式表示的形式语义表示所涵盖的黑箱优化知识（即优化算法、性能数据、优化问题和问题景观数据），以预测

算法性能。该预测模型通过利用基于知识图谱嵌入的算法性能分类器来构建。

我们的贡献：我们测试了以知识图谱形式对黑箱优化数据进行形式化语义表示，并据此预测算法性能的实用性。知识图谱包含有关问题景观、算法性能和算法描述符数据的信息。为捕获这些知识，我们首先扩展了 OPTION 本体。该本体原本已提供用于表示问题景观和算法性能数据的词汇，此次扩展则纳入了用于描述优化算法的表示（即描述符）。我们通过扩展 OPTION 本体进行了概念验证，加入了描述与协方差矩阵自适应进化策略（mod-CMA）和差分进化（modDE）相关的两个模块化框架的算法描述符。同一本体也可用于表示其他模块化算法。不过，我们指出，对所有在模块化框架之外提出的现有算法进行描述，是一项耗时且具有挑战性的工作，需要整个社区共同参与，才能就黑箱优化算法的标准化统一表示达成共识。

接下来，我们将扩展版 OPTION 本体的知识库转换为知识图谱，并利用它训练一个二分类器，用于预测算法能否在固定预算场景下，于预先设定的目标精度内求解给定问题（在知识图谱中分别以“已求解”和“未求解”关系表示）。在知识图谱领域，这项任务称为三元组分类。

我们通过预测 324 种 modCMA 配置和 576 种 modDE算法配置在 BBOB基准测试套件中的 24 个无噪声问题类别上、分别处于 5 维和 30 维时的性能，对所提出的方法进行了实验评估。

我们探索了不同的评估场景，以评估基于知识图谱的性能分类器的预测能力。结果表明，在平衡分类的情况下，我们的分类器能够正确预测算法是否能针对给定实例达到特定的目标精度。然而，在类别不平衡的情况下，基线（即预测多数类的分类器）表现更优。

我们通过修改流程，并基于学习到的嵌入训练一个额外的预测模型，成功提升了知识图谱嵌入分类器在不平衡分类任务中的性能。

数据与代码可用性。 我们的源代码、数据、OPTION 本体扩展、生成的知识图谱及图表可在以下地址获取：https://anonymous.4open.science/r/KG4AlgorithmPerformancePrediction-63C1/;。

论文结构。第2节介绍我们对 OPTION 本体的扩展，用于对模块化优化算法进行形式化表示。第3节介绍知识图谱的构建，以及所提出的模块化优化算法性能预测方法。实验结果在第4节讨论。第5节提出对流程的修改，以处理不平衡分类性能预测问题。

最后，我们总结本文的贡献，并展望第6节中的未来工作计划。

2 模块化优化算法的形式化表示

本文研究两类不同的进化算法：差分进化（DE） [18] 和协方差矩阵自适应进化策略（协方差矩阵自适应进化策略） [7]。这两种算法经过十多年的深入研究，已有许多变体和改进方案。其中一些改动相对较小，例如提出一种替代的种群初始化方法。更大的改动则可能通过引入重启机制或针对内部参数的新自适应方案来影响算法结构。由于这些改动大多是单独提出的，因此往往难以理解不同变体之间如何相互作用。这些因素促成了模块化算法的发展。这类框架将大量变体整合到同一个代码库中，从而能够任意组合不同变体，形成多种可能的算法配置。这不仅能够公平比较算法的两种不同变体，还能更稳健地分析算法组件之间潜在的相互作用。

对于协方差矩阵自适应进化策略，我们使用 modCMA 框架 [14], ，其中包含核心算法的许多变体。这些变体涵盖了采样分布的修改（包括镜像采样或正交采样）、用于重组的不同加权方案，以及不同的重启策略。

对于差分进化，我们使用可通过 https://github.com/Dvermetten/ModDE 获取的 modDE 软件包，版本为 v0.0.1-beta。该框架提供了丰富的变异机制，并通过不同模块来选择基数分量、所包含的差分数量，以及某些分量是否使用档案库。此外，还可以启用常见的交叉机制，以及基于若干先进差分进化版本的内部参数更新机制。

为了对模块化优化算法进行形式化表示，我们扩展了 OPTION 本体，创建了一个与 OPTION 完全兼容的新本体模块。我们遵循与 OPTION 相同的本体设计原则。例如，我们将新类与相同的上层和中层本体对齐，遵循“规范—实现—执行”本体设计模式，并采用关系本体中的关系 [17]。

