介绍最大公共大图问题

Kyle Burns MicheleSevegnani CiaranMcCreesh James Trimble计算机科学学院 格拉斯哥大学 英国格拉斯哥 kyleburnsgla@gmail.com michele.sevegnani@glasgow.ac.uk ciaran.mccreesh@glasgow.ac.uk james.trimble@yahoo.co.uk

双图反应系统为对智能体之间的空间关系和非空间关系进行建模提供了强大而灵活的数学框架，在智能技术、网络、传感器系统和生物学等领域具有实际应用。尽管理论上双图支持通过模拟并比较相应的最小上下文迁移系统来识别双模拟智能体，但目前尚无已知算法能够计算两个双图之间的最大共享结构，而这对于确定给定代理状态可能发生的迁移集合至关重要。本文定义了最大公共双图问题，并提出对 McSplit 最大公共诱导子图算法的改编，用于计算两个大图状态之间的最大公共双图。我们的方法为在基于大图的工具中支持双模拟检查开辟了道路；这些工具已应用于其他建模范式，以简化、优化和验证模型。

1 引言

双图反应系统（BRSs）由 Milner 首次提出 [20]，是一种通用的数学形式体系，能够借助双层大图数据结构来表示实体之间的空间关系和非空间关系。给定一个 BRS 模型，其中包含初始智能体 G 以及一组形式为 $R \to R'$的反应（重写）规则，大图工具包（如 *BigraphER* [24]）便能通过计算模型的原始迁移系统，模拟智能体随时间可能发生的演变。这一过程的核心是高效的底层匹配算法，它能找出G 中R 的所有发生位置，以便在每一步中替换R'；这个过程称为大图匹配。双图反应系统的实际应用包括模拟多个领域中的过程，例如物联网设备中的传感器 [25],、网络通信 [8],、生物现象 [17],、智能建筑的安全 [1] 以及无人机的群体编程 [10, 13]。

除了原始迁移系统外，理论上，双图反应系统还可以计算模型的最小上下文标记迁移系统（MCTS）。这一过程需要找出反应规则红ex$R$ 与代理状态G 之间最大的结构重叠，而非进行完全匹配，从而让更广泛的环境提供缺失的最小上下文，以实现完全匹配（并进行后续替换）。这实际上模拟了智能体如何在任意上下文中演化，而非处于真空环境中。双模拟是一种可用于模型优化和验证的性质；对于大图代理，可以通过确定两个模型的 MCTS 是否等价来验证双模拟，因为这能保证它们在任何可能的上下文中的行为完全相同。现有工具目前只能对原始迁移系统建模，无法有力地保证这一性质。

在这项工作中，我们给出了最大公共大图（MCB）算法的定义，该算法描述了如何找到两个大图状态之间最大的重叠部分，这是找出给定智能体与规则之间所有最小上下文所必需的前置步骤。我们提供了一种将大图编码为图的方法

L. Lambers 和 O. Semeráth（编）：GCM 2025，EPTCS 440，2026年，第 13–35 页，doi:10.4204/EPTCS.440.3

© Burns 等人。本作品依据知识共享署名许可协议授权。

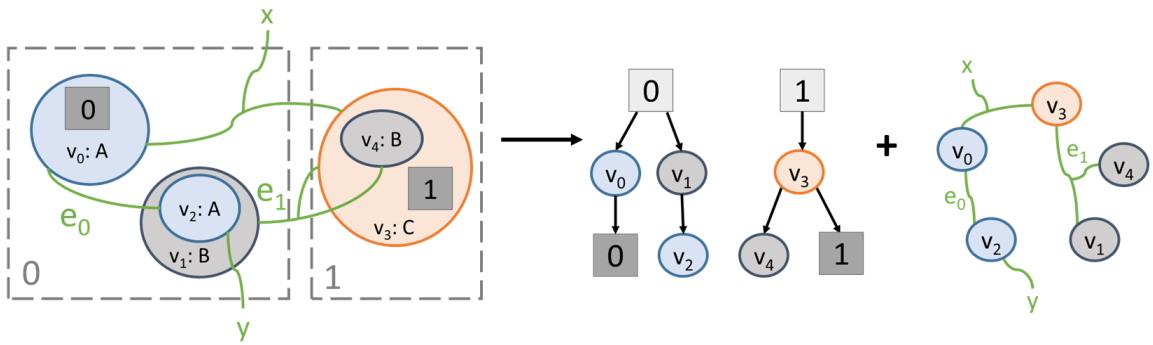


图 1：具体大图示例，其中包含实体、实体的控制，以及它们之间的空间关系和非空间关系，并展示了如何将大图拆分为位置图和链接图两个组成部分。

，从而可以使用*McSplit* MCIS 算法 [19], 求解 MCB。该方法基于早期的研究；早期研究表明，将类似的编码输入子图同构求解器，可以非常高效地完成完整的大图匹配 [2]。

