arXiv:2512.18055v1 [cs.CG] 19 Dec 2025

BlockSets：含大型元素的集合的结构化可视化

Neda Novakova [1]*，Veselin Todorov [1]*，Steven van den Broek [1]，Tim Dwyer [2]，以及 Bettina Speckmann [1]

[1] 埃因霍温理工大学，荷兰
[2] 莫纳什大学，澳大利亚

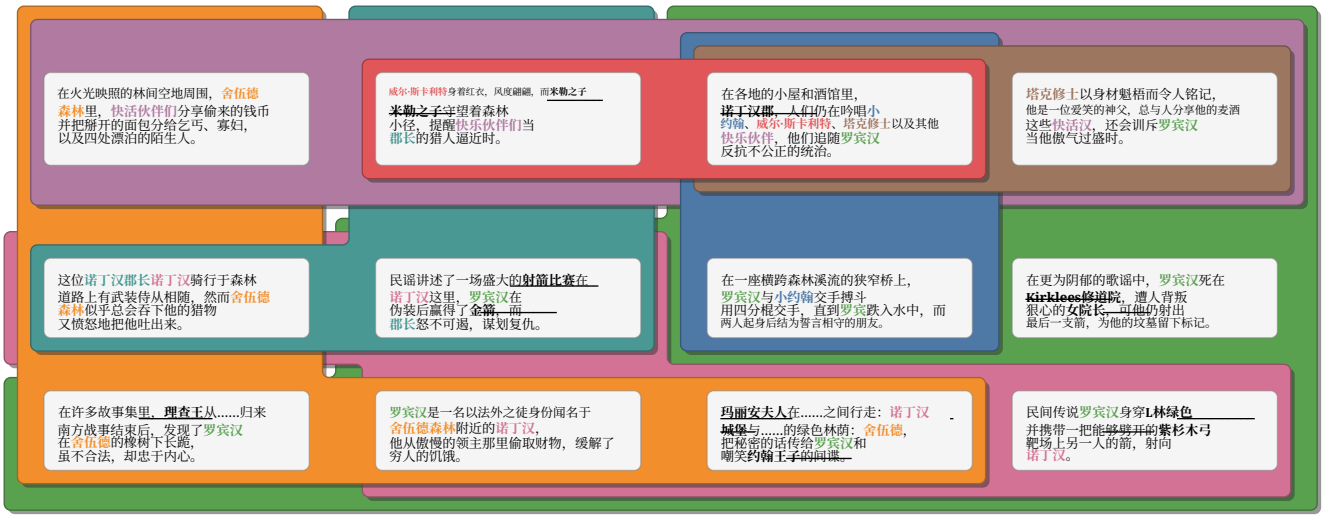


Figure 1: 使用 *BlockSets* 可视化由 *GPT 5.1* 生成的陈述，这些陈述概述了罗宾汉的故事集 .

摘要

集合系统的可视化通常会使用包围几何形状来表示集合，并搭配元素的简化表示，例如简短的文本标签、小型字形或点。因此，这类可视化通常无法充分呈现元素为较长文本片段、图像或图表的集合。本文介绍***BlockSets***，这是一种专为含大型元素的集合设计的新型集合可视化技术。*BlockSets* 将元素放置在网格上，并使用直线形状作为包围几何形状。我们描述了用于在网格上寻找高质量元素布局的整数线性规划。由于并非所有集合系统都能以这种形式实现紧凑且连续的表示，我们还提出了一种算法，在需要时将可视化拆分为多个部分；最终的可视化会通过视觉编码向用户突出显示这些部分。*BlockSets* 采用正交凸形状，在紧凑性与可读性之间实现了良好的平衡。最后，*BlockSets* 将包围几何形状渲染为堆叠的不透明形状。我们描述了一种用于确定堆叠顺序的算法，使所有形状都能被辨认出来。虽然这种堆叠顺序不一定存在，但对于我们测试的所有真实世界数据集，该算法都找到了可行的堆叠顺序。

1. 引言

集合可视化是一个规模庞大且活跃的研究领域。由于可能的集合系统数量巨大，且集合之间的交叠方式复杂，要找到易于理解的可视化表示——

这些交集的可视化是一项具有挑战性的问题。此外，展示元素的集合成员关系往往并非唯一的挑战：元素本身可能带有需要纳入可视化的数据属性。集合可视化需要在集合的视觉需求与元素的视觉需求之间取得平衡。在这方面，大多数集合可视化技术更偏重集合。它们通常

* 这些作者贡献相同。

使用包围几何形状来编码集合关系，而只为元素分配很少的空间；元素通常以简短的文本标签、小型字形或点来表示，有时甚至完全不显示。因此，它们通常无法很好地表示这样的集合系统：其元素是较长的文本片段、图像、图表或上述内容的组合。

例如，大语言模型生成的文本摘要就包含此类数据。虽然文本摘要传统上以陈述序列的形式呈现，但文本中的实体（如人物、物体或地点）会以复杂的方式将各个陈述联系起来。我们可以将陈述和实体视为一个集合系统，其中实体是集合，而陈述则是其所提及实体所属的所有集合中的元素。此类陈述-实体数据是我们开展这项工作的最初动机；我们认为集合可视化是一种很有前景的工具，可用于创建结构化、紧凑的非线性布局，从而帮助理解文本。另一个例子是叠加了集合系统的小多图，例如一份产品库存清单，其中集合代表类别，图表展示随时间变化的销量。还有一种集合系统，其元素是图像。

以上示例表明，针对大型元素的集合可视化需要满足若干具体要求。首先，可视化需要分配一个足够大的区域（通常为矩形）来展示元素数据。其次，应以某种方式对齐元素，以便于在小多图中进行比较，并帮助读者理解文本摘要。第三，由于元素较大，因此应将其相对紧凑地排列，以充分利用（屏幕）空间。当然，除了这些要求之外，集合可视化还需满足标准的可读性要求，即用户能够轻松判断集合成员关系和元素。

尽管可视化集合系统的方法有很多，但只有少数方法能够容纳较大的矩形元素形状。其中包括 Paetzhold 等人提出的 RectEuler 等人 [PKS*23] 以及 Riche 和 Dwyer 提出的“Untangled Euler Diagrams” [RD10]。这两种方法都使用矩形表示集合，因此原则上能够纳入较大的数据块。然而，它们都没有对布局施加对齐要求，而且矩形集合形状不够灵活，无法创建紧凑的可视化效果。相比之下，Mosaic-Sets [RWB*23], 将元素置于网格上，并使用任意复杂的直线多边形作为包围形状。该可视化结构清晰且紧凑，但由于多边形形状过于复杂，难以辨认（见 图3）。

贡献。 本文介绍了**BlockSets**：一种新颖的集合可视化技术，专为含大型元素的集合设计。BlockSets通过将元素放置在网格上，并使用正交凸直线形状作为包围几何形状，创建结构化可视化。因此，我们的方法介于RectEuler的矩形集合表示与MosaicSets的任意直线形状之间。正交凸形状提供了必要的灵活性，既能找到紧凑的布局，又能提高可读性。集合可视化技术通常仅使用轮廓或透明填充来渲染包围几何形状。相比之下，BlockSets使用堆叠的不透明形状。这样可以缓解由集合轮廓交叉和重合以及透明区域颜色混合所导致的可读性问题。具体而言，我们在算法方面的贡献如下：

布局算法。 一种整数线性规划（ILP），用于在网格上为集合元素寻找紧凑布局。该ILP支持一般的正交凸包围形状、限制更多因而视觉上更简单的 ∇-形状和 Γ-形状，以及矩形。

拆分算法。 并非所有集合系统都能在网格上以紧凑、连续的方式表示，并使用正交凸包围形状。因此，我们提出了一种算法，以适当的方式拆分集合系统，独立求解各个部分，然后紧凑地排列这些部分。与 RectEuler 所使用的启发式方法相比，我们的算法生成的重复集合更少 [PKS*23]。我们的视觉编码会突出显示重复集合，供用户识别。

堆叠算法。 最后，我们介绍一种算法，用于为集合计算堆叠顺序，使得即使在不透明渲染下也能推断出所有形状。虽然对于任意集合系统而言，这样的堆叠顺序未必存在，但我们的算法为所有经过测试的真实世界数据集都找到了可行解。

图2展示了我们针对单个连通分量的算法流程。

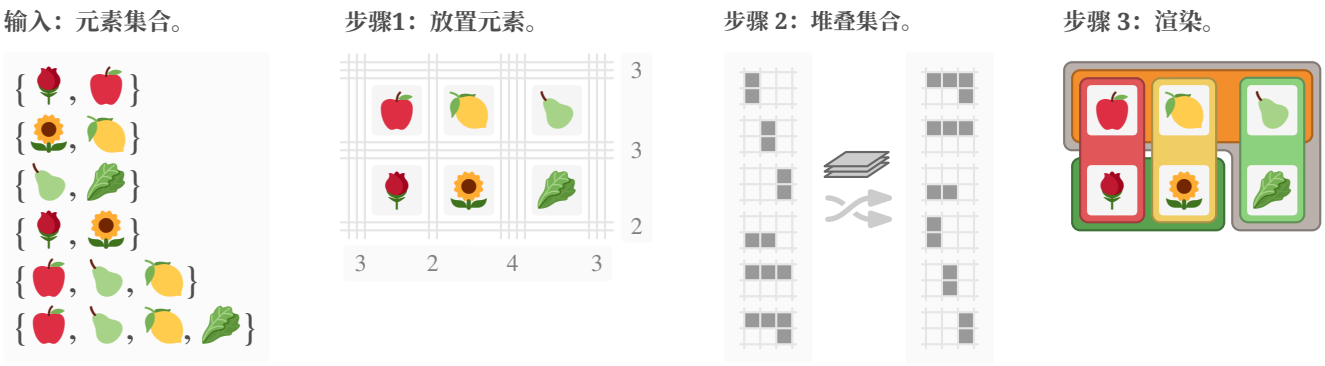
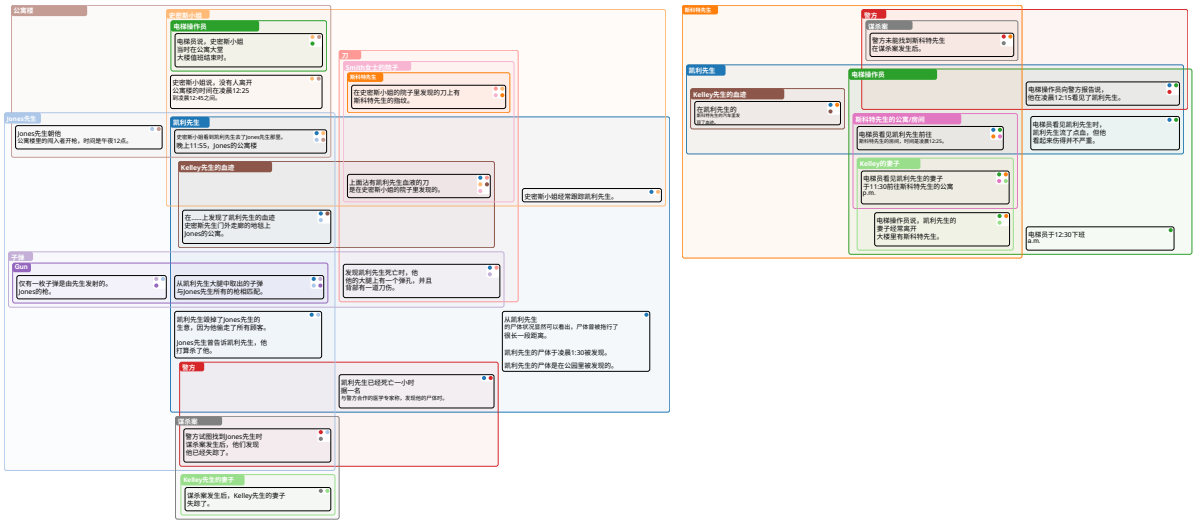