我们的扩展使我们能够指定优化过程中的不同步骤，并将它们关联到相应的模块参数（见图 1）。为此，我们将模块化优化算法 类 [5] 定义为优化算法类的子类，而后者已在 OP- TION 中定义。对于模块化算法，我们还定义了一个专门的模块化

[5] 在本文的其余部分中，我们将用斜体表示本体类，而类之间的关系则用等宽字体表示。

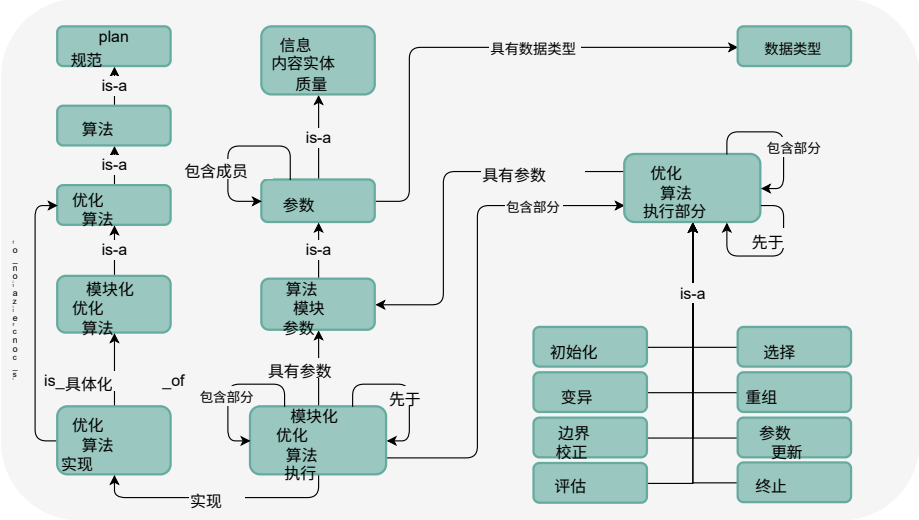

图1：用于表示模块化优化算法的 OPTION 本体扩展中包含的实体和关系。

优化算法执行。优化算法执行可以由多个子过程组成（例如，初始化、变异和重组）。为了在本体中对此进行建模，我们定义了模块化算法执行部分 类，并通过具有部分关系将其关联到模块化算法执行 类。算法执行流程通过先于关系表示。算法模块参数 通过具有参数关系，同时关联到模块化算法执行 和模块化算法执行部分 。

图2展示了 modDE 算法的本体表示。在本体中，我们为通用类创建了与 modDE 版本相对应的专门子类。例如，*modDE* 执行类是模块化优化算法执行的子类。它继承了超类的所有属性，同时还包含 modDE 算法特有的定义，例如不同的执行部分、它们的执行顺序，以及与 modDE 模块参数的关联。这里需要指出的是，图2仅展示了初始化、变异和重组等执行部分；由于篇幅限制，其他部分（即边界修正、评估、选择、参数更新和终止检查）未予展示。最后，图2展示了两个 modDE 配置（作为 modDE 类的实例），它们的区别在于交叉类型；该参数会影响优化过程中的重组部分。modCMA 算法的建模方式与此类似。