致谢。 本研究的扩展版本已作为以下博士学位论文的一部分发表：用于大图匹配的高效且可扩展算法，Kyle Burns，格拉斯哥大学 [5]。

2 双图

大图 G 是一种类图数据结构，包含两个相互独立但共享同一组实体（节点）的组成部分：位置图，一种表示实体之间空间关系（例如设备位于房间内）的有向森林；以及链接图，一种表示非空间关系（即设备连接到本地网络）的超图。大图既可以用代数方式表示，也可以用图形方式表示；在本研究中，我们尽可能采用图形表示法。

图 1 展示了一个大图及其位置图和链接图组成部分的图形示例。位置图关系以父节点包围其子节点的方式表示，而链接图邻接关系则以绿色链接表示。所有实体还都标有（类型化的）控制 值，在所示示例中以颜色表示；每个控制都关联一个元数 整数值，用来规定具有该控制的每个实体必须具有多少个端口 （链接图邻接关系）。位置图有n 个区域 和 m 个站点，分别对应顶层和底层位置，在图中以带编号的方框表示。它们是对当前大图状态上方和下方未知（或为空）的附加结构的抽象，并指明可以在何处复合另一个位置图。类似地，向当前大图上方和下方延伸的未连接链接称为其外名 Y 集和内名 X集，表示可以在何处复合另一个链接图。若链接连接到内名或外名，则称其为开放 ；若它仅在实体之间连接，则称其为闭合。

位置集合m和内名X构成大图的内面，记作 $\langle m,\ X\rangle$。反之，区域n 和外名Y 构成其外面，记作 $\langle n,\ Y\rangle$。两者合起来描述大图的接口，写作G： $\langle m,\ X\rangle \to \langle n,\ Y\rangle$。位置和区域

使用非负整数的有序集进行标识，例如$n = 3$ 表示区域集合为$\{0,1,2\}$。因此，可以推导出图1所示大图的接口为G： $\langle 2,\ \{y\}\rangle \to \langle 2,\ \{x\}\rangle$。所示大图也是一个具体的大图，这意味着所有实体 $\{v_0, \ldots\ldots、v_4\}$ 以及闭合链接 $\{e_0,e_1\}$，连同接口组件，都被赋予唯一标识符。它还是一个实心大图，这意味着内面与外面之间不存在直接邻接（例如连接区域和位置的边），任意两个内面组件都不是兄弟节点，并且每个外面组件至少有一个邻接。要将重写规则$R \to R'$ 应用于大图G，R 和G 始终都是实心的。

通过大图的接口，可以将较大的具体大图 $G = A \circ B$ 由较小的具体大图分量A 和B 构造而成：将A 的内面与B的外面相连，并连接同名项。另一种复合形式是两个大图$G = A \otimes B$ 的张量积，它只需将两个结构并排放置。这些构成了双图反应系统中发生、分解和重写规则的基础。现在，我们给出大图的正式代数定义，以便介绍双图反应系统及迁移系统的类型。为补充定义，我们将用于标识实体、控制和闭合链接的名称集定义为互不相交的无限集 $\mathcal{V}$， $\mathcal{X}$ 和 $\mathcal{E}$。

2.1 双图定义

定义 2.1（具体位置图）。具体位置图 $\mathrm{B} = (\mathrm{V}_B, \mathrm{ctrl}_B, \mathrm{prnt}_B) : \mathrm{m} \to \mathrm{n}$ 是一个三元组，其内面为 m，外面为 n，分别表示 m 个位置和 n 个区域。B 有一个实体有限集 $\mathrm{V}_B \subset \mathcal{V}$、一个控制映射 $\mathrm{ctrl}_B : \mathrm{V}_B \to \mathcal{K}$，以及一个父映射 $\mathrm{prnt}_B : \mathrm{m} \uplus \mathrm{V}_B \to \mathrm{V}_B \uplus \mathrm{n}$，该父映射是无环的，即对于任意 $\mathrm{v} \in \mathrm{V}_B$，$(\mathrm{v}, \mathrm{v}) \notin \mathrm{prnt}_B^+$，其中 prnt_B^+是 prnt 的传递闭包。