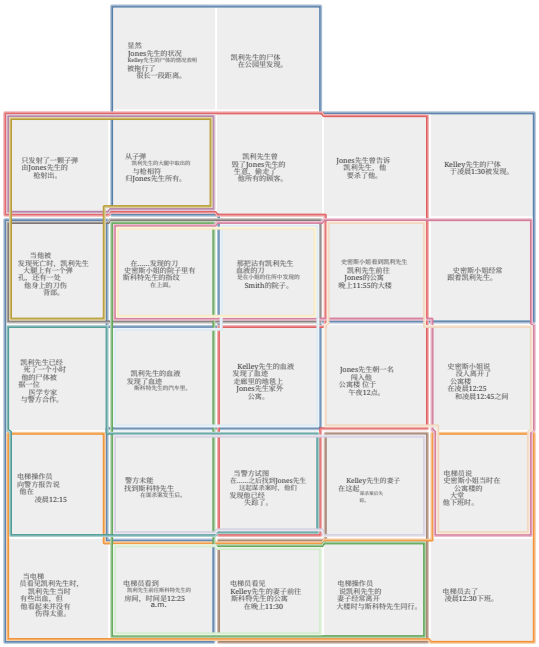
图2： 可视化中单个连通分量的算法流程。*BlockSets* 接收作为输入的元素——每个元素都具有关联的视觉表示——这些元素被分组到各个集合中。首先，*BlockSets* 将元素分配到矩形网格中的单元格内（详见 第*4*节）。元素的位置决定了每个集合所占据的单元格。网格中会预留适当的空间（参见图中的数字），用于放置表示各个集合的包围形状。在下一步中，*BlockSets* 确定各集合的堆叠顺序，使得所有形状都可以被推断出来（第 *6* 节）。最后，创建包围形状、分配颜色，并渲染最终的可视化结果。



(a) *RectEuler* *[PKS*23]*：分为两部分，有7个重复集合，布局不紧凑且未对齐，陈述有时会被分组成条目，透明集合的颜色混合会降低可读性。



(b) 由 ▽ 个形状构成的 *BlockSets*，分为两部分，其中 *4* 个重复集合以虚线轮廓标出。与 *MosaicSets* 相比，它更不紧凑；与 *RectEuler* 相比，它更紧凑；并且比两者都更易读。



(c) *MosaicSets* *[RWB*23]*：布局紧凑，采用任意直角形状，集合边界相交，集合难以追踪。

图3： 使用大语言模型从谋杀悬疑谜题中提取的陈述-实体集合系统 *[SS69]*。*RectEuler* 和 *MosaicSets* 使用作者提供的代码计算。补充材料中的图*21* 展示了同一数据集的连续正交凸*BlockSets*可视化。此外，图*19* 和 图*20* 展示了以 *BlockSets* 风格渲染的 *RectEuler* 和 *MosaicSets* 方案。

组织。 下一节我们将回顾相关工作。然后，在第3节中，我们讨论 BlockSets 背后的具体设计决策和目标。接下来的章节将详细介绍我们的算法贡献：第第4节中的布局算法、第第5节中的拆分算法及各部分的排列，以及第第 6 节中的堆叠算法。最后，我们通过 BlockSet 可视化展示和讨论来总结本文。

2. 相关工作

我们简要概述与 BlockSets 相关的技术，这些技术要么用于可视化集合，要么使用网格。关于集合可视化的全面综述，请参阅 Alsallakh 等人撰写的综述 等 [AMA*16]。他们将集合可视化分为六个类别。BlockSets 并不完全属于其中任何一个类别，但与欧拉图和维恩图最为相关，也与基于矩阵的技术有些相似；因此，本节仅讨论这些类别中的集合可视化方法。

欧拉图和维恩图。 欧拉图和维恩图使用相互重叠的形状，展示一组集合之间的各种交集方式。维恩图展示所有可能的集合交集，而欧拉图则旨在仅展示输入数据中存在的集合交集。如果欧拉图能够如实反映输入数据，那么它就是匹配良好的。若欧拉图中的形状彼此横截相交、每个集合恰好由一个形状表示，且每个集合交集在图中恰好构成一个连通区域，则该欧拉图是形式良好的。尽管 BlockSets 与欧拉图有相似之处，但我们无意将其设计成欧拉图。具体而言，BlockSets 并非形式良好，因为我们不要求集合交集构成单一的连通区域，而且集合可以由多个形状表示。此外，BlockSets 也并非匹配良好；例如，在图 3b 中存在一个红色与蓝色的区域（其中子弹实体与凯利先生实体相互重叠），但没有任何陈述恰好属于这两个集合。由于 BlockSets 关注的是集合系统中的元素，我们认为这不会造成混淆，因此选择利用非匹配良好所带来的灵活性。

欧拉图通常使用弯曲的形状 [CR05；KGWD22；WRD21；PAQ18；MR14；RZF08；SRHZ10], 这妨碍了大型元素视觉表示的整合。正如引言中所述，RectEuler [PKS*23], MosaicSets [RWB*23]以及“Untangled Euler Diagrams” [RD10] 是最适合可视化大型元素的欧拉图式图示（图3）。此外，还有其他类似欧拉图的可视化方法，例如 MetroSets [JWKN21]。尽管 MetroSets 更规整的布局有助于整合元素表示，但其可视化并未高效利用空间，这尤其限制了它在大型元素场景中的应用。

关于哪些形状类别可以表示哪些类型的集合系统，已有一些数学结果。对于矩形，已知在平面上识别轴对齐矩形相交图的问题是 NP 完全的 [Kra94], , 这实际上意味着，除非 P == NP，否则不存在能够高效判定某个集合系统是否可以用良配的欧拉图表示、且其中每个集合都用一个矩形表示的算法。对于凸多边形，已知可以用每个集合对应一个凸多边形、但允许出现不连通区域的良配欧拉图表示任意大且各集合两两相交的集合系统 [Grü75]。这一结果也适用于 BlockSets 所使用的正交凸多边形。

超图绘制。集合可视化可以看作是超图的绘制，其中超边由各个集合构成。关于细分绘图 [KvKS08；vBKK*16]以及超图的平面支撑 [BvKM*11], 的研究均与欧拉图有关。也有研究探讨使用互不相交的多边形表示集合的绘图 [DDKW24；vGKvK*17]。尽管这类研究与集合可视化相关且在理论上颇有意义，但并未提供创建功能完备的集合可视化的算法。

矩阵集合可视化。矩阵集合可视化非常紧凑，能够可视化大规模数据。与 BlockSets 一样，它们使用方形网格来可视化集合。不过，它们的工作方式截然不同，因为矩阵中的单元格编码的是二进制信息，例如元素隶属关系 [KLS07；SMDS14；WDN23]或集合间的交集 [LGS*14], 而不是包含元素的视觉表示。

其他网格可视化。除集合可视化外，还有几种与 BlockSets 相似的网格可视化。Yoghourdjian等人在网格上可视化网络，并绘制包围矩形以形成节点组 [YDG*15]。这些组使绘图更简洁：在他们的可视化中，两个组之间的一条边表示第一组中的所有节点都与第二组中的所有节点相连。网格也广泛用于分析机器学习模型输出 [ZYC*24], 以及小倍图的可视化 [MSS21]。

3. BlockSets

本节将讨论 BlockSets 的目标与设计。BlockSets 可视化的输入由元素以及这些元素的若干子集组成。每个元素都有相应的视觉表示，例如一段文本、一幅图像或一张图表。我们假设每个元素的视觉表示都能放入尺寸相近的矩形中。

设计目标。 遵循欧拉图以及格式塔的共同区域原则 [WEK*12], 我们使用包围几何形状来表示集合。主要有以下两个设计目标。

紧凑性。 可视化（*i*）占用数据元素之外的空间较少；且（*ii*）纵横比较小。**可读性**。 可视化具有清晰的结构，用户能够高效地确定（*i*）某个元素所属的所有集合；以及（*ii*）某个集合包含的所有元素。

需要注意的是，这些目标彼此冲突：减少集合所占用的空间通常会降低这些集合的可读性。

网格布局。 我们选择将元素排列在矩形网格上。网格布局长期以来一直被排版和页面布局领域视为一项基本设计原则，其在图与集合数据排列中的适用性和重要性也得到了可视化领域的探索 [YDG*15；RWB*23]。网格布局使元素的矩形表示彼此对齐，从而形成结构化可视化；同时也相对紧凑。