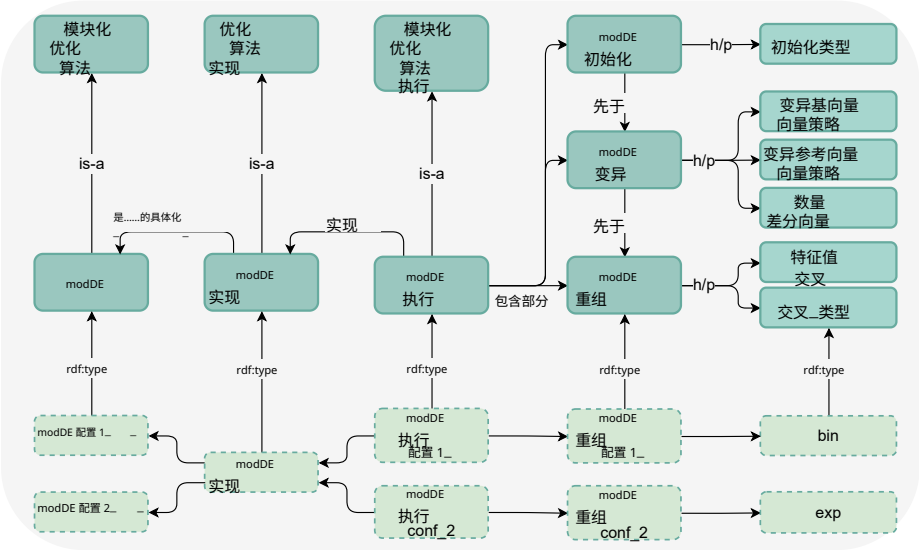

图 2：示意图展示了 modDE 算法在本体中的表示，以及 modDE 配置注释的两个示例。矩形框表示本体类，虚线矩形框表示类实例。

OPTION 本体的扩展，以及本文所研究的不同 modCMA 和 modDE 的注释，均可在我们的 Zenodo 存储库[10]中获取。

3 通过知识图谱三元组分类进行性能预测

在统一的本体框架中表示优化算法、基准问题、性能和问题景观数据，有助于构建知识图谱，使其可作为多种预测建模任务的数据资源。本文通过执行知识图谱补全任务，探讨知识图谱嵌入能否用于预测算法性能。更具体地说，我们关注的是预测问题实例与算法配置之间尚未观测到的性能关系。这对应于三元组分类任务（即判断某个算法配置是否能以给定的目标精度解决某个问题实例）。该任务可转化为二元分类任务，并可利用标准机器学习算法轻松完成。

本节首先重点介绍知识图谱的构建，然后描述基于知识图谱嵌入的自动化算法性能预测流程的详细信息。

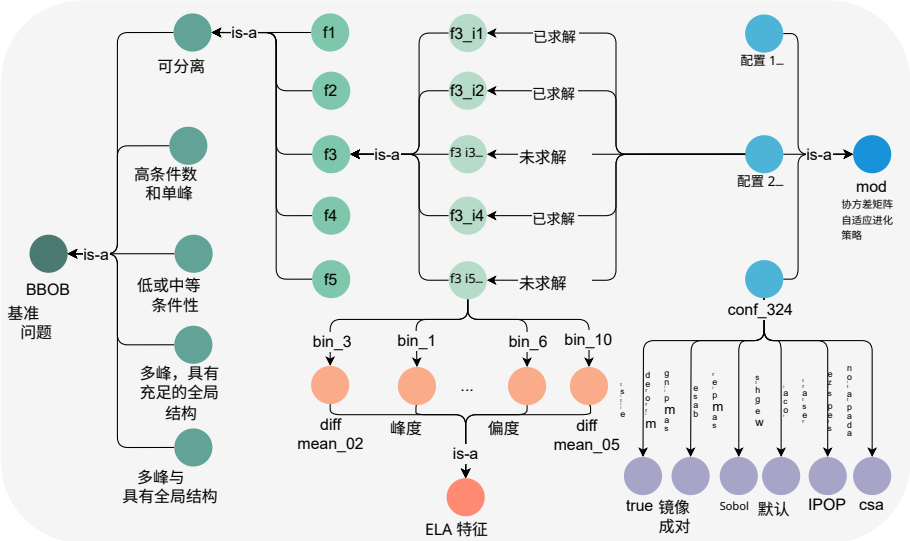

图 3：我们从原始 OPTION 本体和新的算法表示模块构建的知识图谱片段，展示了其总体结构。

3.1 知识图谱的构建

知识图谱中的两种主要节点类型是问题实例和算法实例/配置（见图 3）。我们收集了 24 个无噪声 BBOB 问题中每个问题前五个实例在 [6] 维度 $D = 5$ 和 $D = 30$下的数据，从而得到两个问题集（每个维度一个），每个问题集包含 120 个问题实例。每个问题实例均由高层和低层景观特征描述。作为高层特征，我们采用了 BBOB 测试套件中介绍的五种问题类别（即可分离、条件数低或中等、条件数高、具有适当全局结构的单峰多峰问题，以及具有较弱全局结构的多峰问题），这些类别将属性相似的基准问题归为一组。低层景观特征包括 R 包 flacco [8]中实现的 46 个 ELA 特征。我们使用一个公开可用的数据集 [16]，其中包含采用 Sobol 采样策略、样本量为$100D$ 并进行 100 次独立重复试验后，针对 24 个 BBOB 函数的前五个实例计算得出的 46 个 ELA 特征。为使分析更加稳健，我们使用计算所得 100 个特征值的中位数。最后，采用均匀分箱策略将每个 ELA 特征离散化为 10 个二进制区间。这些数据可通过 OPTION 本体 [9] 知识库获取，我们使用其 API 提取数据。