定义 2.2（具体链接图）。具体链接图 $\mathrm{B} = (\mathrm{V}_B, \mathrm{E}_B, \mathrm{ctrl}_B, \mathrm{link}_B) : \mathrm{X} \to \mathrm{Y}$ 是一个四元组，其内名称集 $\mathrm{X} \subset \mathcal{X}$ 和外名称集 $\mathrm{Y} \subset \mathcal{X}$ 均为有限集。B 包含实体的有限集 $\mathrm{V}_B \subset \mathcal{V}$ 和链接的有限集 $\mathrm{E}_B \subset \mathcal{E}$、控制映射 $\mathrm{ctrl}_B : \mathrm{V}_B \to \mathcal{K}$ 以及链接映射 $\mathrm{link}_B : \mathrm{X} \uplus \mathrm{P}_B \to \mathrm{E}_B \uplus \mathrm{Y}$，其中 $\mathrm{P}_B \,\mathrm{def}= \{(\mathrm{v}, \mathrm{i}) \mid \mathrm{v} \in \mathrm{V}_B,\ 0 \le \mathrm{i} < \mathrm{ar}(\mathrm{ctrl}_B(\mathrm{v}))\}$ 是 B 的端口集，而 $\mathrm{ar}(\mathrm{ctrl}_B(\mathrm{v}))$ 是 v 的控制项的元数值。闭合链接是指定义域限于 P_B 且像属于 E_B的链接；否则，它们就是开放链接。此外，空闲边是定义域限于 /0 的链接，即没有源可供指向。

定义 2.3（具体大图）。具体大图 $\mathrm{B} = (\mathrm{V}_B, \mathrm{E}_B, \mathrm{ctrl}_B, \mathrm{prnt}_B, \mathrm{link}_B) : \langle \mathrm{m}, \mathrm{X}\rangle \to \langle \mathrm{n}, \mathrm{Y}\rangle$由具体位置图 $\mathrm{B}^{\mathrm{P}} = (\mathrm{V}_B, \mathrm{ctrl}_B, \mathrm{prnt}_B) : \mathrm{m} \to \mathrm{n}$ 和具体链接图 $\mathrm{B}^{\mathrm{L}} = (\mathrm{V}_B, \mathrm{E}_B, \mathrm{ctrl}_B, \mathrm{link}_B) : \mathrm{X} \to \mathrm{Y}$ 组成。B 的内接口和外接口分别为 $\langle \mathrm{m}, \mathrm{X}\rangle$ 和 $\langle \mathrm{n}, \mathrm{Y}\rangle$。具体大图的支撑大小 $|\mathrm{B}|$ 为 $|\mathrm{V}_B \uplus \mathrm{E}_B|$。具体大图由具体位置图和具体链接图组合而成，两者共享同一实体集 V_B。

定义 2.4（具体出现）。 给定两个具体大图 A 和 B，如果存在某个上下文 大图 C和参数 大图 D，使得 $A = C \circ (B \otimes \text{恒等性}) \circ D$， 则称$B$在$A$ 中出现；其中恒等性 是恒等大图——一种不含实体的大图实例，仅包含从 $\langle m, X\rangle$ 直接连接到 $\langle n, Y\rangle$的抽象和面；它允许位置图边和链接图超边在必要时直接连接C 与 D，以确保接口兼容性。

2.2 BRS 转移系统

BRS 模型由一个初始大图代理A 和一组形式为$R \to R'$的反应规则组成。当G 中存在R 的某次发生，使得$A = C \circ (R \otimes id) \circ D$，其中$C$ 和D 是某些上下文和参数大图

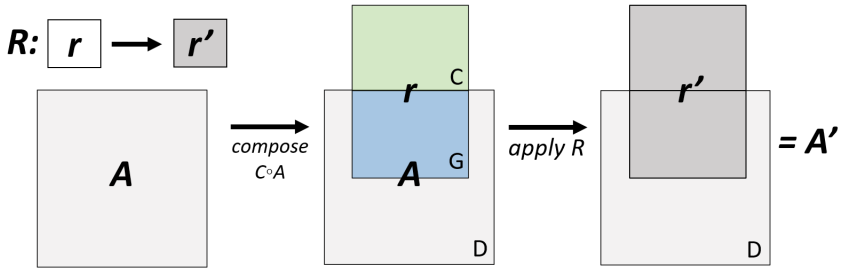


图 2：标记迁移系统中迁移的高层表示。首先将来自智能体 A 环境的上下文 C 组合起来，生成 $C \circ A$，使可约表达式 $r = C \circ G$ 出现在智能体中，并可应用反应规则 $R : r \to r\{\text{v25>}$。G 是同时出现在 r 和 A 中的重叠结构，而参数大图 D 则是 A 中不重叠的剩余部分。