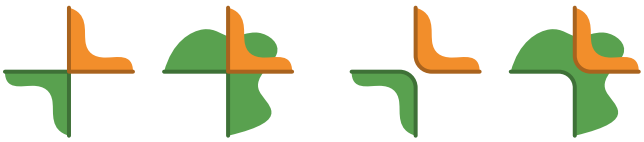


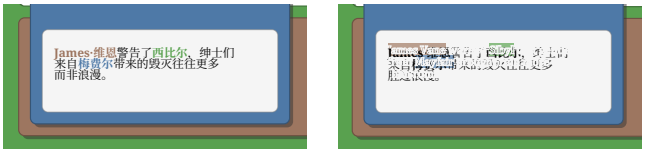
图4： 左：我们允许的点重叠类型；彩色区域表示内部区域。右：我们采用圆角处理以提高可读性。

自然地，BlockSets 也会将包围形状放置在矩形网格上。为保证可读性，我们要求这些包围形状的任意两条边最多只能在一个点上重叠。具体而言，仅在两种情况下允许单点重叠（图4）。为确保形状彼此不重叠，我们在网格的连续行和列之间留出间隙（图2）。我们将包围形状的角稍作圆化，以提升可读性并避免尖角过于醒目。包围形状的边以均匀间距排列。

集合表示。 集合数据不仅可以对元素进行视觉表示，也可以对集合进行视觉表示。在陈述-实体数据中，这指的是实体名称。为了保持紧凑性，我们选择不将这些名称绘制为标签，而是在陈述文本中用单一且独特的颜色突出显示实体名称，并为该实体的包围形状使用相同的颜色。我们考虑了两种设计：彩色文本，以及作为高亮显示的彩色背景（见 图5）。彩色文本需要使用深色，因此仅当非单元素集合的数量相对较少时才适用。不过，它通常显得不那么杂乱；本文中的所有示例均采用这种方式绘制。请注意，单元素实体既不需要颜色，也不需要包围形状；我们只需在其陈述中使用粗体和下划线将其突出显示即可。陈述-实体关系之外的其他集合系统可能适合使用标签；在 图14a 中，我们展示了一个带标签的 BlockSets 模拟图。本文不再进一步讨论标签布局或其算法方面的挑战。

拆分。 如果将每个集合表示为单一形状，某些集合系统就无法实现紧凑且易读的可视化。因此，在这类情况下，我们允许对选定的集合使用多个形状。为确保问题在计算上可行，我们遵循 RectEuler [PKS*23]的做法，将可视化拆分为多个独立部分分别求解（见图 3b 中的示例）。每个部分都有自己的包围形状，但有些集合会在不同部分之间共享。这些集合可以通过颜色相互对应。在 BlockSets 中，表示共享集合的形状

接收一条虚线边界，让用户清楚地看到哪些集合是由多个形状表示的。请注意，拆分输入可以可视化任何集合系统，因为在极限情况下，每个元素

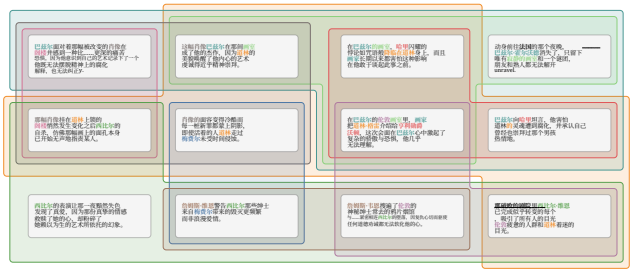


(a) 彩色文本。 **(b)**彩色背景。

图 5： 突出显示实体名称的两种设计。



(a) 不透明填充色。



(b) 透明填充色。

图 6： *BlockSets* 对受《道林·格雷》启发、由 *GPT5.1* 生成的“王尔德风格”叙事语句的两种渲染。

都有自己的连通分量。另一种替代多个包围形状的方法是像 DupEd [RD10]那样复制元素。对于大型元素，复制其表示会占用大量空间；因此，我们决定不采用这种方法。

堆叠。 大多数集合可视化仅使用轮廓或透明填充来渲染包围几何形状 [CPC09； PKS*23； KGWD22； RWB*23]。仅使用轮廓可能会使人难以确定某个元素所属的所有集合（另见 图 3c）。透明度在一定程度上缓解了这些问题，但会导致多个重叠集合的区域发生颜色混合，从而使可视化在许多地方呈现出不具吸引力且效果不佳的浑浊棕色（见 图 6 下方）。因此，我们选择使用不透明填充色（见 图 6 上方）。

虽然不透明性更便于进行颜色匹配，但表示集合的形状现在可能会相互遮挡，因此形状的堆叠方式变得十分重要。这引出了一个问题：我们能否找到一种堆叠方式，确保所有形状都可以被推断出来？在 第 6 节中，我们描述了一种算法；如果存在这样的堆叠方式，该算法便能找到它。尽管可以构造出不存在堆叠方式的示例，但在我们测试过的真实世界数据中，尚未遇到任何此类情况。如果确实出现这种情况，则可以添加视觉辅助元素，以确保能够推断出这些形状。

值得注意的是，Riche 和 Dwyer 的 Untangled Euler Diagrams [RD10] 也使用不透明颜色来渲染集合。不过，他们的设计本质上只堆叠彼此嵌套的形状，因此无需解决集合边界被遮挡的问题。据我们所知，此前并没有集合可视化方面的研究

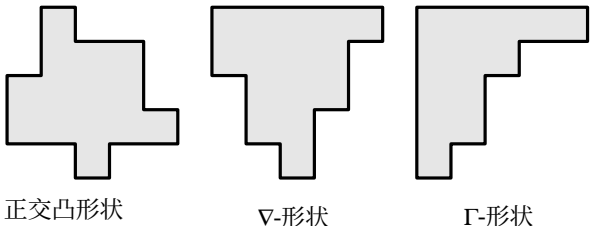


图 7： *BlockSets* 使用的包围形状。

以我们为 BlockSets 提出的方式堆叠形状。Ca- bello等人 [CHvKS10] 介绍了用于生成清晰的比例符号地图的算法，这些地图使用圆盘或正方形等不透明符号。他们旨在最大限度地延长每个符号可见边界的长度，并考虑两类绘图：堆叠式绘图，以及允许形状在第三维中“弯曲”的物理可实现绘图。我们只考虑堆叠式绘图，因为物理可实现绘图可能会让用户感到困惑，而堆叠式绘图已经足以推断出所有形状。

集合颜色。BlockSets 使用颜色来：（1）区分不同的集合；（2）当输入被拆分时，在视觉上将重复的集合归为一组；（3）在陈述-实体数据中，将实体名称与相应的包围形状匹配。我们使用 Tableau20 作为调色板，并采用模拟退火作为启发式方法，为集合分配颜色，使空间位置接近的集合所获得的颜色尽可能不同，具体依据是它们的 CIEDE2000色差。这里，我们认为相互重叠的集合比不重叠的形状接近得多。此外，在堆叠顺序中彼此相邻且相互重叠的集合，比堆叠顺序中相距较远的集合更接近。

包围形状的类型。包围形状的选择会对可视化的紧凑性和可读性产生很大影响。由 [RD10；PKS*23] 所使用的矩形是最容易阅读的形状，但缺乏灵活性。相比之下，MosaicSets 所使用的任意直线多边形 [RWB*23] 具有灵活性，但难以阅读。作为示例，请考虑 图 3c中凯利先生的包围形状：第二列在包含于集合和不包含于集合的陈述之间来回切换了四次。这类“间隙”使人难以一眼看清整个集合。能够避免此类切换的形状是正交凸 形状（orthogonally-convex shapes 的简称），即任意垂直线或水平线与该形状的交集要么为空，要么由单个连通分量组成。

矩形通常无法实现紧凑的集合可视化，但其直边使读者能够根据局部特征轻松推断出该形状的整体结构。对于 BlockSets，我们希望
利用正交凸形状增加的灵活性，同时保留矩形的一些有用结构。我们将形状称为 顶部对齐，如果对于形状中的任意点 (x, y)，从该点(x, y)到 (x, y_{max})的线段完全包含在该形状内；此处 y_{max}表示该形状的最大 y坐标。类似地，我们定义左对齐。我们考虑以下类型的形状，限制程度从低到高依次排列（图 7）：

- 任意直线形状；
- 正交凸形状；
- ∇-形状：顶部对齐的正交凸形状；
- Γ形状：正交凸形状，顶部和左侧对齐；● 矩形。

选择顶部和左侧对齐，是为了符合英语从左到右、从上到下的阅读顺序。我们发现，正交凸形状和 ∇形状在紧凑性与可读性之间取得了良好的平衡。

计算流程。 图2 展示了我们计算 BlockSets 单个连通分量的流程。对于复杂的集合系统或限制严格的包围形状，可能无法将元素紧凑地分配到网格单元中。在这种情况下，BlockSets 会将输入拆分成多个部分，分别定位，并将其结果组合起来（图8）。以下各节将介绍该流程中各个步骤的算法细节。

4. 布局算法

本节介绍计算量最大的任务：寻找将元素紧凑地分配到网格单元中的方案。我们使用整数线性规划（ILP），在一组约束和特定的目标函数下求解该问题的最优解。为简洁起见，我们仅概述这一方法，不列出约束和目标函数的公式；详情请参阅实现*。我们仅介绍针对正交凸形状、 ∇形状和 Γ形状的整数线性规划。关于矩形和任意多边形的建模，请分别参阅 RectEuler [PKS*23]和 MosaicSets [RWB*23]。

元素和集合表示。整数线性规划在固定大小的方形网格上运行，所有元素都位于该网格内。网格的大小

* github.com/tue-alga/BlockSets

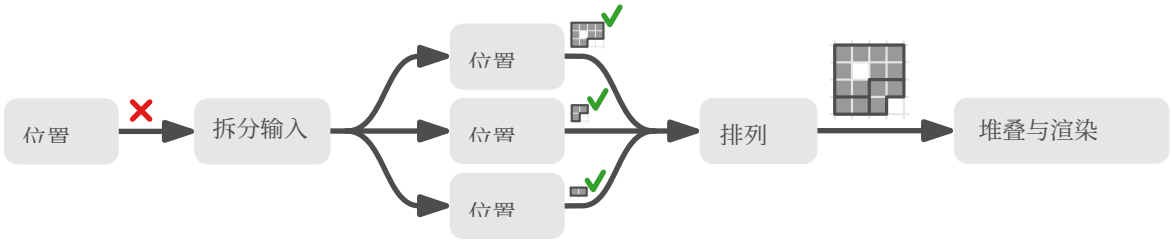


图 8： 如果无法将元素紧凑地分配到网格单元中，*BlockSets* 会将输入拆分为若干部分（第*5*节）。首先分别放置每个部分中的元素（第*4*节），然后将它们紧凑地排列在单个网格中（第*5.1*节），最后将各个集合堆叠并渲染。