下文所述数据是在本研究中生成的，并与从 OPTION 中提取的数据相匹配，以构建知识图谱。这些算法配置来自两种不同的模块化算法：模块化 CMA-ES 和模块化 DE。由于基于模块化 CMA-ES 和模块化 DE 所有可能组合的完整枚举来收集数据在计算上不可行，

表 1：modCMA 模块及其各自参数空间的列表，共形成 324 种算法配置。

模块	参数空间
精英策略	True, 否
镜像 _采样	无、镜像、成对镜像
base_采样器	高斯、Sobol、哈尔顿
权重 _OPTION	默认，相等，1/2^lambda
局部_重新启动	None, IPOP, BIPOP
step_size_自适应 csa,	psr

表2：modDE 模块及其各自的参数空间列表，共产生 576 种算法配置。

模块	参数空间
变异 _base	rand、best、target
变异 _参考	None、pbest、best、rand
变异 _n_组件	1、2
use_档案库	True, False
交叉	二进制，实验
自适应 _方法	None、SHADE、jDE
lpsr	是，否

我们分别使用了 324 个和 576 个配置。表 1 和表 2 列出了所用模块和参数空间的详细信息，我们随后利用这些信息构造笛卡尔积，以获得不同的算法配置。在知识图谱中，每种算法配置都表示为一个节点，并通过带标签的链接/边与不同模块相连。

为获取每种配置的性能数据，我们针对每个问题实例分别进行 10 次独立的算法运行，并计算中位数。作为性能度量，我们采用算法在固定预算（即经过固定次数的函数评估后）下达到的目标精度，并使用经过 $B = \{2\,000, 5\,000, 10\,000, 50\,000\}$ 次函数评估后达到的最佳精度。

最后，根据算法的性能，问题实例与算法配置会关联一条已求解 或未求解 边。针对基准问题的情况，考虑三种不同的目标精度阈值： $T = \{1, 0.1, 0.001\}$ $5D$；针对基准问题，则为 $T = \{10, 1, 0.1\}$ $30D$。更准确地说，如果某个算法配置在给定问题实例上的目标精度等于或低于指定阈值，我们就将该算法配置与问题实例关联一条已求解 边；否则，则关联一条未求解 边。

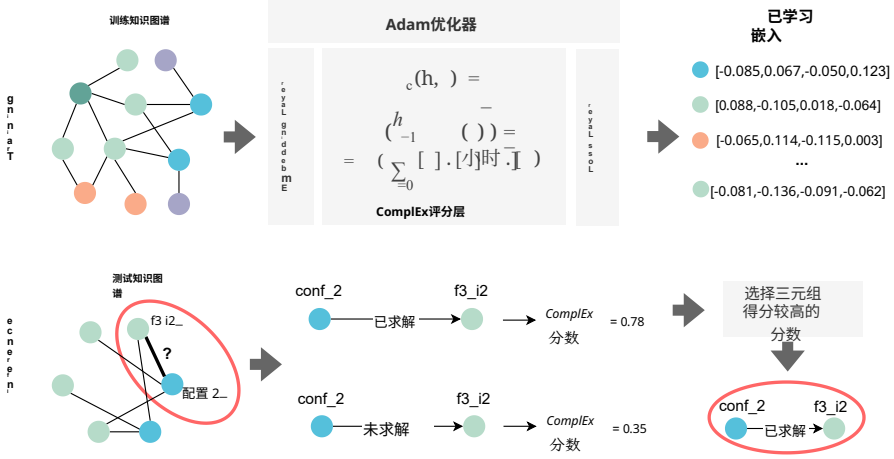

图 4：用于三元组分类的基于知识图谱嵌入的训练和推理流程。

3.2 基于知识图谱嵌入的自动化算法性能预测流程

我们的知识图谱 G 可表示为三元组集合 $\{(h,\ r,\ t)\} \subseteq E \times R \times E$，其中 E 和 R 分别是实体集和关系集。知识图谱补全的任务之一是预测两个现有实体之间未出现的关系 r （h，？t）。本文聚焦于 $\{(a,\ s,\ p)\} \subseteq A \times S \times P$三元组，其中$A \subset E$ 和 $P \subset E$ 分别是算法配置集和问题实例集，而 $S = \{solved, not\text{-}solved\} \subset R$ 是性能关系。为了预测算法配置与问题实例之间未出现的性能关系（a，？p），我们进行三元组分类。