C 和 D，那么通过将 R 替换为 R'，即可实现状态变化（例如，设备从一个房间移动到另一个房间），并生成结果智能体 $G' = C \circ (R' \otimes id) \circ D$。这一过程的关键在于找出智能体中 R 的所有发生，这被称为大图匹配 问题。当存在匹配时，可将其表示为一种映射：把模式规则中的所有支撑元素（实体和闭合链接）映射到智能体中相匹配的元素。这种映射也称为嵌入，即 R 在 A [9] 中的嵌入。大图匹配是一个 NP困难问题，双图工具包通常依赖 SAT求解 [23, 22], 约束求解 [9, 2]工具来完成此任务。另一种已知方法是将双图转换为图格式，以便使用现有的图变换工具解析，并结合子图同构和单 [12] [14] 或双 [11] 推出构造来完成整个重写过程。

通过对智能体递归穷举地应用和替换反应规则，可以生成模型的原始迁移系统。它是一种有向图结构，其中节点表示智能体状态，即所有可能到达的结果状态；边则表示通过应用反应规则从一个状态迁移到另一个状态。原始迁移系统的建模能力有限，因为它假设被模拟的智能体必须孤立存在，不受任何额外上下文影响，而不是作为更广阔周围环境中的一个分量。现实中，我们希望考虑与智能体相连的未知实体和链接，因为它们可能影响状态随时间演化的行为。在标记迁移系统（LTS）中，迁移表示为 $A \xrightarrow{f} A'$，其中 f 是提供环境结构的上下文大图。应用反应规则之前，上下文会与智能体 A 复合，从而实现依赖外部结构的迁移。原始迁移系统可视为受限形式的 LTS，其中 $f = /0$，这意味着只有在完整匹配已经存在时才能应用反应规则。LTS 通过纳入环境影响提供了表达能力更强的建模框架，但其一般性也带来了挑战：可能存在无限多个可用上下文 f，且上下文在 A 中的复合位置未明确规定。为解决这一问题，Milner [20] 提出了上下文迁移系统（CTS），其迁移标注为 $A \xrightarrow{(\hat{f}, j)} A'$，其中 j 指定接口组件的映射，f 据此与 A 复合。还可以进一步引入最小迁移的概念进行细化，即 f 仅提供发生反应所需的最少结构。最小上下文迁移系统（MCTS）只包含此类迁移，足以捕捉智能体受环境影响可能到达的任何状态。图 2 展示了一个在 LTS 中应用最小上下文的示例。