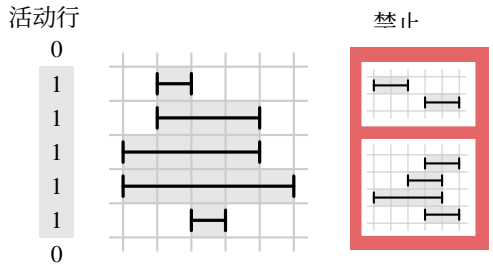


图 9：我们用一组连续活动行上的区间来表示一个正交凸多边形。

可以设为例如$\lceil\sqrt{n}\rceil+1\times\lceil\sqrt{n}\rceil+1$，其中$n$是元素总数。布尔变量用于确定每个元素与网格单元的组合中，该元素是否放置在相应的网格单元内。适当的约束可确保每个元素恰好分配到一个网格单元。每个集合用一组区间表示（图 9），且每行至多使用一个区间。包含区间的所有行称为活动行。要将集合表示为正交凸多边形：

1. 其活动行应当连续；2. 其连续行区间应当彼此重叠；3. 这些区间应构成一个垂直凸多边形。

最后一个约束意味着，对于给定区间，不允许其上方的区间和下方的区间同时结束得更晚，或同时开始得更早。示意见图 9。

∇- 和 Γ-形状。为了对 ∇-形状建模，我们添加一个约束，要求每个区间都是其上方区间的子集。对于 Γ-形状，一个变量决定给定集合中所有区间的起点。

元素隶属关系。整数线性规划约束确保，当且仅当某个元素属于某个集合时，该元素才会被放置在表示该集合的某个区间内。最后，如果两个形状所属的集合没有共同元素，我们不允许它们重叠；因此，我们要求这类集合中位于同一行的区间彼此不相交。

目标函数。我们最小化以下三个项的线性组合（权重相同）：

多边形面积。为提高可读性，我们避免集合内出现空网格单元。对此，我们通过最小化多边形面积之和来建模。**包围盒总尺寸。**为保证紧凑性，我们希望可视化占用空间小且纵横比较低。因此，我们最小化已填充网格的尺寸之和。**多边形复杂度。**最后，为提高可读性，我们最小化每个多边形的顶点数。

5. 拆分算法

如果整数线性规划（第4节）无法找到紧凑布局，BlockSets 就会将输入拆分成多个部分。RectEuler 也会根据需要拆分输入；下面我们将说明我们采用的拆分方法与其有何不同。理想情况下，我们会将输入拆分，使每个部分都能以最紧凑且最具可读性的方式进行可视化。然而，可能的拆分方式有很多，对所有这些拆分都运行布局算法在计算上不可行。因此，我们需要采用启发式方法。

RectEuler [PKS*23] 会根据元素所属的集合（通过 Jaccard度量），递归地将元素聚类为两组。我们发现，这种方法并不总能得到良好的拆分，参见补充材料中的图22和图23。根据集合的时间限制，RectEuler 要么不进行拆分，导致可视化结果非常稀疏（图23）；要么进行拆分，并复制许多集合。这是因为 RectEuler 在计算拆分时关注的是元素，而非集合。然而，重复的集合会降低可读性。我们采用以集合为中心的拆分方法，旨在尽量减少重复集合的数量。

决定何时拆分。BlockSets 对整数线性规划（ILP）设置了用户指定的时间限制。尤其是，布局算法中的整数线性规划可能需要很长时间才能完成，但也可能很快就找到最优解或接近最优的解。如果布局算法未能在此限制时间内找到布局，我们就会拆分输入。如果算法返回的布局不够紧凑，我们也会拆分输入。我们通过统计集合的已填充网格单元的正交凸包内的空网格单元数量来衡量紧凑性；该正交凸包是包含这些已填充网格单元的最小正交凸形状。如果这一数量超过$15n$，我们就进行拆分，其中n是布局中的元素数量。使用矩形作为包围形状时，通常会出现许多空网格单元；为避免拆分过多，可以采用更大的阈值，或完全移除这一拆分条件。

拆分的质量。除了尽量减少重复集合外，我们还希望拆分后的每个部分尽可能简单。为此，我们寻找能将集合大致均匀分配到各个部分的拆分。因此，拆分后应得到少量连通分量，并且这些连通分量：

- 重复集合尽可能少；
- 大小均衡；并且
- 无需进一步拆分。

对偶超图。我们通过集合系统的对偶超图G来寻找划分。该超图为每个集合设置一个顶点，并为每个元素设置一条超边，连接所有包含该元素的集合（图10）。如果拆分后每个部分至少包含一个未重复的集合，我们就称这种拆分是适当的。适当拆分通过移除一组顶点来断开G。也就是说，适当拆分基于G中的顶点分隔集。给定一个顶点分隔集S，我们按如下方式进行拆分。顶点

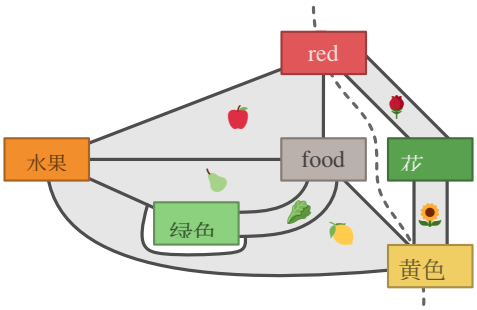


图10：输入内容对应的对偶超图的齐科夫绘图图2（每条超边都是绘图中的一个面；与该面关联的所有顶 {红色、黄色} 点构成该超边）。顶点集形成一个顶点分隔集。

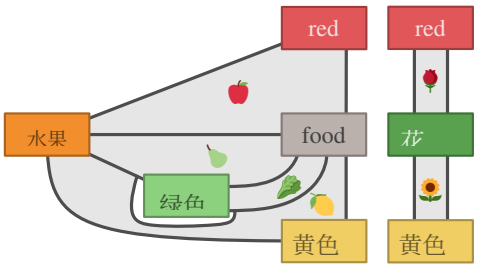


图11： 分割后的对偶超图。

集S将G分隔为若干连通分量C_1、……、C_k。为了构造拆分后作为 BlockSets 输入部分的G_1、……、G_k，我们将每个顶点$v \in S$“切开”：在它所连接的每个连通分量中放置该顶点的一个副本，并将与其关联的超边相应地分配到各个连通分量中（图11）。唯一不进行分配的超边，是那些仅连接S中顶点的超边。与这些超边对应的元素可以自由放入任意连通分量中。我们按超边大小递减的顺序逐个放置这些元素：将其放入包含该超边中顶点数量最多的最小连通分量中，并在需要时将缺失的顶点添加到该连通分量中。

分隔集应移除尽可能少的顶点，使得剩余图中的连通分量大小相近：连通分量的最大大小为 αn，其中n是超图中的顶点数。这类分隔集称为最小α-顶点分隔集，而在正则图中寻找或近似求得这样的分隔集是NP困难的 [BJ92]。此外，最小顶点分隔集可能无法形成良好的分割。考虑一个星图。中心顶点构成一个大小为一的顶点分隔集。移除中心顶点后，会产生许多连通分量。更好的分割方式是将其中一些连通分量合并成更大的部分，使得它们共享的节点只需表示一次。因此，为了构造分隔集，我们使用蛮力算法，反复合并最小的两个连通分量，直到所有连通分量都处于我们的大小限制之内。

蛮力算法。 我们遍历所有大小不超过 5 的顶点分隔集，使得合并后每个连通分量的大小均处于 $[\frac{1}{3}n, 23\ n]$, 之间，其中 n 表示超图中的顶点总数。对分隔集大小设置上限，是为了将枚举范围限制在较小的分隔集内。我们选择使以下各项加权和最小的顶点分隔集：

- 连通分量的数量；
- 已删除顶点的数量；
- 所有连通分量中已删除顶点副本的数量；
- 10 乘以大小超过最大阈值的连通分量数量（以尽可能避免某个连通分量还需再次拆分才能可视化）；以及
- 最大连通分量与最小连通分量之间的大小差（以确保各连通分量大小均衡）。

如果蛮力算法未能找到顶点分隔集，我们就复制所有集合。在这种情况下，我们创建两个空的连通分量；所有超边的顶点都只位于分隔集 S 中，并按大小降序分配到各个连通分量中。

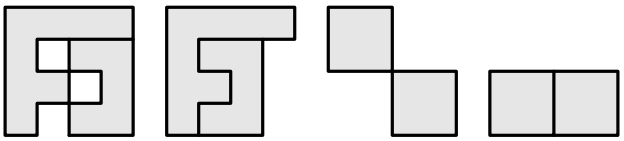


图 12： 排列 *(a)* 和 *(b)* 的包围盒尺寸相同，但 *(b)* 的正交凸包面积更小。排列 *(c)* 和 *(d)* 的正交凸包面积相同，但 *(d)* 的包围盒尺寸更小。

5.1. 排列组件

如果 BlockSets 可视化被拆分，我们就为每个连通分量计算布局。随后需要以紧凑的方式排列这些布局。与布局问题一样，这个优化问题可以使用整数线性规划求解。有效解的两个主要约束是：

1. 每个组件恰好放置一次；以及 2. 组件之间不存在重叠。

如果所使用的形状类别在镜像或旋转操作下是封闭的，我们允许对组件进行镜像和旋转。也就是说，使用 ∇-形状和 Γ-形状时，我们不会进行垂直旋转或镜像；使用 Γ-形状时，我们不会进行水平镜像。我们也不会镜像矩形，也不会将其旋转 180 度。只有当变换后组件的形状与该组件的其他版本不同时，我们才会使用变换后的组件。对于每一对组件和可放置该组件的位置，我们定义一个布尔变量。对于每个组件，这些变量中必须恰好有一个被设为 true，从而确保满足第一个约束。对于第二个约束，我们统计网格中的每个单元格在该位置放置了多少个占用单元格，并将其约束为至多一个。