我们提出的算法性能预测流程如图 4 所示。训练知识图谱嵌入时，我们使用 AmpliGraph 库 [3]。在训练阶段，我们使用 Xhavier 初始化器 [4] 初始化知识图谱嵌入，并在多个训练轮数中持续更新。在训练过程中，我们使用 Adam优化器最小化一个包含 ComplEx评分函数 [19] 的损失函数。该评分函数是模型特有的函数，用于为三元组赋予分数。知识图谱嵌入的评分函数用于衡量嵌入空间中两个实体相对于关系的距离。一般而言，目标是最大化正三元组的 ComplEx 模型分数，并最小化负三元组的分数。

在推理阶段，我们遍历缺少性能关系的（a，？p）三元组，并利用学习到的嵌入计算（a，$solved$，p）和（a，$not\text{-}solved$，p）三元组的 ComplEx 模型分数。我们选择 ComplEx 分数较大的三元组。

为了找出三元组分类器的最佳超参数，我们采用了网格搜索方法，对选定的超参数及其对应的搜索空间进行穷举搜索。

表 3：网格搜索中考虑的知识图谱嵌入模型超参数及其对应取值。

超参数	搜索空间
k	$[50, 100, 150, 200]$
优化器_lr	$[1e-3, 1e-4]$
loss	[成对，负对数似然，自身_对抗性]

我们选择了三个不同的超参数进行调优：(1) k——嵌入空间的维度；(2) optimizer_lr——优化器的学习率；(3) loss——训练期间使用的损失函数类型，例如成对边际损失、负损失概率和对抗采样损失。本研究所用超参数的搜索空间如表 3 所示。

使用单独的验证集估计最优超参数集。我们最初将训练轮数设为 500，但会启用一种机制：如果连续 10 次验证检查/轮数内性能没有提升，就提前终止训练。

对于三元组分类器，我们报告 F1 分数，其定义为精确率和召回率的调和均值：$F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$。~~作为基线，我们使用预测多数类~~（已求解/未求解类）的分类器。在网格搜索步骤中也使用相同的评估指标作为启发式算法，并根据 F1 分数确定表现最佳的模型。

4 评估结果

本节中，我们报告两种不同评估场景的结果，这两种场景根据训练和测试数据的选取方式而有所不同。第一种采用随机留出性能三元组验证，第二种则采用留出问题实例/算法配置的验证方式。

4.1 留出随机算法性能三元组进行验证

在第一组实验中，我们使用第3节所述的方法进行算法性能预测。由于我们考虑了两种问题维数、四种预算和三个目标精度阈值，因此对于两种算法（modCMA 和 modDE），每种算法总共有 24 个不同的知识图谱。

对于每个知识图谱，我们按 60:20:20 的比例划分性能三元组。也就是说，60% 的三元组分配到训练集，20% 分配到验证集，其余 20% 分配到测试集。我们以分层方式进行划分，使性能链接的分布与原始知识图谱保持一致。由于划分依据的是性能链接的分层样本，与特定问题实例或算法配置相关的性能链接可能会被分到

训练/验证集和测试集中。当所选问题组合中的大多数问题实例都已知算法配置的性能，但部分问题实例的性能未知时，可以使用这种方法。请注意，训练知识图谱不仅包含性能三元组，还包含其他类型的实体和关系，例如高层和低层景观特征，以及以模块及其参数描述的算法配置；而验证集和测试集仅包含关注的链接/三元组——“已求解”和“未求解”性能链接。

表4展示了在 5 维和 30 维的知识图谱复合问题中，modCMA 和 modDE 知识图谱里“已求解”性能关系相对于“未求解”性能关系的占比。可以看出，在某些情形下，我们面对的是不平衡分类，尤其是在$30D$ 问题中。

表4：在不同的固定预算情形和目标精度阈值下，由 a) $5D$ 和 b) $30D$ 问题构成的知识图谱中 modCMA 和 modDE 算法的已求解 链接占比。

	modCMA				modDE			
	2000	5000	10000	50000	2000	5000	10000	50000
1	62.9	68.2	71.3	78.9	27.7	42.2	58.1	81.4
0.1	46.8	54.1	57.1	63.7	13.2	23.3	33.1	62.8
0.001	36.9	47.8	50.7	55.9	9.4	14.4	21.7	56.2

(a) $5D$ 问题

	modCMA				modDE			
	2000	5000	10000	50000	2000	5000	10000	50000
10	35.1	46.2	49.9	68.1	13.0	21.6	29.1	46.1
1	10.7	16.0	21.0	40.9	1.6	4.4	8.0	17.5
0.1	6.1	08.8	12.5	31.5	1.2	3.2	6.2	12.7

