arXiv:2304.01642v2 [cs.NE] 7 Apr 2023

通过交互式质量多样性实现设计空间的可控探索

预印本

康斯坦丁诺斯·斯菲卡斯
数字游戏研究所
马耳他大学
姆西达，马耳他
konstantinos.sfikas@um.edu.mt

Antonios Liapis
数字游戏研究所
马耳他大学
姆西达，马耳他
antonios.liapis@um.edu.mt

Georgios N.Yannakakis马耳他大学数字游戏研究所，马耳他姆西达
georgios.yannakakis@um.edu.mt

摘要

本文介绍了一种基于质量多样性（QD）搜索的用户驱动进化算法。在设计过程中，用户反复从展示的备选方案中进行选择，而这些选择会影响后续结果。我们旨在解决交互式进化中的两个主要问题：（a）向用户展示的备选方案应尽量少，以减轻认知负担；（b）展示的备选方案应具备多样性，同时与用户上一次的选择相似，以减少用户疲劳。为解决这些问题，我们实现了 MAP-Elites 算法的一种变体：从行为域的一个小区域（窗口）中抽取并展示备选方案。用户做出选择后，窗口会以所选个体的行为特征为中心，进化过程从窗口内选择父代以产生后代，并抽取新的备选方案。本质上，我们定义了一种局部质量多样性的自适应系统，由用户的选择引导搜索朝行为域的特定区域推进。我们通过生成建筑布局这一约束优化任务来测试该系统，并采用双档案库方法利用质量多样性。结果表明，尽管该系统的全局探索不如 MAP-Elites 显著，但基于对可控人工用户的实验，它能找到更符合用户偏好的解。

关键词 质量多样性 · 交互式进化 · 人机交互 · 创意支持工具 · 平面图生成

1 引言

交互式进化计算（IEC）是一种“在优化过程中使用真实人类的主观评价进行系统优化”的形式 [1]。狭义而言，用户的主观评价实质上在进化计算（EC）优化过程中承担适应度的作用 [2]。IEC 的主要优点在于，它适用于难以甚至无法定义适应度函数、但可以由真实人类进行评价的问题。然而，IEC 很容易导致用户疲劳 [2]：EC 通常在种群规模较大、迭代代数较多时效果更好，而人类检查和比较解决方案的能力是一种宝贵且有限的资源。为缓解用户疲劳，人们提出了多种解决方案；其中最有前景的包括通过用户模型进行函数逼近 [3, 4], ，使进化能够在两次用户交互之间运行多个迭代代数，以及交互式差分进化 [1]。然而，将用户模型训练为近似器，可能难以快速适应用户当前的交互 [5], ；与此同时，基于种群的元启发式算法适用于实值连续基因型表示，而这并非总是可行。

本文介绍一种新颖的 IEC 算法，旨在提供高度的用户控制，同时减少用户疲劳。我们展示了如何利用质量多样性（QD）算法的照亮能力实现这一目标 [6]。我们设想一种名为“用户控制的 QD 探索”的混合系统：用户的主观标准将质量多样性搜索定位并控制在行为空间的一部分区域内。我们通过修改一种流行的 QD 算法 MAP-Elites [7]，的基本运行方式来实现这一概念，并设计了如下交互循环：我们将算法的运行限制在一个覆盖特征图小范围区域的窗口内，使其在此局部扩展档案库若干代。随后，从窗口内采样设计备选方案并呈现给设计者。最后，用户的选择决定窗口接下来移动的方向。这些步骤概述了本文所介绍的具体算法实现的功能，我们将其称为用户控制的 MAP-Elites（UC-ME）。

我们使用 [8] 中提出的一种生成方法，在生成建筑布局这一受约束的设计问题上测试 UC-ME。该问题包含多个约束，并采用了特定的表示方式和遗传算子。为了将 UC-ME 应用于受约束问题，我们借鉴 FI-MAP-Elites [8] 的思路，并对 UC-ME 进行调整，使其能够使用可行与不可行精英个体的双档案库。为了测试 UC-ME 如何满足不同的潜在用户目标，我们采用了具有不同选择标准的人工用户。

2 相关工作

以下章节将介绍与本研究直接相关的主题：交互式进化与质量多样性。

2.1 交互式进化计算

如第 1 节所述，对于难以构建适应度函数、且构建难度不亚于问题本身的领域，IEC 是一种强大的工具。IEC 已应用于抽象图像 [9], 3、D 模型 [10], 和声音 [11], 的生成，其过程也被游戏化，即完全融入游戏机制，作为游戏玩法的一部分 [12, 13, 14]。

人们提出了多种方法来缓解交互式进化计算（IEC） [2], 中的用户疲劳，例如向用户展示种群的一个子集、通过代理适应度函数 [3] 进行多代进化，以及使用在线工具让多个用户分担交互式进化计算的工作量 [13, 9, 12]。作为交互式进化计算中代理适应度函数的一个具体例子，既有研究提出了设计者的计算模型 [15, 16] 来驱动进化。 [16] 中的相关研究表明，在每次人机交互后更新的设计者模型可在中间代（用户选择之间）用作适应度函数，从而减少用户参与。该方法的主要缺点是牺牲了用户控制的即时性，而且必须在对用户最新表达的偏好过度拟合与拟合不足之间取得平衡。

其他方法试图通过加快进化过程及其向最优解的收敛，来解决用户疲劳问题。这些方法包括适应度景观近似 [17, 18]，交互式差分进化 [19, 20, 21] 和收敛点估计 [22]，等。这些方法的主要局限在于，它们大多适用于基因型表示为（或可表示为）定义连续空间的数值向量的问题。然而，设计问题（例如本文研究的用例）往往需要采用特定的基因型表示方式，这些表示方式很难甚至不可能简化为实数向量。

2.2 质量多样性

质量多样性（QD）搜索 [6] 是一种进化计算（EC）形式，能够同时优化并增加所生成解群体的多样性。这种方法的代表性算法包括带局部竞争的新颖性搜索 [23] 和 MAP-Elites [7]。QD 算法最初旨在解决进化计算中的欺骗问题，方法是确保群体在与问题适应度无关的维度上保持多样性。然而，人们已认识到，这类算法能够展现问题空间，从而激发人类创造力 [24, 25]。因此，QD 搜索已从最初在进化机器人领域的应用 [7, 23] 扩展到各种与设计相关的问题 [26, 27, 28]。

MAP-Elites [7]是研究最为充分的质量多样性（QD）算法之一。该算法将多维特征空间细分为多个单元格。每个单元格可以包含一个个体，从而在各自的行为表征 niche 中形成一个精英个体档案库。在运行过程中，MAP-Elites 从已占用的单元格中随机选择个体，对其进行变异并生成后代。根据适当的行为表征 niche 为后代分配位置，该 niche 由作为特征图坐标的多个行为表征（BCs）定义。如果该 niche 对应的单元格当前为空，则将后代放入其中；如果该单元格已被占用，则仅当后代的