目标函数。 我们将以下各项的线性组合最小化。

包围盒总尺寸。 我们将包围盒尺寸中宽度与高度之和最小化。**尺寸差。** 为避免布局过度拉伸，我们将宽度与高度之差的绝对值最小化。**正交凸包面积。** 对于包围盒大致相同的解，我们倾向于选择组件排列得更紧凑的解（图 12）。为体现这一偏好，我们加入正交凸包面积这一项。

6. 堆叠算法

BlockSets 会以不透明方式渲染包围几何形状，因此它们可能会相互遮挡。我们会计算一个堆叠顺序，使每个形状都尽可能易于辨认。通常无需让形状的全部区域都可见，只要能根据可见部分推断出其整体形状即可。对于直线多边形，我们将这种可见性定义如下：当且仅当多边形（集合）的所有边都可见时，该多边形才可见。只要一条边有部分可见，该边就视为可见。如果多边形或边不可见，则视为被遮挡。可能无法通过堆叠多边形使它们全部可见（参见合成示例 图 13），但我们测试的所有真实示例都能通过我们的算法找到合适的堆叠方式。

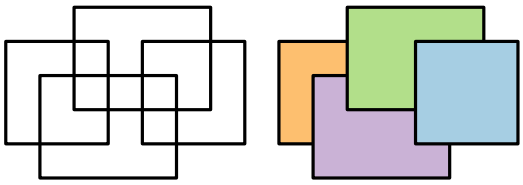


图 13： 在任何堆叠方式中，至少有一个矩形会被遮挡。

6.1. 算法

我们的堆叠算法以布局算法构建的形状（占用单元格的多边形边界）为输入，然后从下到上迭代堆叠这些多边形。直观而言，我们总是优先放置被所有尚未放置的多边形遮挡最少的多边形。为此，算法会维护一个尚未放置的多边形集合 S，该集合最初包含所有多边形。随后，算法会反复执行以下步骤。

1. 对于每个多边形 $P \in S$，检查其所有边是否被多边形集合 $S \setminus \{P\}$ 遮挡。2. 放置满足以下条件的多边形：

a. 可见边最多；或在平局时：b. 边最多；或在平局时：c. 面积最大。

边可见性。 为了执行可见性检查，我们首先计算所有多边形的排列（交集）。在最后的渲染步骤中，我们会将重合的边分开渲染，以便让各集合正确地层叠在一起。为此，我们会在处理流程的早期阶段预留网格空间（回想一下图2中的步骤 1：数字表示在该位置重合的形状多边形的边）。因此，就我们的目的而言，两条完全重叠的边都被视为可见；只有当一条边被另一个多边形的内部区域覆盖时，它才算被遮挡。对于排列的顶点，也适用类似的推理：只有当顶点位于另一个多边形的内部区域，或落在另一条边的边界上且不属于图4所示的两种情形时，该顶点才算被遮挡。

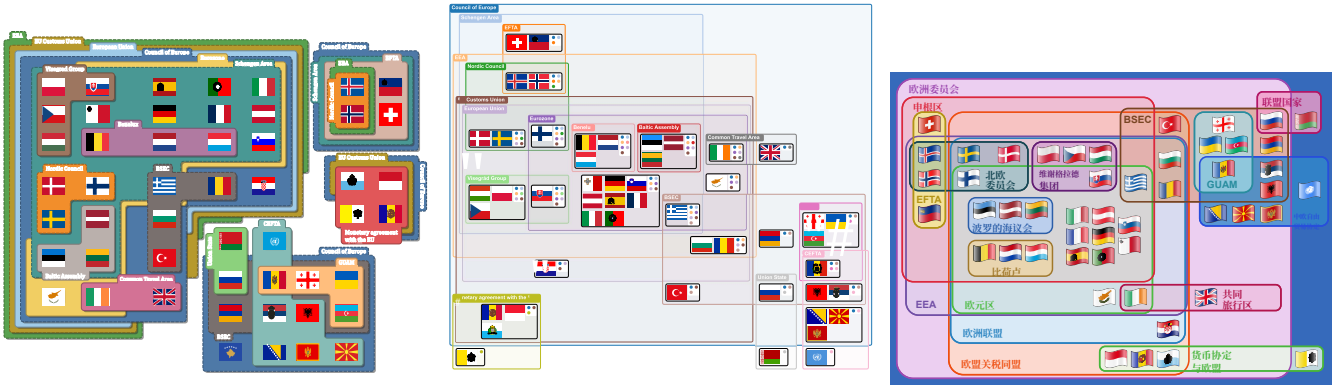
7. 展示

本节将展示 BlockSets 在各种数据集上的应用，并使用多种不同的包围形状。所有图均由 BlockSets 的原型实现生成，该实现已公开发布于github.com/tue-alga/BlockSets。

陈述-实体数据。 图1、图3、图 6 分别展示了 BlockSets 在不同类型的陈述-实体数据上的应用：一组故事集、一则谋杀谜案和一个单独的故事。罗宾汉故事集和谋杀谜案没有既定的线性顺序，而道林·格雷的故事则有其内在的叙事顺序。因此，在这些可视化中使用 BlockSets 的方式会有所不同。尽管故事集和谋杀谜案中的陈述没有既定顺序，但它们会
传统上仍以线性方式呈现。BlockSets 则提供了二维阅读体验。读者可以从网格中的第一条陈述开始，按顺序阅读；BlockSets 的布局算法确保网格中彼此靠近的陈述，在其提及的实体方面也彼此相似。如果读者对某个实体产生疑问，可以根据该实体的颜色，在可视化中找到对应的包围形状，从而有效地找到其他提及该实体的陈述。

此外，包围形状使读者能够有效判断陈述和实体的出现频率。例如，图 1 中代表罗宾汉的形状很大，覆盖了除三条以外的所有陈述，显示了他在这些故事集中的重要地位。再看道林·格雷的可视化，规则的网格布局与包围形状之间的间距相结合，使一条重要的核心陈述格外醒目：巴兹尔、道林·格雷和亨利·沃顿勋爵在巴兹尔的伦敦画室相遇。这也展现了 BlockSets 在故事分析方面的潜力：它提供了一种解构式视图，揭示实体的出现频率及其相互作用。

小多图和图像。 图14 和 图15 使用 BlockSets 来可视化具有底层集合系统的图像和图表。这里，我们手动将标签附加到包围形状上，以显示集合名称。 图14 展示了欧洲超国家机构的 BlockSets 可视化，并将其与 RectEuler 和手工绘制的欧拉图进行比较。由于这里的元素较小，我们不



(a) 带有 ∇-形状的 *BlockSets*　　**(b)** *RectEuler [PKS*23]*　　**(c)**手工制作 *[WEN21]*

图14： 一项比较 超国家欧洲机构的集合可视化　　体。

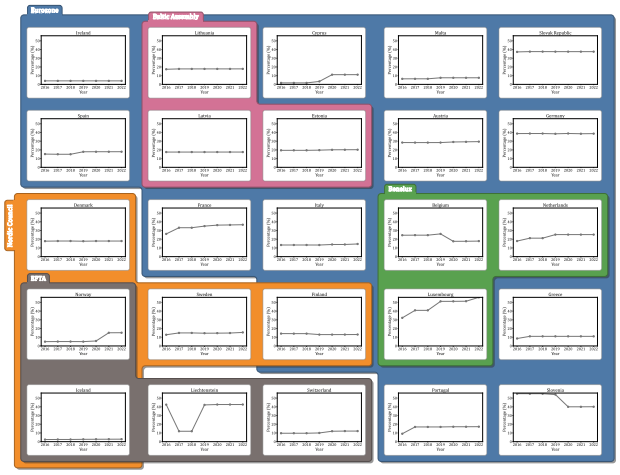


图15： 使用 *BlockSets* 将陆地和海洋保护区可视化为总面积的百分比（*2016–2022* 年）。来源：世界银行，环境、社会和治理（*ESG*）数据，指标 *WB ESG ER PTD TOTL ZS*。_ _ _ _ _



图16： 用于罗宾汉故事集的 ∇-形状 *BlockSets*。

需要对齐不同连通分量中的元素，因此我们采用手动排列，而非使用第5.1节中的排列算法。第5.1节。利用 ∇形状的灵活性，并将输入拆分为多个部分，BlockSets 得以为元素找到比 RectEuler 紧凑得多的布局。

不同的形状类型。 最后，我们比较了使用不同形状类型的 BlockSets。我们不将任意多边形作为正交凸多边形进行详细讨论，因为后者通常已经达到了最大紧凑程度（图1），而在其他情况下，任意多边形会导致难以阅读的可视化结果（见图 3c 和补充材料中的 图 20 ）。在图 16–18 中，我们可以看到，随着形状类型变得不那么灵活，可视化结果的紧凑性降低，但形状变得更加结构化。使用矩形时，需要将输入拆分为多个连通分量，才能创建紧凑的可视化结果。哪种形状类型效果最佳取决于具体数据集。对于罗宾汉故事集，图1 中的正交凸形状能够创建紧凑且易于阅读的可视化结果；使用结构化程度更高的形状会使可视化结果明显不那么紧凑。然而，对于图14a 和 图 3b ∇形状能够提供紧凑的可视化结果，因此是首选方案。



图17： 用于罗宾汉故事集的 Γ-形状 *BlockSets*。



图18： 用于罗宾汉故事集的矩形 *BlockSets*。

8. 未来工作

BlockSets 是一种用于可视化大型数据集合系统的新型结构化方法。我们的设计决策和算法解决方案受到具体质量标准和要求的指导；这些标准和要求源于相关文献，并由我们旨在可视化的数据类型所决定。我们认为，所得的可视化结果确实能够很好地支持用户处理大型集合数据，尤其是陈述-实体数据。我们发现，当以 BlockSets 形式呈现谋杀谜案及其他相关谜题时，同事们能够轻松解决这些问题。然而，未来若能通过适当的用户研究，在更大的群体中探索 BlockSets 对其他数据类型的可用性，将会有所裨益。

在算法方面，我们希望进一步探索划分算法的不同选项。如前所述，RectEuler采用以元素为中心的方法，而我们更倾向于以集合为中心的方法。不过，通常而言，哪种方法最佳可能取决于要可视化的集合数据类型。因此，我们希望进一步探索数据的哪些特征最适合采用哪种拆分方式，并开发能够针对当前数据自动选择最佳拆分方式的（混合）方法。

最后，我们在图 14 和15 中简要展示了如何为 BlockSets 添加标签，以应对元素数据未直接包含集合信息的情况。开发相应的自动化标注方法是一项有趣的算法挑战。

致谢。感谢 Tomasz Soróbka 对初步设计提出的宝贵意见。

参考文献

[AMA*16] ALSALLAKH, BILAL, MICALLEF, LUANA, AIGNER, WOLF- GANG, HAUSER, HELWIG, MIKSCH, SILVIA,and RODGERS, PETER J.《集合可视化的最新进展》。 *Computer Graphics Forum* 35.1(2016),234–260. DOI:10.1111/CGF.127224.