(b) $30D$ 问题

表5展示了三元组分类器在不同固定预算场景和目标精度阈值下，对 $5D$ 和 $30D$ 问题的 F1 分数，以及该分类器相较于基线的提升百分比。结果表明，在不存在不平衡分类的情况下，三元组分类器能够提升性能。而在不平衡分类的情况下，性能会下降，这意味着此时我们提出的流程需要进行调整。

表5：使用 ComplEx 评分模型，对由 $5D$ 和 $30D$ 问题组成的知识图谱，在不同固定预算场景和目标精度阈值下得到的 modCMA 和 modDE 算法性能三元组分类器的 F1 分数及其相较于基线的提升百分比。

	2000	5000	10000	50000
1	0.922/19.43%	0.942/16.15%	0.944/13.33%	0.953/8.05%
0.1	0.905/30.22%	0.933/32.91%	0.937/29.06%	0.942/21.08%
0.001	0.893/15.37%	0.944/37.61%	0.944/40.48%	0.946/31.75%

(a) $5D$ 问题 - modCMA

	2000	5000	10000	50000
1	0.848/1.07%	0.876/19.67%	0.901/22.59%	0.946/5.46%
0.1	0.788/-15.18%	0.82/-5.53%	0.858/6.98%	0.922/19.43%
0.001	0.831/-12.62%	0.745/-19.20%	0.803/-8.65%	0.919/27.64%

(b) $5D$ 问题 - modDE

	2000	5000	10000	50000
10	0.937/19.06%	0.927/32.62%	0.939/40.78%	0.953/17.51%
1	0.902/-4.45%	0.808/-11.50%	0.855/-3.17%	0.929/25.03%
0.1	0.935/-3.41%	0.89/-6.71%	0.852/-8.68%	0.921/13.28%

(c) $30D$ 问题 - modCMA

	2000	5000	10000	50000
10	0.9/-3.23%	0.931/5.92%	0.947/14.10%	0.948/35.24%
1	0.504/-49.19%	0.792/-19.02%	0.846/-11.69%	0.87/-3.76%
0.1	0.695/-30.08%	0.735/-25.30%	0.835/-13.74%%	0.885/-5.04%

(d) $30D$ 问题 - modDE

4.2 留出问题实例/算法配置进行验证

第二组实验评估了一种具有实际意义的场景：对于给定的问题实例/算法配置，不存在性能数据。为了评估三元组分类器在此设置下的性能，我们研究了另外两种评估场景：

– **留出问题实例进行验证：**在此场景中，我们从 24 个 BBOB 问题中各选取一个问题实例，将其所有性能三元组用于测试；从另一个问题实例中选取性能三元组用于验证，其余三个用于训练。例如，我们将 24 个 BBOB 问题各自的前三个实例用于训练，第四个实例用于验证，第五个实例用于测试。我们

表6：对于移除了部分问题及由 5D 问题组成的算法配置的所有性能链接的知识图谱，在不同预算和目标精度阈值为 0.1 的情况下，modCMA 算法性能三元组分类器的 F1 分数及其相较于基线的提升百分比。

	留出问题验证	留出算法验证
2000	0.728（0.006）/4.90	0.893 (0.009)/28.67
5000	0.761 (0.018)/8.40	0.915 (0.011)/30.34
10000	0.766(0.008)/5.36	0.91 (0.011)/25.17
50000	0.797(0.014)/2.44	0.913 (0.002)/17.35

表7：对于移除了所有性能链接的知识图谱，在由不同预算和目标精度阈值 0.1 下的 5D 问题组成的问题与算法配置子集中，modDE算法性能三元组分类器的 F1 分数及其相对于基准线的提升百分比。

	留出问题验证	留出算法验证
2000	0.854(0.061)/-8.07%	0.79(0.035)/-14.96%
5000	0.837(0.022)/-3.57%	0.85(0.021)/-2.07%
10000	0.796(0.024)/-0.75%	0.825(0.010)/2.87%
50000	0.83(0.010)/7.51%	0.822（0.013）/6.48%

重复此操作五次，以便五个实例各自在测试集中出现一次。

– **留出算法配置进行验证**：在此情形下，算法配置按照 60:20:20 的比例进行划分，其性能三元组分别用于训练、验证和测试。为评估结果的稳健性，我们独立重复此过程五次。

我们将这些评估场景应用于由 $5D$基准问题和 modCMA 算法配置构成的知识图谱，覆盖四种不同的预算，目标精度阈值为 0.1。表6展示了三元组分类器的平均 F1分数（取五次运行的平均值）、标准差以及改进百分比。与第4.1节类似，在采用平衡分类时，我们的方法优于基线。表7给出了 modDE性能分类器的评估结果，其模式与前一种情况类似。