适应度更高时，后代才会替换该单元格中的精英个体。经过多次重复这一过程，算法既会扩展档案库（朝向所有可能的 BCs），也会提升其中精英个体的适应度。

2.2.1 约束质量多样性

当问题包含硬约束时，如果解满足一组准则，则被视为可行解，否则被视为不可行解。重要的是，可行性通常需要特殊处理，因为它往往与适应度无关，而且可能要求同时满足不同的约束。尽管这类问题并非质量多样性算法关注的重点，文献中仍有多种方法将质量多样性与约束求解相结合 [29, 26]。其中许多方法在约束求解方面借鉴了 FI-2-Pop GA [30]。简而言之，FI-2-Pop GA 在两个种群上运行，一个包含可行解，另一个包含不可行解。父代从任一种群中选出，只与同一种群中的其他父代交叉繁殖，但产生的后代可能会根据其可行性被分配到另一个种群。

在约束质量多样性搜索中，FI-MAP-Elites [8] 结合了 FI-2Pop GA [30] 和 MAP-Elites [7]，维护两个档案库（一个包含可行精英个体，另一个包含不可行精英个体）。父代选择在两个档案库之间交替进行，而变异产生的后代可根据其可行性转入不同的档案库。不可行档案库中的质量取决于个体距离可行状态的远近。两个档案库无需共享行为表征（尽管通常会共享）。有关该算法的更多详情，请参阅 [8]。

2.2.2 交互式质量多样性

尽管 QD 非常适用于设计问题，但在混合主动式场景下对 QD 的研究相对较少。Alvarez 等人关于交互式约束质量多样性的研究是其中值得关注的例子。在这些研究中，用户交互通过两种方式实现：(a) 用户可以动态控制 MAP-Elites 算法的参数，从而以不同方式探索设计空间。其次，在较低层面上，设计者可以手动干预生成的设计，或从头设计自己的方案，再将其作为种子交由 MAP-Elites 自动生成变体。需要强调的是，尽管这些操作模式依赖用户主动发起，但并不符合 IEC 的狭义定义（见第 1 节）。

3 用户可控 MAP-Elites

我们提出用户可控 MAP-Elites（UC-ME）算法，使用户能够控制 QD 探索的方向和计算资源。该算法仅允许从精英个体档案库的一个小窗口中进行父代选择，并根据用户的选择移动此选择窗口。用户可以从当前窗口内采样出的少量备选方案（本文中为四个）中选出一个最喜欢的方案。本文介绍 UC-ME 算法的总体流程，以及向用户展示设计备选方案的不同采样方法。

3.1 算法初始化

UC-ME 首先通过随机初始化方法生成一定数量的初始个体，并根据其行为表征将其放入 MAP-Elites 档案库中。此步骤为档案库提供初始种子，以便与人类用户进行交互。大小为 $w \times w$ 的初始选择窗口以现有精英个体 BC 值的平均值所在的单元格为中心；如果该单元格为空，则以最近的精英个体为中心。窗口大小（w）是 UC-ME 的一个参数，其值应远小于特征图的分辨率。

3.2 交互操作

算法初始化后，交互会话即可开始。在交互会话期间，以下步骤会无限重复，直到设计者决定结束会话。

1. **设计备选方案采样：** 算法从选择窗口内采样 D 个设计备选方案，供设计者选择。2. **设计者输入：** 设计者选择一个心仪的设计。3. **选择窗口定位：** 选择窗口以设计者上一次选择的坐标为中心。4. **窗口化档案库扩展：** 算法运行 N_e 次评估，从窗口内选择父代。变异后的后代经过评估后，根据其行为表征坐标放入相应的档案库单元，且不受窗口限制，

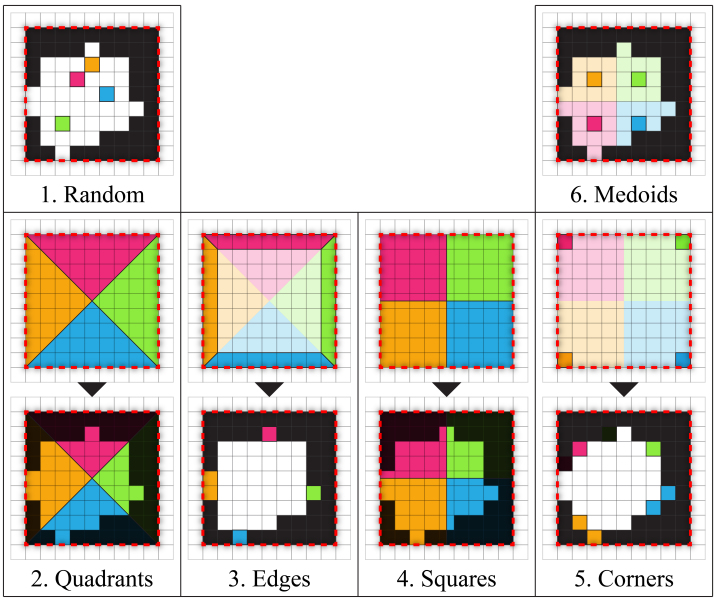


图 1：对可用 DAS 方法的可视化表示，将其应用于包含 9×9 个单元格的选择窗口，假设其底层覆盖率如图中白色区域所示。如果有多个单元格颜色相同，则从这些同色单元格中随机选择一个候选项。

如果某个后代落入已被占用的单元，则适应度最高的个体存活。

3.3 设计备选方案采样方法

如第 3.2 节所述，算法会从选择窗口内采样若干设计备选方案并呈现给用户。本文仅测试包含四个备选方案的 UC-ME，并考察了六种设计备选方案采样（DAS）方法，这些方法在不同程度上都具有随机性。以下列表介绍了每种 DAS 方法；图 1 展示了一个假设的 DAS 示例。

- **随机**（A_R）从窗口中随机抽取 4 个精英个体。
- **象限**（A_Q）和**正方形**（A_S）使用对角线（A_Q）或x-y 轴（A_S）将窗口划分为 4 个相等的部分。每个部分中随机抽取一个个体。
- **边缘**（A_E）从窗口的 4 条边各随机抽取一个个体。如果边缘上没有个体，则优先选择距离该边缘最近的个体。
- **角点**（A_C）从窗口的每个角点抽取一个个体，或者选择距离该角点最近的个体（如 A_E 所示）。
- **中心点**（A_M）：将选择窗口内个体的坐标作为数据点，输入k-中心点聚类算法，其中本文中 $k = 4$。向用户展示这些聚类的四个中心点。

4 用例：布局生成

我们在生成建筑布局这一复杂且受约束的问题上测试 UC-ME。这项任务的变体见于计算建筑学 [33, 34, 35] 和游戏 [36, 37]。我们沿用 [8], 的方法：问题定义为一组拓扑约束及其他约束，输出则是满足这些约束的几何解。下面概述此用例的流程；更多详情请参见 [8]。

我们选择聚焦于这一复杂问题，原因有二：首先，建筑布局具有许多可量化但又带有主观性的特征，因此非常适合用于测试 MI-CC 方法。其次，这项任务提供了一个在受限领域内测试所提方法的机会，展示了其向其他 MAP-Elites 变体扩展的能力。