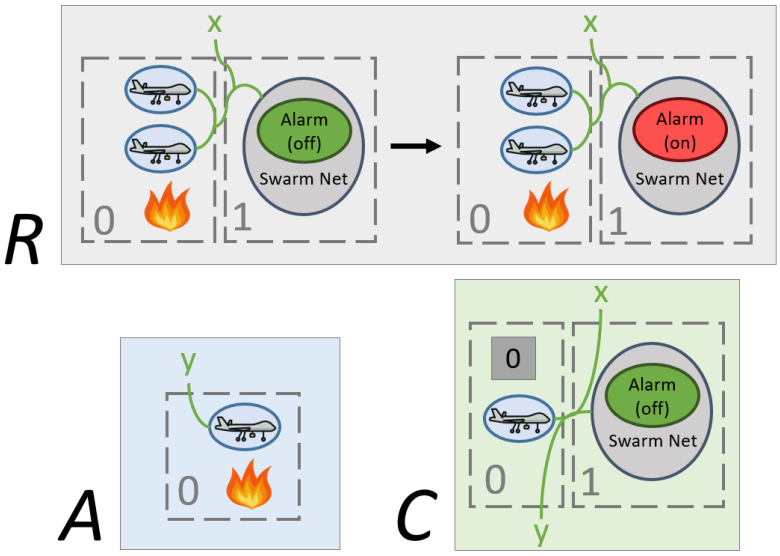


图3：使用最小上下文转移系统对用于野火探测的无人机群进行建模时，迁移的一个实际应用示例。当向智能体 A提供上下文C 时，即可应用反应规则R 。

实现 MCTS 模拟的首要动机，是能够确保模型中智能体的双模拟相似性。两个智能体状态 A 和B之间的双模拟记作$A \sim B$，它是一种对称关系，表示两个智能体在所有迁移下都表现出无法区分的行为，因此无论结构上有何差异，都可以视为功能等价 [20]。双模拟是一种强大的模型检验工具，在数据库重构 [28], 、图处理 [21], 和强化学习 [16], 等领域有实际应用，因为它能够简化、优化并验证系统分量。在双图反应系统（BRSs）中，还可以通过检查两个智能体各自的 MCTS 树结构是否等价来保证这种一致性，而这一检查可在多项式时间内完成。这一方法有望带来实际应用，例如模型简化（用较小的双模拟状态替换复杂分量）、行为验证（确保更新后没有出现意外的行为变化）和容错（将被认定为双模拟的分量用作备份）。

图3给出了一个更实际的最小上下文转移系统示例，其中提供了一个用于模拟无人机群体行为的部分 BRS 模型。反应规则R 定义了这样一种行为：当两架连接到无人机群网络的无人机位于火灾上方时，将激活该网络的警报。然而，如果我们希望模拟从单架无人机发现火灾开始的所有可能结果（如智能体A所建模），则R 无法在原始转移系统中应用，因为原始转移系统无法考虑超出A范围的更广泛环境（因而也无法考虑附近的其他无人机）。不过，最小上下文转移系统能够对另一架无人机和无人机群网络处于通信范围内的可能性进行建模，因此可以通过向A 提供最小上下文C 来对此可能性进行建模。

为了在实践中模拟一个最小上下文转移系统，需要一种方法来为每个允许的转移找到最小上下文C_m。这可以通过寻找反应规则r与代理状态A之间的最大重叠来计算，使得$r = C_m \circ G$且 $A = C_r \circ G \circ D_r$，由此生成结果双图$C_m \circ A = (C_m \otimes C_r) \circ (r \otimes id) \circ D_r$。图 2 展示了一个示例，可以看到$C_M$ 可通过先找到G，再从R中分解G 的所有元素得到（为求简洁，此处未以图形方式展示C_r ）。这遵循了类似的原理，即使用最大公共子图算法来解决图编辑距离问题 [3]。

3 最大公共（诱导）子图

最大公共子图*maximum commonsubgraph* (MCS)问题是一种 NP完全优化问题，旨在寻找一种图结构G_S，它与两个给定输入图G_1 和G_2, 中的子图同构，并且不存在任何更大的（按某种度量标准）子图满足相同条件 [18]。MCS 主要有两种形式：最大公共诱导子图（MCIS），关注的是最大化G_1 与G_2, 之间可映射的顶点数量；以及最大公共边子图*maximum common edge subgraph* 问题（MCES），它改用G_S 的边数作为要最大化的规模度量。下文主要关注 MCIS, 因为如后文所述，它能更准确地反映双图的行为。我们将其正式定义如下。

定义 3.1（最大公共诱导子图）。给定两个输入图 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$，公共诱导子图 $G_S = (V_S, E_S)$ 是指存在一对单射映射 $f_1 : V_S \to V_1$ 和 $f_2 : V_S \to V_2$,，使顶点和边能够映射，并且 $(u, v) \in E_S \iff (f_1(u), f_1(v)) \in E_1$，且 $(u, v) \in E_S \iff (f_2(u), f_2(v)) \in E_2$。

G_S在不存在满足上述条件且还满足 $|V'_S| > |V_S|$ 的其他解 $G'_S = (V's, E'_S)$ 时，称其为最大值 。

目前，MCIS 最先进的方法是 *McSplit* 分支定界算法。该算法以类似约束求解器的方式对问题建模，将图V_1 中的顶点赋值给V_2, ；当无法再进行有效赋值时，则进行回溯 [19]。McSplit 使用标签类, 即由尚未赋值、且与已匹配顶点具有相同邻接模式的顶点对组成的集合，并在每个赋值步骤中根据邻域兼容性对其进行细化。这些类通过三个数组高效维护：两个顶点标识符排列（V_1 和V_2）以及一个标签类描述符数组LC，从而无需存储显式位集，即可在$O(|V_1| + |V_2|)$ 时间内完成所有细化和回溯。在每个状态下，McSplit 会按如下方式计算可进行的后续赋值数量的紧确上界：

$$bound = |M| + \sum_{l \in LC} min(|(u \in V_1 \wedge LC(u) = l)|, |(v \in V_2 \wedge LC(v) = l)|)$$

如果上界不可能超过当前最佳解，就会剪去该分支。另有 McSplit 的优化实现采用基于 PageRank算法 [7] 的高效启发式策略。研究还证明，McSplit 可扩展以支持更多功能，例如顶点标签/类型、有向边、枚举所有解以及强制连通性；后文将说明，这些功能对于支持大图的编码是必需的 [19]。

4 最大公共大图

现在，我们正 式介绍最大公共大图（MCB）问题的概念，定义如下 s.