表8：在 $30D$ 问题实例上，对用于modDE性能预测的两种所提出流程进行比较，目标精度为0.1。结果格式为：分类器的F1分数/基线的F1分数/相较于基线的改进百分比。

	知识图谱 - ComplEx评分	随机森林分类器
2000	0.695/0.994/-30.08%	0.999/0.994/0.52%
5000	0.735/0.984/-25.30%	0.998/0.984/1.43%
10000	0.835/0.968/-13.74%	0.996/0.968/2.91%
50000	0.885/0.932/-5.04%	0.991/0.932/6.27%

5 解决不平衡分类问题

为了解决不平衡分类情况下出现的问题，我们修改了第3.2节所述的流程。具体而言，在知识图谱训练阶段之后，我们增加了一个训练层，基于学习到的嵌入训练随机森林（RF）分类器。我们的数据实例是性能三元组。为了生成随机森林分类器所需的数据，我们将每个 $(a,\ s,\ p)$ 三元组表示为a实体和 p 实体的嵌入向量拼接。

我们使用随机森林分类器进行推理，而不是使用 ComplEx 模型的分数。

我们使用第4.1节实验中类别最不平衡的场景来评估这一方法，即在目标精度阈值为 0.1 时，预测 modDE 在 $30D$ 问题实例上的性能。我们使用 10 个估计器训练随机森林分类器，该分类器由 scikit-learn 库[15]实现。其余超参数均采用默认值。表8中，我们将使用第3.2节所述流程训练的分类器的 F1分数，与本节所述随机森林分类器的分数进行比较。结果表明，在学习到的嵌入上训练随机森林分类器能够提升 F1 分数。由于我们处理的是类别不平衡的分类问题，因此评估指标的选择至关重要。这里，我们还报告 AUC ROC分数，作为第二种评估指标。使用 AUC ROC 评估指标，我们得出了相同结论：在 2000、5000、10000 和 50000 的固定预算场景下，随机森林分类器的 AUC ROC分数分别为 0.994、0.998、0.998 和 0.987。我们的基线（预测多数类的分类器）在不同固定预算场景下的 AUC ROC分数均为 0.5。我们认为，结果有所改善，是因为嵌入空间具有可分性，随机森林模型在预测算法性能时能够捕捉到这种可分性。不过，这一假设仍需进一步研究。

6 结论与未来工作

本文研究黑箱优化的形式化语义表示在自动预测算法性能方面的预测能力。为此，我们评估了使用知识图谱预测 modCMA 和 modDE 优化算法在无噪声 BBOB 函数上的性能是否可行。具体而言，我们旨在探究能否训练知识图谱嵌入，用于预测固定预算场景下，具有给定目标精度的算法配置与问题实例之间的性能链接/三元组（已求解 或 未求解 链接）。知识图谱将问题景观和算法性能数据与模块化算法配置相关的数据相结合。

结果表明，当我们随机选择性能三元组作为测试集（经典的知识图谱补全场景）时，在平衡分类的情况下，我们提出的三元组分类器优于基线。在不平衡分类的情况下，分类器的性能会下降，且不如基线。在第二组实验中，我们采用了一种“更严格”的评估场景，尝试预测测试集中出现的问题实例和算法配置所对应的所有性能链接（训练集中不包含任何性能链接）。我们观察到与前一种情况类似的规律。为了解决不平衡分类情况下的性能下降问题，我们修改了所提出的流程，并在学习到的嵌入之上训练了一个随机森林分类器。

对于未来工作，我们计划测试不同的知识图谱嵌入训练方法，并提高其可解释性。将我们的结果与其他方法（如关系学习和图频繁模式挖掘）进行比较，也会很有意义。

我们已经证明，由本体知识库构建的 modCMA 和 modDE 模块化优化算法的实验数据知识图谱，可用于预测研究。测试这种方法是否适用于其他模块化框架，将会很有意义。不过，更具挑战性的任务是将其扩展到模块化框架之外，因为我们需要开发一种正式的标准词汇，用于表示算法算子、它们的超参数及其相互作用。我们希望这项工作能够激励学界合作开发适当的算法表示方法。

致谢。作者感谢斯洛文尼亚研究署通过项目资助（编号 P2-0103 和 P2-0098）、项目资助（编号 N2-0239 和 J2-4460）、给予 AK 的青年研究人员资助，以及斯洛文尼亚与法国双边项目资助（编号 BI-FR/23-24-PROTEUS-001，即 PR-12040）提供的支持；同时感谢欧盟委员会通过资助（编号 952215，即 TAILOR）提供的支持。我们的研究还获得了 ANR-22-ERCS-0003-01 项目 VARIATION 的支持，并通过 aSPECIES 奖学金支持 Ana Kostovska。