4.1 布局表示

建筑布局的表示包含两个方面：设计规范（设计空间）和设计实现（DI）。

设计空间是对当前问题的用户定义描述。它包含布局的连通图，其中顶点表示空间单元（房间或其他区域），边表示两个空间单元之间通过门形成的连接（例如见图2）。它还包括每个空间单元所需的面积、每个空间单元通向外部的门和窗户数量，以及该空间是室内空间还是室外空间（例如见表1）。

DI 是设计空间的具体几何实现，其中每个空间单元都占据平面上的特定区域，并具有特定形状。设计空间还包括门和窗户的精确位置。为了支持自由形态几何图形的生成（即不受网格等固定几何系统的限制），我们实现了一个基于平面 Voronoi 镶嵌的系统。生成的空间单元可以放置在该镶嵌的特定区域中。在算法运行过程中，空间单元的放置位置和底层结构都会发生变异。

4.2 约束条件

建筑布局是受到诸多约束的问题，既受物理要求限制，也受设计空间（DS）中的设计约束限制。若满足以下条件，我们便认为布局是可行的：(a) 每个空间单元都占据一个独立且连通的区域；(b) 设计空间中的相邻空间单元共享一条边界；(c) 每个空间单元的面积与设计空间中指定的面积足够接近；(d) 所有规定的门和窗户均已正确放置；(e) 所有室内通道的宽度至少为 0.5 米；(f) 整个 Voronoi 图是连通的；(g) 至少 50% 的 Voronoi 单元格不接触边界。后两项是引擎底层约束，用于确保能够生成空间单元。系统会根据这些约束逐一检验每个生成的个体；若个体未能满足其中任一项，则会根据其满足的约束数量及满足程度，赋予相应的可行性评分。

4.3 适应度

在这个建筑布局问题中，质量主要通过满足约束来保证。任何满足第 4.2 节约束的个体，最终都可视为“足够好”的布局。然而，为了引导 QD 算法，我们将对设计空间（DS）的遵循程度作为可行种群的主要质量标准，并优化各空间单元的面积与指定面积之间的接近程度。我们将平均面积精度（P_s）定义为每个空间单元的指定面积与实际面积之差的平均值。对于每个空间单元，其面积精度通过公式（1）计算；如果设计空间（DS）中的某个空间单元在设计实例（DI）中缺失，则其 P_s 为 0。请注意， P_s 也是判断可行性的标准：确实，如果 $P_s < 0.6$ ，则该个体不可行。如果 $P_s \geq 0.6$，则将此指标作为可行档案的质量表征。

$$P_s = \begin{cases} A/A_t & A < A_t \\ A_t/A & A_t \leq A \end{cases} \tag{1}$$

其中， A 是生成的空间单元的面积， A_t 是设计空间（DS）中为该空间单元规定的目标面积。

4.4 行为表征

我们使用两种行为表征（BCs）来衡量生成的设计实例（DIs）的多样性。这些行为表征用于可行档案库和不可行档案库，其定义如下：

空间单元平均紧凑度 (C_s)基于紧凑度这一概念。紧凑度是一种无量纲指标，用于表示形状的周长与面积之间的关系 [38]。单个空间单元的紧凑度（C_s）按式（2）计算；如果设计实例（DI）中定义了某个空间单元，但该空间单元在设计空间（DS）中缺失，则其 $C_s = 0$。最后， C_s 用于衡量设计实例（DI）中所有空间单元的平均紧凑度。

$$C_s = 2\pi A/\Pi^2 \tag{2}$$

其中， A 为空间单元的表面积， Π 为其周长。

平面正交度(O_θ) 的计算方法是：取布局中相连墙体之间所有角度的正交度（θ）的平均值。单个角度的正交度（O_θ）按公式 (3) 计算；对于锐角， O_θ 从 0 线性增加到 1。对于钝角， $\theta \in [\pi/2, 3\pi/4]$, O_θ 从 1 线性降低到 0。5。对于钝角，
$\theta \in [3\pi/4, \pi]$,

O_θ 从 0。5 线性增加到 1。

$$O_\theta = \begin{cases} 2\theta/\pi & 0 \le \theta < \pi/2 \\ 2 - 2\theta/\pi & \pi/2 \le \theta < 3\pi/4 \\ 2\theta/\pi - 1 & 3\pi/4 \le \theta \le \pi \end{cases} \tag{3}$$

其中 θ 表示两个连续线段之间的无符号角，范围始终为 $[0 \dots \pi$ $]$.

4.5 初始生成

布局的初始生成以半随机方式进行，因此无法保证其可行性。

首先，在边界矩形内随机放置一组点，并据此对平面进行 Voronoi 剖分；随后，从连接数最多的空间单元开始，逐步将设计空间中规定的空间单元放置在平面上；为每个空间单元分配与设计空间中规定面积相符的单元格数量。然后，在可行的情况下，在每个房间的墙上随机设置规定的开口。

4.6 遗传算子

布局的变异分为两个阶段：随机破坏和脚本化修复。这种方法生成的后代与父代部分相似（因为只会破坏布局的部分区域），而修复函数旨在降低生成不可行后代的概率。

在破坏步骤中，算法会从以下随机操作中选择并执行 1 到 3 项：(1) 沿同一方向移动底层 Voronoi 网格的所有点；(2) 沿随机方向移动一小组 Voronoi 点；(3) 删除整个房间；(4) 扩展现有房间，可以从其他房间占用单元格（不安全扩展），也可以避开其他房间（安全扩展）；(6) 腐蚀房间的外部单元格，同时保持连通性；(7) 删除部分现有开口。

在修复步骤中，依次执行以下操作：(1) 在空间足够时添加缺失的房间；(2) 重新连接可能已断开的同一房间的各个片段；(3) 扩展房间，以满足与其他房间的相邻要求；(4) 删除或添加房间片段，直到其面积与 DS 中的目标面积足够接近；(5) 删除多余的开口并设置缺失的开口。

4.7 受约束的质量多样性流程

由于该问题具有受约束的特性，我们借鉴受约束 FI-MAP-Elites [8] 算法，对 UC-ME 的一般操作（第 3 节）进行了修改。

受约束的 UC-ME 采用双档案库方案：一个档案库存放可行精英个体，另一个存放不可行个体。我们采用第 4.4 节所述的两个行为表征，并将其应用于两个档案库。对于不可行存档，我们根据个体的可行性评分（见第 4.2 节）为精英个体赋予质量；对于可行档案，我们则将质量设为 $_P_s$。我们按如下方式修改第 3.1 节中的初始化过程：根据可行性，将 100 个初始个体分配到可行或不可行档案库。之后，我们运行 FI-MAP-Elites 质量多样性搜索，直到可行档案的覆盖率至少达到 1%。这样便能获得足够的种子个体，以运行算法的交互式操作；就该操作而言，唯一的变化是：(a) DAS方法仅应用于可行档案；(b) 选择窗口同时应用于可行档案和不可行存档；(c) 父代在可行档案和不可行存档之间交替选取。与第 3.2 节的基本 UC-ME 一样，我们会检验后代是否可行，并计算其行为表征，然后将其放入相应档案库的相应单元中；如果后代的可行性更高（对于不可行存档）或 P_s 评分更高（对于可行档案），则替换该单元中已有的精英个体 $^-$。

5 实验方案

作为建筑布局生成的一个具体案例研究，我们针对一套中等大小公寓的特定设计规范（第5.1节）、算法参数（第5.2节）以及用于测试算法的可控（人工）用户（第5.3节）开展了一项实验。我们还确定了大量性能指标，用于评估该算法的整体效能和针对用户的效能，相关指标详见第5.4节。

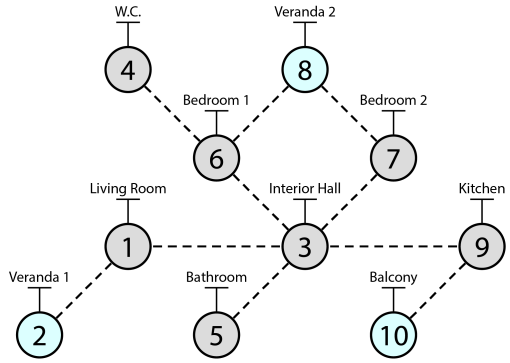


图 2：设计空间中空间单元之间的连接 室内空间单元以灰色显示，外部空间单元以青色显示 .