[BJ92] BUI, THANG NGUYEN 和 JONES, CURT。“寻找优质近似顶点和边划分是 NP困难的。”*Information Processing Letters* 42.3 (1992), 153–159. DOI: 10.1016/0020-0190(92)90140-Q 8。

[BvKM*11] MEULEMANS, WOUTER, SONDAG, MAX, 和 SPECKMANN, BETTINA。《用于相干网格地图的简单流水线》。IEEE 可视化与计算机图形学汇刊 27.2（2021年），1236–1246。DOI：10.1109/TVCG.2020.3028953 4.[PAQ18] PÉREZ-SILVA, JOSÉ G、ARAUJO-VOCES, MIGUEL 和 QUESADA, VÍCTOR。《nVenn：广义准比例维恩图和欧拉图》。Bioinformatics 34.13（2018年），2322–2324。DOI：10.1093/bioinformatics/bty109 4.[PKS*23] PAETZOLD, PATRICK、KEHLBECK, REBECCA、STROBELT, HENDRIK、XUE, YUMENG、STORANDT, SABINE 和 DEUSSEN, OLIVER。《RectEuler：使用矩形可视化相交集合》。Computer Graphics Forum，第42卷，第3期。2023年，87–98。DOI：10.1111/CGF.14814 2–7、9、13、15、16.[RD10] RICHE, NATHALIE HENRY 和 DWYER, TIM。《解开欧拉图》。IEEE 可视化与计算机图形学汇刊 16.6（2010年），1090–1099。DOI：10.1109/TVCG.2010.210 2、4–6.[RWB*23] ROTTMANN, PETER、WALLINGER, MARKUS、BONERATH, ANNIKA、GEDICKE, SVEN、NÖLLENBURG, MARTIN 和 HAUNERT, JAN-HENRIK。《MosaicSets：将集合系统嵌入网格图》。IEEE 可视化与计算机图形学汇刊 29.1（2023年），875–885。DOI：10.1109/TVCG.2022.3209485 2–6、14.[RZF08] RODGERS, PETER、ZHANG, LEISHI 和 FISH, ANDREW。《通用欧拉图生成》。国际图示理论与应用会议。施普林格。2008年，13–27。DOI：10.1007/978-3-540-87730-1 6 4.[SMDS14] SADANA, RAMIK、MAJOR, TIMOTHY、DOVE, ALISTAIR D. M. 和 STASKO, JOHN T。《OnSet：大规模二进制集合数据的可视化技术》。IEEE 可视化与计算机图形学汇刊 20.12（2014年），1993–2002。DOI：10.1109/TVCG.2014.2346249 4.[SRHZ10] STAPLETON, GEM、RODGERS, PETER、HOWSE, JOHN 和 ZHANG, LEISHI。《归纳生成欧拉图》。IEEE 可视化与计算机图形学汇刊 17.1（2010年），88–100。DOI：10.1109/TVCG.2010.28 4.[SS69] STANFORD, GENE 和 STANFORD, BARBARA DODDS。《通过游戏学习讨论技巧》。（1969年）3、15.[vBKK*16] Van BEVERN, RENÉ、KANJ, IYAD A.、KOMUSIEWICZ, CHRISTIAN、NIEDERMEIER, ROLF 和 SORGE, MANUEL。《超图细分绘图中的孪生》。图绘制与网络可视化会议论文集（GD 2016）。第9801卷。计算机科学讲义。施普林格，2016年，67–80。DOI：10 . 1007 / 978 - 3 - 319 - 50106-2_6 4.[vGKvK*17] Van GOETHEM, ARTHUR、KOSTITSYNA, IRINA、van KREVELD, MARC J.、MEULEMANS, WOUTER、SONDAG, MAX 和 WULMS, JULES。《画家问题：用有色连通多边形覆盖网格》。图绘制与网络可视化会议论文集（GD'17）。第10692卷。计算机科学讲义。施普林格，2017年，492–505。DOI：10.1007/978-3-319-73915-1_38 4.[WDN23] WALLINGER, MARKUS、DOBLER, ALEXANDER 和 NÖLLENBURG, MARTIN。《LinSets.zip：压缩线性集合图》。IEEE 可视化与计算机图形学汇刊 29.6（2023年），2875–2887。DOI：10.1109/TVCG.2023.3261934 4.[WEK*12] WAGEMANS, JOHAN、ELDER, JAMES H、KUBOVY, MICHAEL、PALMER, STEPHEN E、PETERSON, MARY A、SINGH, MANISH 和 von der HEYDT, RÜDIGER。《视觉感知中的格式塔心理学百年：I. 知觉分组与图形—背景组织》。Psychological bulletin 138.6（2012年），1172。DOI：10.1037/a0029333 4.[WEN21] WDCF, EMIRR, THE 和 NIKNAKS93。Supranational eu- ropean bodies-en.svg。在线；访问于2022年2月19日。2021年。URL：
h
t
t
p
s://
upload.wiki
media.org/wikipedia/commons/archive/6/6a/20220310193917%21Supranational_ European_Bodies.svg 9。

[CHvKS10] CABELLO, SERGIO, HAVERKORT, HERMAN J ., van KREVELD, MARC J., and SPECKMANN, BETTINA. “比例符号地图的算法方面”。*Algorithmica* 58.3 (2010), 543–565. DOI: 10.1007/s00453-009-9281-8 6。

[CPC09] COLLINS, CHRISTOPHER, PENN, GERALD,and CARPENDALE, SHEELAGH. “气泡集：通过叠加于现有可视化上的等值线揭示集合关系”。 *IEEE* 可视化与计算机图形学汇刊 15.6 (2009), 1009–1016. DOI: 10.1109/TVCG.2009.122 5。

[CR05] CHOW, STIRLING and RUSKEY, FRANK. “朝着绘制面积比例欧拉图的一般解决方案迈进”。 电子理论计算机科学札记 134 (2005), 3–18 4。

[DDKW24] DEPIAN, THOMAS、DOBLER, ALEXANDER、KERN, CHRISTOPH 和 WULMS, JULES。“彩色直角网格中的角最小化”。《国际算法与计算会议及研讨会论文集》（WALCOM'24）。第14549 卷。《计算机科学讲义》。施普林格，2024年，134–148。DOI：10.1007/978-981-97-0566-5_11 4.[Grü75] GRÜNBAUM, BRANKO。“维恩图与集合的独立族”。《数学杂志》48.1（1975），12–23 4.[JWKN21] JACOBSEN, BEN、WALLINGER, MARKUS、KOBOUROV, STEPHEN G. 和 NÖLLENBURG, MARTIN。“MetroSets：将集合可视化为地铁地图”。IEEE 可视化与计算机图形学汇刊 27.2（2021年），1257–1267。DOI：10 . 1109 / TVCG . 2020 . 3030475 4.[KGWD22] KEHLBECK, REBECCA、GÖRTLER, JOCHEN、WANG, YUN- HAI 和 DEUSSEN, OLIVER。“SPEULER：语义保持欧拉图”。IEEE 可视化与计算机图形学汇刊 28 3114834 4, 5 1 2022 433 442 10 1109 2021 . ( ), – . DOI: . /TVCG. . .[KLS07] KIM, BO HYOUNG、LEE, BONGSHIN 和 SEO, JINWOOK。“利用置换矩阵和扇形图可视化集合一致性”。《与计算机交互》19.5-6（2007），630–643。DOI：10.1016/ J.INTCOM.2007.05.004 4.[Kra94] KRATOCHVÍL, JAN。“一个特殊的平面可满足性问题及其 NP 完全性的推论”。《离散应用数学》52.3（1994），233–252 4.[KvKS08] KAUFMANN, MICHAEL、van KREVELD, MARC J. 和 SPECKMANN, BETTINA。“超图的细分绘图”。《图绘制》（GD'08）。第 5417 卷。《计算机科学讲义》。施普林格，2008，396–407。DOI：10 . 1007 / 978 - 3 - 642 - 00219-9 39 4.[LGS*14] LEX, ALEXANDER、GEHLENBORG, NILS、STROBELT, HEN- DRIK、VUILLEMOT, ROMAIN 和 PFISTER, HANSPETER。“UpSet：相交集合的可视化”。IEEE 可视化与计算机图形学汇刊 20.12（2014），1983–1992。DOI：10.1109/TVCG. 2014.2346248 4.[MR14] MICALLEF, LUANA 和 RODGERS, PETER。“eulerForce：欧拉图的力导向布局”。《可视语言与计算杂志》25.6（2014），924–934。DOI：10.1016/j.jvlc.2014.09.002 4.