参考文献

1. Basto-Fernandes, V., Yevseyeva, I., Deutz, A., Emmerich, M.：《多样性导向优化综述：问题、指标与算法》。载于：EVOLVE——概率、集合定向数值方法与进化计算之间的桥梁 VII，第 3–23 页。Springer（2017） 2. Chen, X., Jia, S., Xiang, Y.：《综述：知识图谱上的知识推理》。载于《专家系统与应用》**141**，112948（2020） 3. Costabello, L., Pai, S., Le Van, C., McGrath, R., McCarthy, N., Tabacof, P.：《Ampligraph：用于知识图谱表示学习的库》。检索日期：2019 年**10**月 10 日（2019） 4. Glorot, X., Bengio, Y.：《理解深度前馈神经网络训练的难点》。载于：AISTATS 会议论文集，第 249–256 页。JMLR 研讨会与会议论文集（2010） 5. Gruber, T.：《探讨知识共享本体设计原则》。载于《国际人机研究杂志》**43**(5-6)，907–928（1995） 6. Hansen, N., Finck, S., Ros, R., Auger, A.：《2009年实参数黑箱优化基准测试：无噪声函数定义》。研究报告 RR-6829，法国国家信息与自动化研究所（2009 00362633 7 ），https://hal.inria.fr/inria- . Hansen, N., Ostermeier, A.：《在进化策略中自适应任意正态变异分布：协方差矩阵自适应》。载于：CEC 会议论文集，第 312–317 页。IEEE（1996） 8. Kerschke, P., Trautmann, H.：《用于探索性景观分析的 R 软件包 FLACCO 及其在多目标优化问题中的应用》。载于：CEC，第 5262 7748359 9 5269 2016 10 1109 2016 –（7 月）。https://doi.org/ . /CEC. . . 9. Kostovska, A., Vermetten, D., Doerr, C., Dzeroski, S., Panov, P., Eftimov, T.：《OPTION：优化算法基准测试本体》。载于：GECCO 会议论文集，伴随材料（2021） 10. Kostovska, A., Vermetten, D., Dzeroski, S., Panov, P., Eftimov, T., Doerr, C.：《使用知识图谱预测模块化优化算法的性能——数据》（2022 年 11 月） 11. Li, L., Yevseyeva, I., Basto-Fernandes, V., Trautmann, H., Jing, N., Emmerich, M.：《构建和使用基于偏好的多目标进化算法本体》。载于：EMO 会议论文集，第 406–421 页。Springer（2017） 12. López-Ibáñez, M., Branke, J., Paquete, L.：《进化计算的可复现性》。载于《ACM进化学习与优化汇刊》**1**(4)，1–21（2021） 13. Mahmoudi, G., Muller-Schloer, C.：《语义多准则决策 sem-cdm》。载于：2009 年 IEEE 计算智能多准则决策研讨会（MCDM），第 149–156 页。IEEE（2009） 14. de Nobel, J., Vermetten, D., Wang, H., Doerr, C., Bäck, T.：《通过调优评估新思路与现有算法模块协同作用的益处》。载于：GECCO 会议论文集，伴随材料，第 1375–1384 页。ACM（2021） 15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.：《Scikit-learn：Python 中的机器学习》。载于《机器学习研究杂志》**12**，2825–2830（2011） 16. Renau, Q., Doerr, C., Dreo, J., Doerr, B.：《用于研究“探索性景观分析对采样策略高度敏感”的实验数据集》（2020 年 6 月）。https://doi.org/10.5281/zenodo.3886816

17. Smith, B., 等：生物医学本体中的关系。Genome Biology **6**(5), R46 (2005) 18. Storn, R., Price, K.：差分进化——一种用于连续空间全局优化的简单高效启发式算法。Journal of Global Optimization **11**(4), 341–359 (1997) 19. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.：用于简单链接预测的 ComplEx 嵌入。载于：国际机器学习会议。第 2071–2080 页。PMLR (2016) 20. Wang, Q., Mao, Z., Wang, B., Guo, L.：知识图谱嵌入：方法与应用综述。IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017) 21. Yaman, A., Hallawa, A., Coler, M., Iacca, G.：介绍 ECO：进化计算本体。载于：EvoApplications 论文集。第 603–619 页。Springer (2017)