ID	Name	Type	Area	入口	窗户
1	客厅	室内	$40m^2$	1	2
2	阳台1	外部	$25m^2$	0	0
3	室内走廊	室内	$5m^2$	0	0
4	W.C.	室内	$4m^2$	0	1
5	浴室	室内	$6m^2$	0	1
6	卧室1	室内	$15m^2$	0	0
7	卧室2	室内	$12m^2$	0	0
8	阳台2	外部	$15m^2$	0	0
9	厨房	室内	$14m^2$	0	1
10	阳台	外部	$4m^2$	0	0

表 1：实验中纳入 DS 的空间单元。

5.1 公寓布局规范

本文采用一套典型地中海式公寓的设计空间。该设计空间由10个空间单元组成，如表1所述。它们之间规定的连接关系如图2所示。

5.2 算法参数

在本实验中，我们将每个档案库划分为 4,096 个单元格（64×64）。行为表征、可行和不可行适应度与第 4 节中的定义相同。选择窗口的大小为 9×9。算法以 100 个个体进行初始化，FI-MAP-Elites 运行至覆盖可行档案的 1% 为止。所有实验中的所有运行均从上述生成的同一初始种群开始。在用户选择之间，UC-ME 在选择窗口内运行，通过迭代选择 10,000 个父代（在可行档案和不可行存档之间交替选择），随后再向用户展示下一批设计备选方案。

5.3 人工用户

为了测试交互式质量多样性方法的性能，我们需要一种控制用户选择标准的方法。因此，我们开发了若干人工用户：每位人工用户（U_i）在从所呈现的备选方案中进行选择时，都会遵循一种启发式规则（h_i）作为其用户选择准则（USC）。所有智能体都会选择h_i 最高的个体（最大化问题），并且所有 $h_i \in [0, 1]$。其中，八个人工智能体（U_1 至 U_8）在整个进化过程中始终采用一致的用户选择准则，而另外四个人工智能体（U_9 至 U_{12}）会在进行5次选择（s）后改变其用户选择准则，以测试算法如何适应不断变化的用户偏好。这种通过人工用户（包括会改变用户选择准则的人工用户）评估依赖个人偏好的算法的方法基于 [5]。我们选择了能够由案例研究中的两种行为表征（BC1和BC2）刻画的用户选择准则，因为当用户偏好与所探索的质量多样性维度并非正交时，所提出的方法效果最佳。不同用户所遵循的启发式规则如下：

- U_1 最大化 BC1：$h_1 = C_s$。
- U_2 最大化 BC2：$h_2 = O_\theta$。
- U_3 最大化 BC1 和 BC2：$h_3 = \frac{1}{2}(\bar{C}_s + \bar{O}_\theta)$。

- U_4 最大化 BC1 或 BC2：$h_4 = max(\bar{C}_s, \bar{O}_\theta)$。
- U_5 最小化 BC1：$h_5 = 1 - \bar{C}_s$。
- U_6 最小化 BC2：$h_6 = 1 - \bar{O}_\theta$。
- $U_7$5 同时最小化 BC1 和 BC2：$h_7 = 1 - \frac{1}{2}(\bar{C}_s + \bar{O}_\theta)$。

- U_8 最小化 BC1 或 BC2：$h_8 = 1 - max(\bar{C}_s, \bar{O}_\theta)$。
- U_9 最大化 BC1，然后将其最小化：

$$h_9 = \begin{cases} \bar{C}_s \text{ 如果 } 1 - \bar{C}_s & \text{如果 } s > 5 \end{cases}$$

- U_{10} 最大化 BC2，然后最小化 BC2：

$$h_{10} = \begin{cases} \bar{O}_\theta \text{ 如果 } 1 - \bar{O}_\theta & \text{如果 } s > 5 \end{cases}$$

- U_{11} 最大化 BC1，然后最大化 BC2：

$$h_{11} = \begin{cases} \bar{C}_s \text{ 如果 } \bar{O}_\theta & s > 5 \text{ 如果} \end{cases}$$

- U_{12} 最大化 BC2，然后最大化 BC1：

$$h_{12} = \begin{cases} \bar{O}_\theta \text{ 如果 } \bar{C}_s & s > 5 \text{ 如果} \end{cases}$$

5.4 性能指标

我们遵循相关文献，通过观察可行档案库 [7]的覆盖率（已占用单元格的百分比）、最大适应度（精英个体中的最高适应度）和质量多样性分数（所有精英个体适应度的总和），来评估 MAP-Elites 算法的性能。具体而言，我们考察这些指标从进化开始到结束的曲线下面积（AUC），从而衡量整个运行过程中的性能，而不仅仅是最终状态。

由于我们希望实现用户可控的问题空间探索，因此采用多种形式的用户选择准则（USC）来评估算法对用户偏好的契合程度。以下指标用于衡量精英个体是否符合用户的选择标准：档案库中精英个体的 USC 最大值和平均值（最大 USC、平均 USC）。此外，我们还希望评估按照 USC 衡量表现良好的精英个体，其质量是否也较高。为此，我们将可行精英个体的 USC 分数与其适应度相乘，得到加权USC（W-USC）。受 QD 分数启发，我们同时评估档案库中的平均加权USC（平均加权 USC），以及可行精英个体的加权USC总和（加权 USC 总和）。这四项指标均在算法的每个时间步进行评估，并与上文类似，计算各自的 AUC 分数。

鉴于第 3.3 节介绍了许多 DAS方法，评估这些设计备选方案对用户有多大用处十分重要。因此，我们引入以下指标：局部多样性，即同时呈现的备选方案在特征空间中的平均距离；局部平均适应度，即它们的平均适应度；局部平均 *USC*，即它们的平均 USC 值。与上文相同，使用这些指标的 AUC 来比较不同 DAS方法。

最后，DAS 方法会影响选择窗口到达 USC 分数较高的搜索空间区域的效率，而不是随机移动。我们通过公式 (4) 评估*USC* 效率；它主要反映连续的设计者选择使 USC 高于此前所获得水平的程度。与其他所有指标不同，效率衡量的是整个进化过程，因此无需像 AUC 那样随时间记录。

$$\text{USC Efficiency} = \frac{\sum_{s=2}^{N_s}(U_s - U_{s-1})}{\sum_{s=2}^{N_s}|U_s - U_{s-1}|} \tag{4}$$