[MSS21] MEULEMANS, WOUTER, SONDAG, MAX, 和 SPECKMANN, BETTINA。《用于相干网格地图的简单流水线》。IEEE 可视化与计算机图形学汇刊 27.2（2021年），1236–1246。DOI：10.1109/TVCG.2020.3028953 4.[PAQ18] PÉREZ-SILVA, JOSÉ G、ARAUJO-VOCES, MIGUEL 和 QUESADA, VÍCTOR。《nVenn：广义准比例维恩图和欧拉图》。Bioinformatics 34.13（2018年），2322–2324。DOI：10.1093/bioinformatics/bty109 4.[PKS*23] PAETZOLD, PATRICK、KEHLBECK, REBECCA、STROBELT, HENDRIK、XUE, YUMENG、STORANDT, SABINE 和 DEUSSEN, OLIVER。《RectEuler：使用矩形可视化相交集合》。Computer Graphics Forum，第42卷，第3期。2023年，87–98。DOI：10.1111/CGF.14814 2–7、9、13、15、16.[RD10] RICHE, NATHALIE HENRY 和 DWYER, TIM。《解开欧拉图》。IEEE 可视化与计算机图形学汇刊 16.6（2010年），1090–1099。DOI：10.1109/TVCG.2010.210 2、4–6.[RWB*23] ROTTMANN, PETER、WALLINGER, MARKUS、BONERATH, ANNIKA、GEDICKE, SVEN、NÖLLENBURG, MARTIN 和 HAUNERT, JAN-HENRIK。《MosaicSets：将集合系统嵌入网格图》。IEEE 可视化与计算机图形学汇刊 29.1（2023年），875–885。DOI：10.1109/TVCG.2022.3209485 2–6、14.[RZF08] RODGERS, PETER、ZHANG, LEISHI 和 FISH, ANDREW。《通用欧拉图生成》。国际图示理论与应用会议。施普林格。2008年，13–27。DOI：10.1007/978-3-540-87730-1 6 4.[SMDS14] SADANA, RAMIK、MAJOR, TIMOTHY、DOVE, ALISTAIR D. M. 和 STASKO, JOHN T。《OnSet：大规模二进制集合数据的可视化技术》。IEEE 可视化与计算机图形学汇刊 20.12（2014年），1993–2002。DOI：10.1109/TVCG.2014.2346249 4.[SRHZ10] STAPLETON, GEM、RODGERS, PETER、HOWSE, JOHN 和 ZHANG, LEISHI。《归纳生成欧拉图》。IEEE 可视化与计算机图形学汇刊 17.1（2010年），88–100。DOI：10.1109/TVCG.2010.28 4.[SS69] STANFORD, GENE 和 STANFORD, BARBARA DODDS。《通过游戏学习讨论技巧》。（1969年）3、15.[vBKK*16] Van BEVERN, RENÉ、KANJ, IYAD A.、KOMUSIEWICZ, CHRISTIAN、NIEDERMEIER, ROLF 和 SORGE, MANUEL。《超图细分绘图中的孪生》。图绘制与网络可视化会议论文集（GD 2016）。第9801卷。计算机科学讲义。施普林格，2016年，67–80。DOI：10 . 1007 / 978 - 3 - 319 - 50106-2_6 4.[vGKvK*17] Van GOETHEM, ARTHUR、KOSTITSYNA, IRINA、van KREVELD, MARC J.、MEULEMANS, WOUTER、SONDAG, MAX 和 WULMS, JULES。《画家问题：用有色连通多边形覆盖网格》。图绘制与网络可视化会议论文集（GD'17）。第10692卷。计算机科学讲义。施普林格，2017年，492–505。DOI：10.1007/978-3-319-73915-1_38 4.[WDN23] WALLINGER, MARKUS、DOBLER, ALEXANDER 和 NÖLLENBURG, MARTIN。《LinSets.zip：压缩线性集合图》。IEEE 可视化与计算机图形学汇刊 29.6（2023年），2875–2887。DOI：10.1109/TVCG.2023.3261934 4.[WEK*12] WAGEMANS, JOHAN、ELDER, JAMES H、KUBOVY, MICHAEL、PALMER, STEPHEN E、PETERSON, MARY A、SINGH, MANISH 和 von der HEYDT, RÜDIGER。《视觉感知中的格式塔心理学百年：I. 知觉分组与图形—背景组织》。Psychological bulletin 138.6（2012年），1172。DOI：10.1037/a0029333 4.[WEN21] WDCF, EMIRR, THE 和 NIKNAKS93。Supranational eu- ropean bodies-en.svg。在线；访问于2022年2月19日。2021年。URL：
h
t
t
p
s://
upload.wiki
media.org/wikipedia/commons/archive/6/6a/20220310193917%21Supranational_ European_Bodies.svg 9。

[WRD21] WYBROW, MICHAEL、RODGERS, PETER 和 DIB, FADI K.《按面积比例绘制的椭圆欧拉图（Edeap）》。BMC 生物信息学 22.1（2021年），214。DOI：10 . 1186 / s12859 - 021 - 04121-8 4.[YDG* 15] YOGHOURDJIAN, VAHAN、DWYER, TIM、GANGE, GRAEME、KIEFFER, STEVE、KLEIN, KARSTEN 和 MARRIOTT, KIM。《分组网络的高质量超紧凑网格布局》。IEEE 可视化与计算机图形学汇刊 22.1（2015年），339–348。DOI：10.1109/TVCG.2015.2467251 4.[ZYC* 24] ZHOU, YUXING、YANG, WEIKAI、CHEN, JIASHU、CHEN, CHANGJIAN、SHEN, ZHIYANG、LUO, XIAONAN、YU, LINGYUN 和 LIU, SHIXIA。《簇感知网格布局》。IEEE 可视化与计算机图形学汇刊 30.1（2024年），240–250。DOI：10 . 1109/TVCG.2023.3326934 4。

BlockSets：含大型元素的集合的结构化可视化——补充材料

Neda Novakova、Veselin Todorov、Steven van den Broek、Tim Dwyer 和 Bettina Speckmann

1. 补充图。



图19： *RectEuler* *[PKS*23]* 谋杀谜案的图*3* 以 *BlockSets* 风格渲染。为生成此渲染结果，我们对 *RectEuler* 的整数线性规划进行了调整，使其不使用集合标签或集合圆形标记，也不将属于同一集合的集合的陈述分组。