其中，N_s 是实验中的用户选择总数，U_s 是第 s 次选择中所选个体的 USC 值。

6 结果

为总结我们基于第4节用例开展实验所得的发现，我们首先比较采用不同DAS方法的UC-ME结果（第6.1节），然后从中选出最有前景的DAS方法，与不含用户控制的MAP-Elites进行比较（第6.2节）。结果来自10次独立运行，显著性通过Student's t-检验确定，$p < 0.05$。

6.1 DAS 方法比较

参数	A_R	A_Q	A_E	A_S	A_C	A_M
局部多样性	0	0	48	0	58	13
局部平均适应度	17	2	0	2	0	31
局部平均用户选择准则	0	2	2	1	7	0
用户选择准则效率	0	10	10	10	7	6

表 2：所有 DAS方法在局部质量多样性指标上的比较。数值表示在同一实验中，该 DAS方法显著优于另一种 DAS方法的次数（$p < 0.05$）。结果在进行 10 次选择后汇总。

表2总结了不同DAS方法之间的比较。我们将每个人工用户视为一次独立实验，评估在10次选择后，某种DAS方法的指标显著优于另一种方法的成对比较次数。共有6种DAS方法（即每种方法有5种比较）和12个人工用户，因此每个单元格中的最大值为60。对于多重比较，采用了Bonferroni校正 [39] 。请注意，我们还测试了所有其他指标（涉及QD和用户选择准则），但不同DAS方法之间几乎没有差异。
表2表明，Edges和Corners DAS方法在局部多样性方面具有明显优势。这是意料之中的，因为这两种方法都优先选择距离选择窗口中心尽可能远的个体。直观而言，Corners方法在局部多样性方面略胜一筹，因为它最先选择的个体与窗口内所有候选个体相比具有绝对最大距离（见图1）。

从局部平均适应度来看，表现最好的方法是中心点，其次是随机 DAS。直观而言，这两种方法都不太可能迅速将窗口移离当前位置（这也体现在它们较差的效率上），因此更多的 QD 迭代会针对相同的单元格，从而可能提高窗口内的适应度。此外，可行空间边缘处的个体往往适应度较低（见图 4），因此边缘或 Corners DAS 方法很可能会采样到至少一些适应度较低的个体。
我们还注意到，就局部平均 USC 和 USC 效率而言，没有哪种方法明显胜出； A_E、 A_S 和 A_M的效率略高于其他方法。甚至在选择次数达到 10 次之前， A_C 就往往会到达可行空间中 USC 最佳的边缘（见图 4）；此后，窗口的移动变得反复无常，效率也很低。
根据表 2 的结果，Corners DAS 方法表现最佳，因为它展示的个体具有更高的多样性，同时效率也相当不错。与其他 DAS方法相比，Medoids 方法向用户展示的个体适应度更高，同时在根据 USC 进行调整方面也有一定效率。因此，我们将在下一节中对这些 DAS 方法与基线 MAP-Elites 进行测试比较。

6.2 与 MAP-Elites 的比较

参数	MAP-Elites	A_C	MAP-Elites	A_M
覆盖率	12	0	12	0
最大适应度	0	0	0	0
QD 分数	12	0	12	0
最大 USC	1	9	1	4
平均 USC	0	12	0	10
平均加权 USC	0	11	0	11
加权 USC 总和	11	0	12	0

表3：基线 MAP-Elites 与表现最佳的两种 DAS 方法的比较，显示在每项指标上，一种方法的得分显著高于另一种方法的实验次数（$p < 0.05$）。结果是在经过10次选择后汇总的。

根据第6.1节中对 DAS 方法的比较，我们重点将 Corners 和 Medoids 方法与基线 MAP-Elites 进行比较。基线 MAP-Elites 不考虑用户偏好，而是在搜索空间中进行无引导探索。该基线实现了 FI-MAP-Elites [8] ，并通过在两个档案库之间交替，随机选择个体进行变异。

表 3 展示了无引导的 MAP-Elites 基线与各个 UC-ME 变体之间的两两比较。结果显示，在 10 次用户选择后，基线或 UC-ME 变体在所选指标上表现更优的实验次数（共 12 次实验）。显然，在所有实验中，无引导的 MAP-Elites 对问题空间的覆盖率更高，因此 QD 分数也更高。这并不令人意外，因为

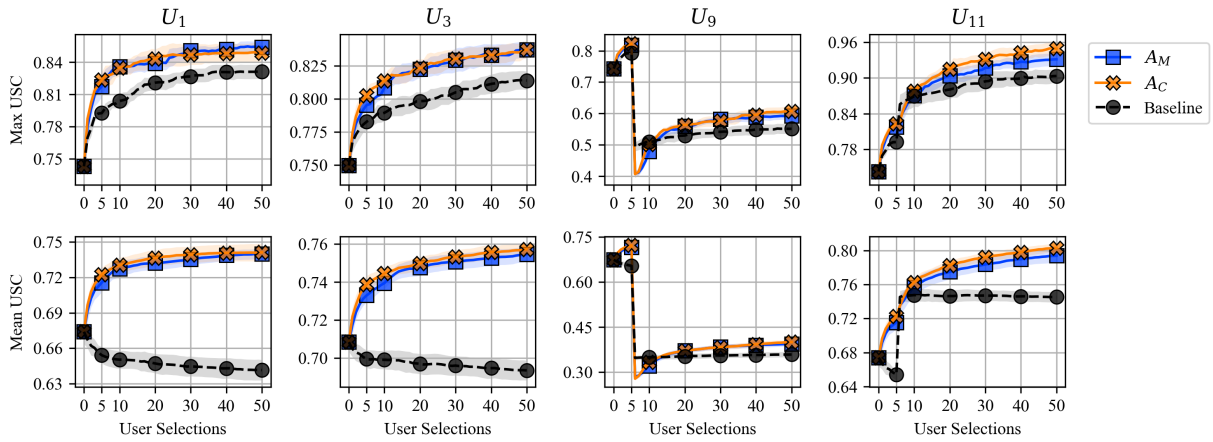


图3：图表展示了四种不同人工用户（U_1、 U_4、 U_9 和 U_{11}）的最大值和平均 USC，并比较了象限（A_C）与中心点（A_M）DAS方法和基线（不含用户控制的 MAP-Elites）。数值取10次不同运行的平均值，阴影区域表示95%的置信区间。

UC-ME 会将搜索引导至问题空间的特定部分（以及特征图的特定区域），而 MAP-Elites 则尽可能覆盖整个特征图。我们还注意到，在最大适应度方面没有差异。这多少令人意外，因为特征图的不同部分（由不同用户指定目标）可能并非都具有同样高的适应度。在这一用例中，找到一个适应度很高的个体似乎并不困难。

正如预期，无引导探索的基线 MAP-Elites 在档案库中所有精英个体的最大 USC 分数和平均 USC 分数方面，表现均不如两个 UC-ME 版本。与 A_C相比， A_M 方法在达到极高 USC 分数方面效率较低；这并不令人意外，因为后者能更快地将选择窗口移向问题空间中 USC 较高的区域。主要由于平均 USC 较高，UC-ME变体的平均加权 USC高于基线也并不令人意外。然而，基线更高的覆盖率使所有精英个体的加权USC分数总和更高，这与 QD 分数的情况类似。

图3展示了四位具有代表性的人工用户在平均 USC 和最大 USC 变化趋势上的对比：其中两位始终如一（U_1、U_3），另外两位在进行 5 次选择后改变了准则（U_9、 U_{11}）。请注意，在整个第6节中，我们选择关注 10 次选择这一阈值，因为在用户疲劳开始显现之前，用户与此类工具交互 10 次是一个现实的次数。不过，为了完整起见，我们让实验持续运行，直到完成 50 次用户选择，并在图3中展示了完整的运行结果。

从图3可以明显看出，尽管无引导的 MAP-Elites 可能会偶然找到用户选择准则较高的问题空间区域（即最大 USC 持续增加），但这并不意味着更广泛的群体也能获得这一结果（平均 USC 可能增加，也可能降低），具体取决于空间中哪些部分更容易到达。正如预期的那样，随着档案库在局部质量多样性的引导下逐渐聚焦于空间的特定部分，UC-ME变体持续改善这两项 USC 指标。

图3还展示了算法如何处理第 5 次选择后用户标准的突然变化（U_9, U_{11}）。 U_9 的变化更加突然，因为它突然转而追求与先前标准相反的目标，导致 USC 下降。需要经过数次用户交互，才能将选择窗口移至问题空间中更合适的区域；但从标准改变后再进行 5 次选择，UC-ME 的表现便接近持续演进的基线的平均 USC 和最大 USC，该基线一直在追求 的高 C_s 和低 C_s （两者均在 h_9中呈现）。给定足够的时间，两种方法都会超越基线（例如，选择 25 次后）。 U_{11} 改变目标的态度没有那么“激进”；事实上，即使是不受引导的 MAP-Elites 也能找到较高的个体 O_θ，也就是从第 6 次选择开始的 USC。由于选择窗口不必像 U_9那样折返重走，UC-ME 方法再经过 5 次选择后就能找到与基线相当或略优的个体，而给定足够的时间则能找到好得多的个体。

UC-ME 的进展也可以通过特征图本身来可视化。图4展示了每次用户选择后覆盖率的变化（对于 MAP-Elites，则展示相同评估阈值下的变化）。此外，图中用红色标出 UC-ME 的选择窗口，随着其向更高的 USC 分数移动（本例中为 U_3）。我们重点关注 A_C方法，因为它效率最高。图4顶排的图像展示了 UC-ME 与 MAP-Elites 探索模式之间的差异：灰色表示两种方法共同发现的单元格，洋红色表示仅由 MAP-Elites 发现的单元格，蓝色表示仅由 UC-ME 发现的单元格。我们可以看到，较高 USC 值对应的单元格完全属于 UC-ME。MAP-Elites 覆盖率较高，是因为大多数单元格占据了

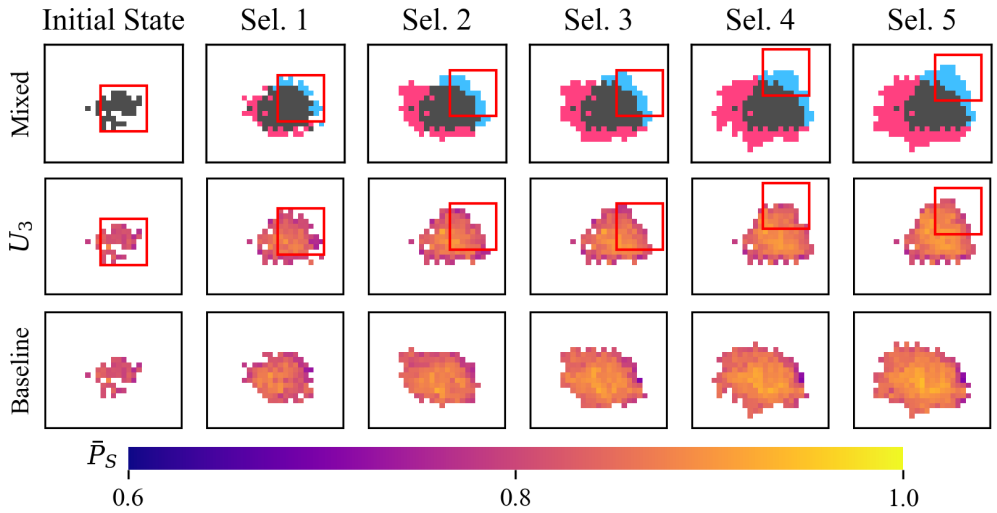
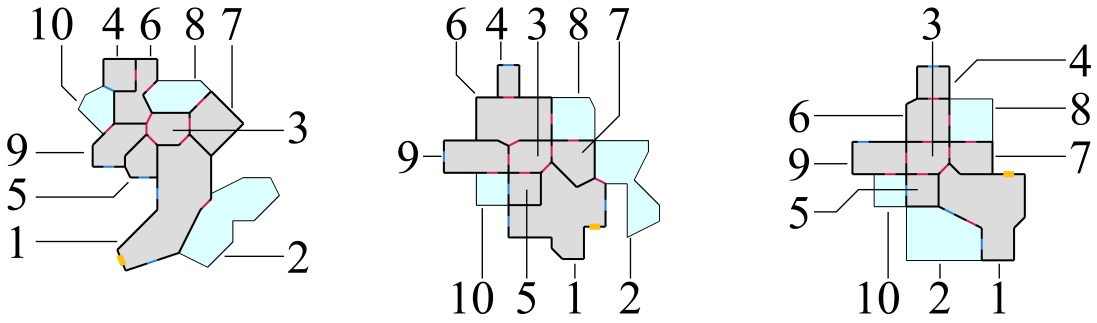


图4：基线 MAP-Elites（底行）和由 U_3 引导的、采用 Corners DAS 的 UC-ME（中行）在前 5 次选择中的行为空间探索。底部显示的共享色阶表示适应度 $P_s \in [0.6, 1]$。顶行显示覆盖率差异：红色单元格仅由基线发现，蓝色单元格仅由 UC-ME 发现，灰色单元格则为两者共有。在这些图中，x 轴为 $C_s \in [0.44, 0.86]$，y 轴为 $O_\theta \in [0.61, 0.97]$。



（a）选择 1　（b）选择 10　（c）选择 50

图 5：使用 A_C DAS 方法，在与 UC-ME 交互 1 次、10 次和 50 次后，由 U_3 选择的代表性个体。

较低的 C_s 和 O_θ 值对于 U_3来说并不理想。图4还展示了选择窗口如何首先朝着更高的 C_s移动；一旦到达可行空间的边缘，无法在那个方向找到具有更高分数的个体，它就会转而朝着更高的 O_θ 分数移动。我们还看到，在前3次选择中，带有 A_C 的UC-ME已经找到了具有最高 USC 分数的可行空间边缘，并开始在该附近相当随意地移动，从而在问题空间的特定区域中进行更多选择，并提升个体的质量。