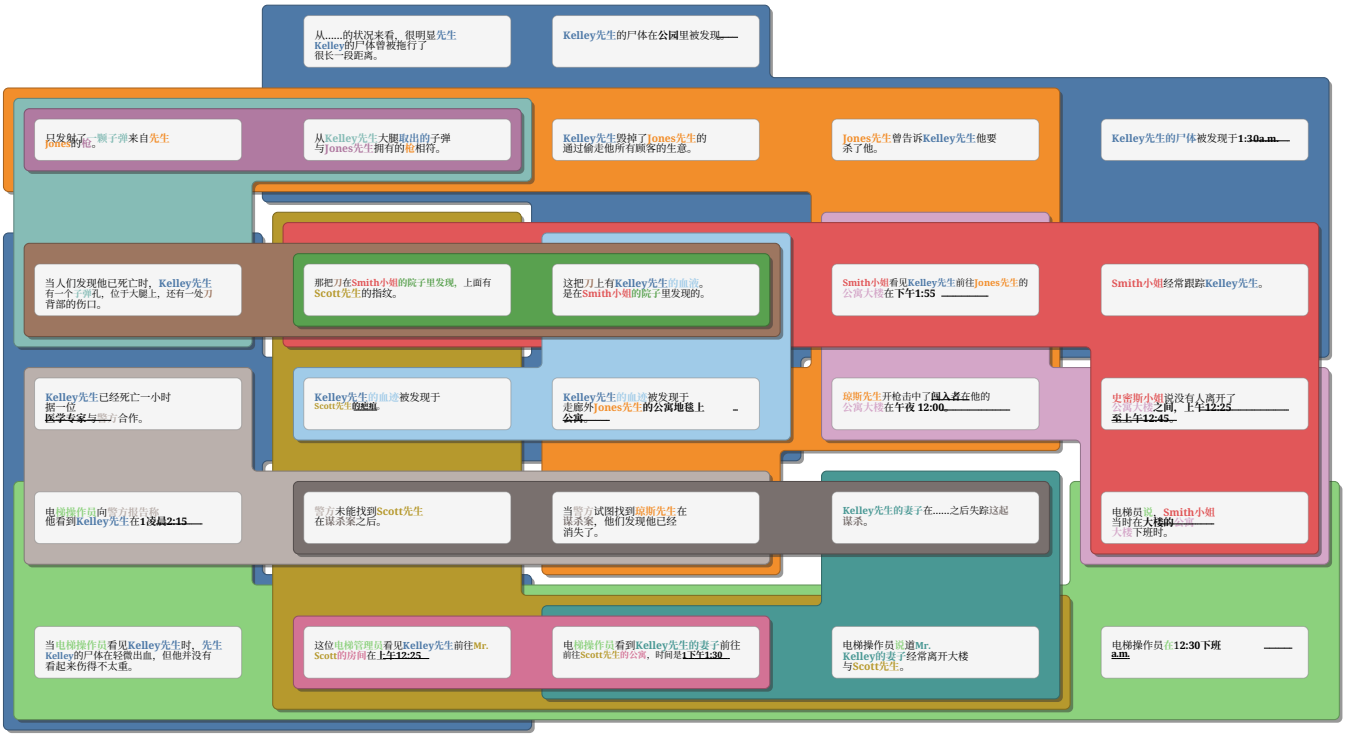


图20： *MosaicSets [RWB*23]* 图3中的谋杀谜案解答图3以 *BlockSets* 风格渲染。

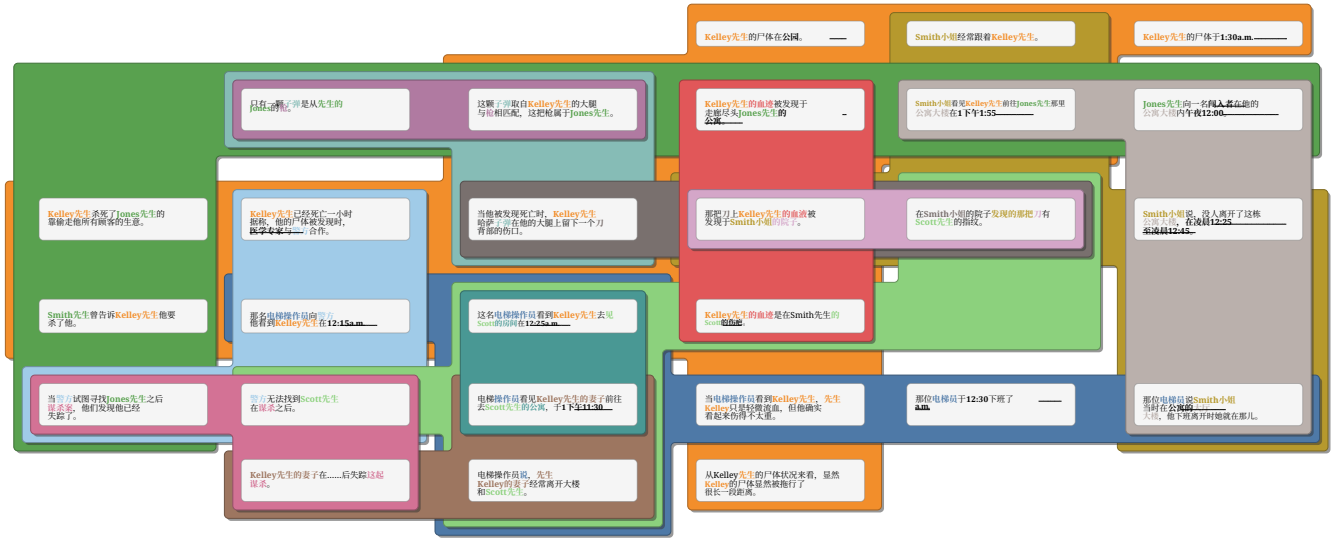
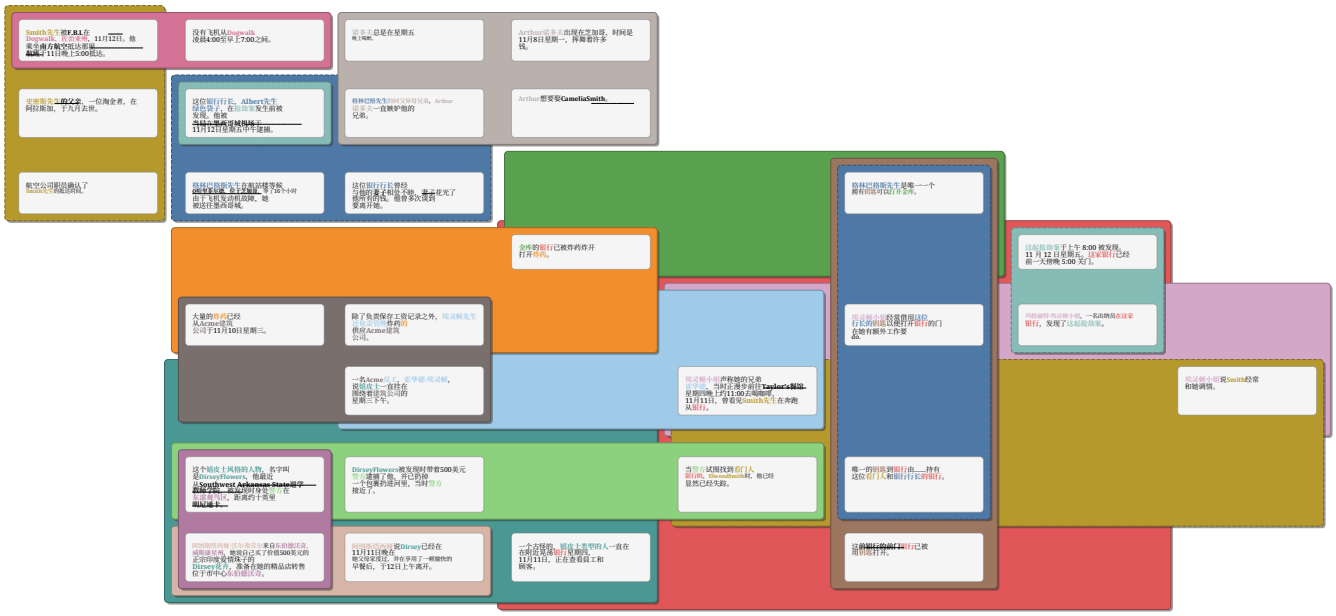
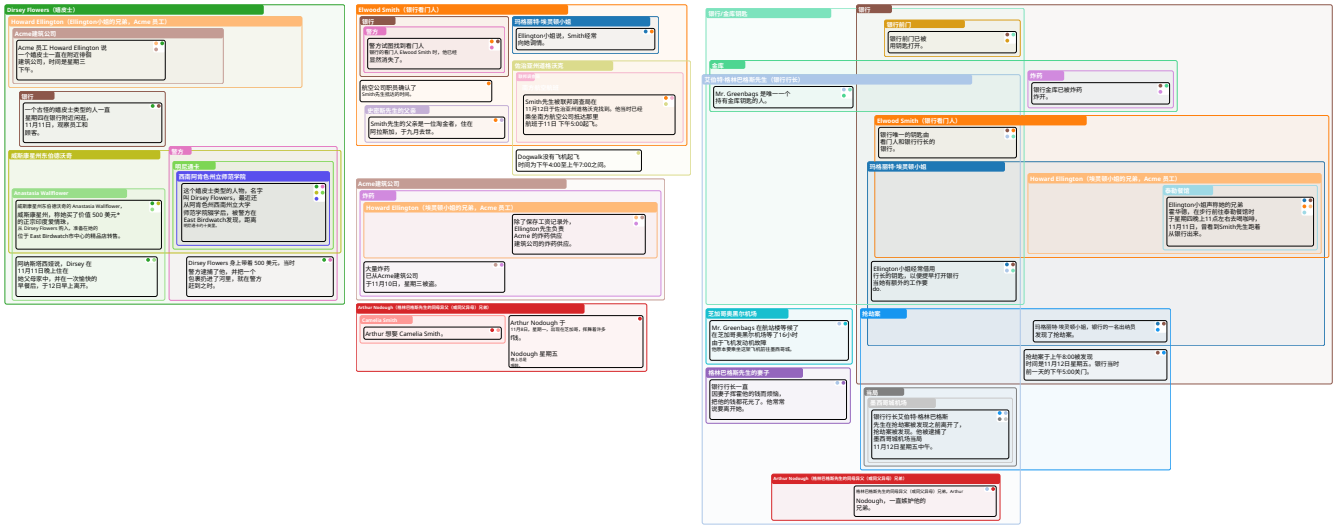


图21： 图3中的 *BlockSets* 谋杀谜案解答图3采用正交凸形状。



(a)带矩形的 *BlockSets*。只有两个集合重复。



(b)*RectEuler* [*PKS*23*] 限时一分钟。八个集合重复，其中两个重复了三次。

图22： 在银行抢劫悬疑谜题上对比 *BlockSets* 和 *RectEuler* 的拆分算法 [*SS69*]。参见 图*23* 其中展示了一个不进行拆分的 *RectEuler* 版本。

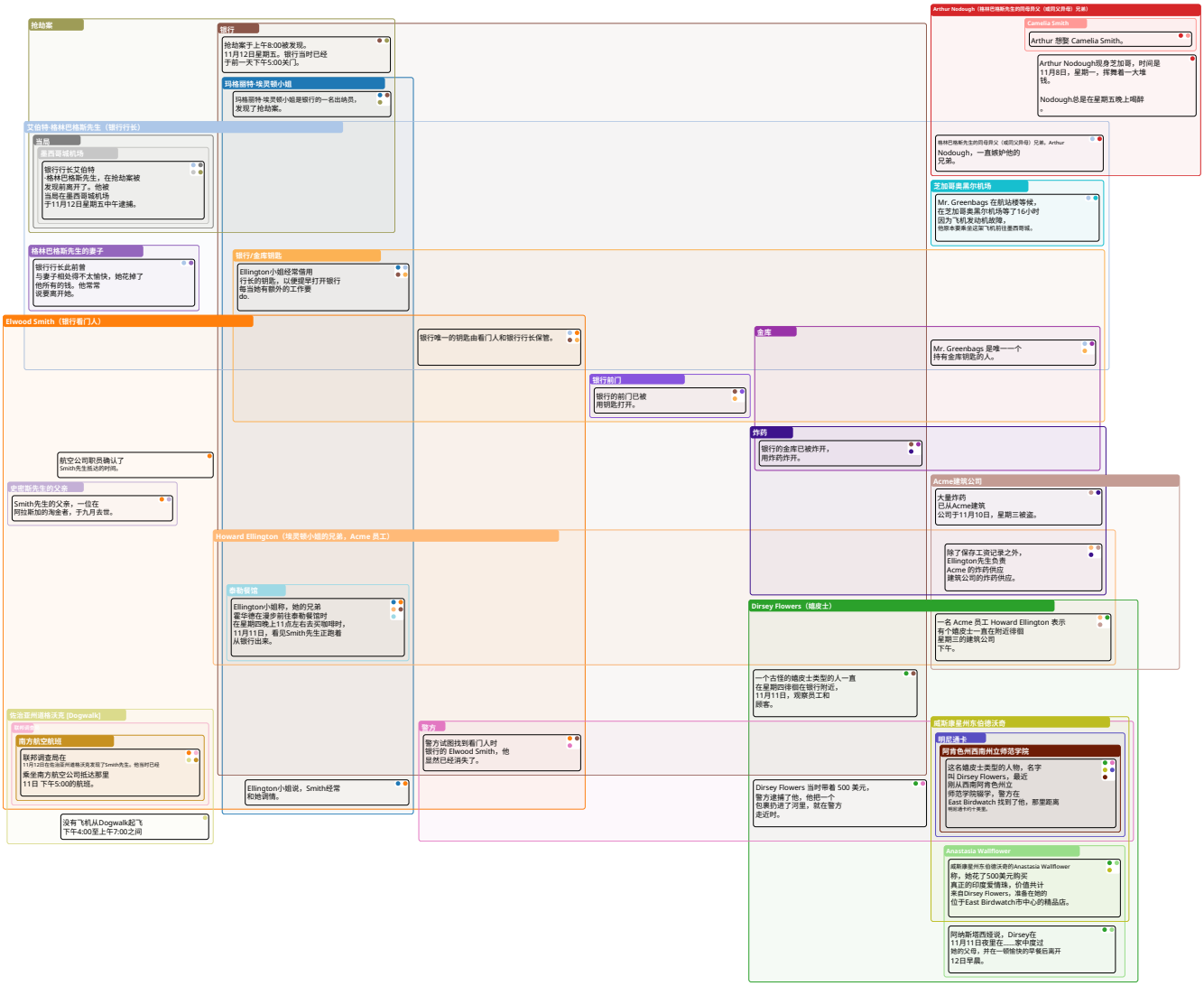


图23： *RectEuler* [PKS*23] 在设有*100*分钟时限的银行抢劫悬疑谜题中的表现。*RectEuler* 仅在找不到可行解时才进行拆分；*BlockSets* 还会在解不够紧凑时进行拆分。