最后，我们在图 5 中展示了 U_3 在一次具有代表性的 UC-ME 运行中使用 A_C 所做的选择。我们观察到，初始个体的 USC分数并不理想，因为所示选择中的房间包含许多锐角和复杂形状。经过 10 次选择后，用户找到了一个房间大多紧凑且呈方形的个体。经过 50 次选择后，结果与 10 次选择时相比没有太大差异；因此，对于这一问题，通常 10 次选择就足够了，也不会让用户过于疲劳。

7 讨论

本文认为，MAP-Elites 的空间划分有益于交互式进化：移动的选择窗口是一种简单直观的方式，可将质量多样性搜索限定在局部，并将窗口（以及行为表征）视为隐式函数

近似器 [2]。我们的实验表明，尽管无引导的 MAP-Elites 探索的问题空间更大，UC-ME 找到的个体更符合用户偏好，并覆盖了可行空间中相当大的、具有理想特征的区域（见图 4）。有趣的是，使用人工用户进行的实验并未揭示 DAS 方法之间存在显著差异；不过，我们预计，展示不同的备选方案可能会对人类用户产生更大影响，使其选择更加困难并加剧疲劳。

因此，一个直观的下一步是将 UC-ME 部署给具有建筑布局经验的人类设计者，并评估第 5.4 节所列的性能指标，以及用户在研究结束后通过可用性问卷反馈的体验 [40]。与人工用户不同，建筑布局这一复杂问题可能难以由人类用户处理，同时他们还需要在相似程度不同的选项之间作出选择。因此，未来工作应探索 UC-ME 在无约束（或约束较少）的问题中的表现，包括那些需要人类用户花费较少认知精力、便于通过视觉处理结果的问题。一个直观的无约束案例研究方向是图像生成，MAP-Elites 已应用于这一领域 [41]。未来工作的其他方向可以探索游戏等受约束领域，在这些领域中，质量多样性搜索有助于为游戏设计师提供多样且优质的选项[24]；游戏领域中认知负担较轻的任务可能包括精灵图生成 [42] 或叙事结构 [43]。

除了在其他领域和人类用户中进行测试之外，还有大量未来方向可用于改进 UC-ME 的算法属性。具体而言，实现一种在更高维特征空间（超过 2 个维度）中运行的变体将是值得采取的一步，因为这将为用户提供更多搜索方向上的控制权。这种方法可以利用 CVT-MAP-Elites [44], ，该方法理论上能够在更多维度上运行，并将选择窗口转换为在该空间中移动的点云（由最近的 CVT 邻居构成）。一条更具雄心的未来研究路径，是将用户模型 [15] 重新引入 UC-ME，作为局部质量多样性的（审美）函数逼近 [1] 的补充方式。此外，不断更新的用户模型 [4] 还有可能用于调整行为表征本身，使其更好地符合用户偏好，从而实现更简单的 UC-ME 流程，仅需在空间中移动选择窗口。这一雄心勃勃的方向在 UC-ME 之外也可能具有多种应用，并且可以利用现有的自动增量行为表征方法 [45, 46, 47]。

8 结论

本文提出了一种以较低认知负担控制质量多样性搜索的探索方向和计算预算的方法。用户控制的 MAP-Elites 算法（UC-ME）是交互式进化范式（狭义上）首次应用于 MAP-Elites 的实例，这与以往工作不同 [32]。UC-ME 算法通过将父代的选择集中在特征图中较小的窗口内来运行，因此能够通过两个档案库处理无约束和有约束问题 [8]。在针对一个复杂且受到严格约束的问题所开展的实验中，我们观察到，UC-ME 能够根据用户的选择标准，将搜索集中于问题空间中感兴趣的部分，但代价是覆盖率较低、精英个体总数较少。所提出的方法展现出应用潜力，但接下来的重要工作包括：让真实用户参与测试其效率及对用户疲劳的影响；将研究拓展到更多领域和更多行为特征（BC）维度；并结合持续更新的用户偏好模型开展研究 [15]。

9 致谢

本研究获得欧盟地平线2020研究与创新计划中 PrismArch（资助协议编号：952002）和 AI4media（资助协议编号：951911）项目的支持。

参考文献

[1] Yan Pei 和 Hideyuki Takagi。交互式进化计算研究进展综述。《环境智能与人性化计算期刊》，第 1–14 页，2018 年。[2] Hideyuki Takagi。交互式进化计算：融合进化计算优化能力与人工评估。《*IEEE*会议录》，89(9):1275–1296，2001 年。[3] Antonios Liapis、Georgios N. Yannakakis 和 Julian Togelius。调整视觉美学模型以创建个性化内容。《*IEEE*计算智能与游戏人工智能汇刊》，4(3):213–228，2012 年。[4] Alexander Hagg、Alexander Asteroth 和 Thomas Bäck。对质量多样性中的用户选择进行建模。载于遗传与进化计算会议论文集，2019 年。

[5] Antonios Liapis、Héctor P. Martínez、Julian Togelius 和 Georgios N. Yannakakis。通过基于排名的交互式进化创建自适应游戏关卡。载于 IEEE 计算智能与游戏会议论文集，2013 年。[6] Justin K Pugh、Lisa B Soros 和 Kenneth O Stanley。质量多样性：进化计算的新前沿。《机器人与人工智能前沿》，2016 年。[7]Jean-Baptiste Mouret 和 Jeff Clune。通过映射精英个体照亮搜索空间。arXiv 预印本 arXiv:1504.04909，2015 年。[8] Konstantinos Sfikas、Antonios Liapis 和 Georgios N. Yannakakis。用于房间式环境的通用表达算法。载于 FDG 游戏程序化内容生成研讨会，2022 年。[9] Jimmy Secretan、Nicholas Beato、David B. D'Ambrosio、Adelein Rodriguez、Adam Campbell、Jeremiah T. Folsom-Kovarik 和 Kenneth O. Stanley。Picbreeder：设计空间协作式进化探索的案例研究。《进化计算》，19(3):373–403，2011 年。[10] Jeff Clune、Anthony Chen 和 Hod Lipson。上传任意物体并使其进化：将复杂几何图案注入 CPPN 以供进一步进化。载于 IEEE 进化计算大会论文集，第 3395–3402 页，2013 年。[11] Björn Þór Jónsson、Amy K. Hoover 和 Sebastian Risi。交互式进化组合式声音合成网络。载于遗传与进化计算会议论文集，第 321–328 页，2015 年。[12] Sebastian Risi、Joel Lehman、David B. D'Ambrosio、Ryan Hall 和 Kenneth O. Stanley。Petalz：面向休闲玩家的基于搜索的程序化内容生成。《IEEE计算智能与游戏人工智能汇刊》，8(3):244–255，2016 年。[13] Erin J. Hastings、Ratan K. Guha 和 Kenneth O. Stanley。在《银河军备竞赛》电子游戏中进化内容。载于 IEEE 计算智能与游戏研讨会论文集，第 241–248 页，2009 年。[14] Terence Soule、Samantha Heck、Thomas E. Haynes、Nicholas Wood 和 Barrie D. Robison。达尔文的恶魔：进化会改善游戏吗？载于《进化计算应用》论文集，第 435–451 页，2017 年。[15] Antonios Liapis、Georgios N. Yannakakis 和 Julian Togelius。为个性化游戏内容创作工具建立设计者模型。载于人工智能与游戏美学研讨会论文集，2013 年。[16] Antonios Liapis、Georgios N. Yannakakis 和 Julian Togelius。为 Sentient Sketchbook 建立设计者模型。载于 IEEE 计算智能与游戏会议论文集，2014 年。[17] Yan Pei 和 Hideyuki Takagi。利用在投影到一维空间后获得的精英个体加速进化计算。载于遗传与进化计算会议论文集，第 89–92 页，2011 年。[18] Yan Pei 和 Hideyuki Takagi。通过傅里叶分析适应度景观以加速进化搜索。载于 IEEE 进化计算大会论文集，2012 年。[19] Hideyuki Takagi 和 Denis Pallez。基于成对比较的交互式差分进化。载于世界自然与生物启发计算大会论文集，第 475–480 页，2009 年。[20] Yan Pei 和 Hideyuki Takagi。基于三重和四重比较的交互式差分进化与差分进化。载于遗传算法基础研讨会论文集，第 173–182 页。ACM，2013 年。[21] Yan Pei 和 Hideyuki Takagi。交互式差分进化与差分进化中，基于成对比较的模因搜索所形成的局部适应度景观。《国际自组网与泛在计算期刊》，2017 年。[22] Noboru Murata、Ryuei Nishii、Hideyuki Takagi 和 Yan Pei。种群收敛点的解析估计。载于 IEEE 进化计算大会论文集，第 2619–2624 页，2015 年。[23] Joel Lehman 和 Kenneth O Stanley。通过新颖性搜索和局部竞争，使虚拟生物的多样性不断进化。载于遗传与进化计算会议论文集，第 211–218 页，2011 年。[24] Daniele Gravina、Ahmed Khalifa、Antonios Liapis、Julian Togelius 和 Georgios N. Yannakakis。通过质量多样性实现程序化内容生成。载于 IEEE 游戏会议论文集，2019 年。[25] Antonios Liapis、Héctor P Martınez、Julian Togelius 和 Georgios N Yannakakis。通过 DeLeNoX 转化探索式创造力。载于计算创造力国际会议论文集，第 56–63 页，2013 年。

[26] Ahmed Khalifa、Scott Lee、Andy Nealen 和 Julian Togelius。《Talakat：通过受约束的 MAP-Elites 生成弹幕游戏》。载于《遗传与进化计算会议论文集》，第 1047–1054 页，2018 年。[27] Alberto Alvarez、Steve Dahlskog、Jose Font 和 Julian Togelius。《通过交互式受约束 MAP-Elites 推动地牢设计中的质量多样性》。载于《IEEE 游戏会议论文集》，2019 年。[28] Theodoros Galanos、Antonios Liapis、Georgios N. Yannakakis 和 Reinhard Koenig。《Arch-elites：城市设计中的质量多样性》。载于《遗传与进化计算会议论文集》，第 313–314 页，2021 年。[29] Antonios Liapis、Georgios N Yannakakis 和 Julian Togelius。《受约束的新颖性搜索：游戏内容生成研究》。《进化计算》，23(1):101–129，2015 年。[30] Steven Orla Kimbrough、Gary J Koehler、Ming Lu 和 David Harlan Wood。《关于用于约束优化的可行–不可行双种群（FI-2pop）遗传算法：距离追踪与无免费午餐》。《欧洲运筹学杂志》，190(2):310–327，2008 年。[31] Antonios Liapis、Gillian Smith 和 Noor Shaker。《混合主动式内容创作》。载于 Noor Shaker、Julian Togelius 和 Mark J. Nelson 编，《游戏中的程序化内容生成：教材与当前研究综述》，第 195–214 页。施普林格，2016 年。[32] Alberto Alvarez、Jose Maria Maria Font Fernandez、Steve Dahlskog 和 Julian Togelius。《交互式受约束 MAP-Elites：特征维度表达能力的分析与评估》。《IEEE 游戏汇刊》，14(2)，2020 年。[33] Philippe Charman。《一种用于生成平面图的基于约束的方法》。载于《IEEE 人工智能工具会议论文集》，第 555–561 页，1994 年。[34] Benachir Medjdoub 和 Bernard Yannou。《空间规划中的拓扑与几何分离》。《计算机辅助设计》，32(1):39–61，2000 年。[35] Eugénio Rodrigues、Adélio Rodrigues Gaspar 和 Álvaro Gomes。《一种结合局部搜索技术的进化策略，用于建筑学中的空间分配问题，第一部分：方法论》。《计算机辅助设计》，45(5):887–897，2013 年。[36] Daniel Karavolos、Antonios Liapis 和 Georgios N Yannakakis。《进化任务以创建游戏空间》。载于《IEEE 计算智能与游戏会议论文集》，2016 年。[37] Joris Dormans。《关卡设计探险：为动作冒险游戏生成任务与空间》。载于《FDG 游戏程序化内容生成研讨会论文集》，2010 年。[38] Petros Koutsolampros、Kerstin Sailer、Tasos Varoudis 和 Rosie Haslem。《剖析可见性图分析：指标及其在理解工作场所人类行为中的作用》。载于《国际空间句法研讨会论文集》，2019 年。[39] Olive Jean Dunn。《均值间的多重比较》。《美国统计学会杂志》，56:52–64，2012 年。[40] James R. Lewis、Suzanne C. Henry 和 Robert L. Mack。《集成办公软件基准测试：案例研究》。载于《国际人机交互会议论文集》，1990 年。[41] Matthew Fontaine 和 Stefanos Nikolaidis。《可微质量多样性》。载于 M. Ranzato、A. Beygelzimer、Y. Dauphin、P.S. Liang 和 J. Wortman Vaughan 编，《神经信息处理系统进展》，第 34 卷，第 10040–10052 页，2021 年。[42] Antonios Liapis。《探索街机游戏素材的视觉风格》。载于《进化与仿生音乐、声音、艺术与设计论文集（EvoMusArt）》。施普林格，2016 年。[43] Alberto Alvarez 和 Jose Font。《TropeTwist：基于套路的叙事结构生成》。载于《FDG 程序化内容生成研讨会论文集》，2022 年。[44] Vassilis Vassiliades、Konstantinos Chatzilygeroudis 和 Jean-Baptiste Mouret。《使用质心 Voronoi 镶嵌扩展表型精英多维档案算法》。《IEEE 进化计算汇刊》，22(4):623–630，2018 年。[45] Antoine Cully。《通过质量多样性与无监督描述符自主发现技能》。载于《遗传与进化计算会议论文集》，2019 年。[46] Alexander Hagg、Alexander Asteroth 和 Thomas Bäck。《使用质量多样性进行原型发现》。载于《国际自然问题并行求解会议论文集》，2018 年。[47] Alexander Hagg、Sebastian Berns、Alexander Asteroth、Simon Colton 和 Thomas Bäck。《质量多样性搜索中参数化表示与数据驱动表示的表达能力》。载于《遗传与进化计算会议论文集》，2021 年。