定义 4.1 （最大公共大图）。 给定两个输入的实心双图G_1 和 G_2, ，集合最大公共双图 $(G_M, M) \in \text{MCB}(G_1, G_2)$中的一个成员是一个包含实心双图$G_M$的有序对，该实心双图满足

$$G_1 = C_1 \circ (id \otimes G_M) \circ D_1,\ G_2 = C_2 \circ (id \otimes G_M) \circ D_2$$

使得不存在其他满足以下条件的大图$G'_{M'}$

$$G_1 = C_3 \circ (id \otimes G'_{M'}) \circ D_3,\ G_2 = C_4 \circ (id \otimes G'_{M'}) \circ D_4,$$
$$(|G'_{M'}| > |G_M|) \vee (G'_{M'} = C_5 \circ G_M \circ D_5,\ C_5 \neq id,\ D_5 \neq id)$$

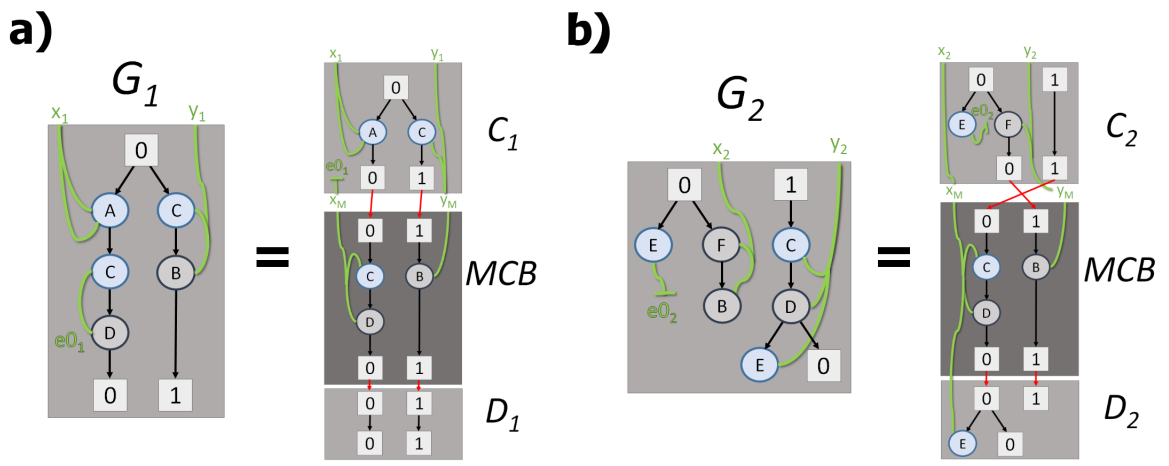


图 4：最大公共二部图问题的一个实例，其中G_1 和G_2 是输入大图，*MCB* 是解。(a) 对G_1 进行分解，以显示*MCB* 作为一个分量存在于该大图中。(b) 对G_2 进行类似的分解，以显示*MCB* 也存在。*C* 和*D* 的端口必须在*MCB*中连接起来，因为这样仍能得到有效解。

分别对于任意上下文和参数双图C_{1-5} 和D_{1-5}。

$M = \{(m_1, m2, ..., m_n)\}$是一个三元组集合$m_k = \{(g_k, a_k, b_k)\}$，表示$g \in G$ 中的实体、端口和闭合链接分别到$a \in G_1$ 和$b \in G_2$ 中对应对象的嵌入映射。我们希望枚举所有此类实例及其对应的映射。

从该定义可以看出，候选解G_M 要被视为最大值，必须满足两个独立的要求：

1. 不存在任何支撑大小更大的其他解二部图G'_M。2. 不存在任何其他解二部图G'_M，使得G_M 出现在G'_M 中，且二者不同构。

第一项要求显而易见：任何违反该要求的候选解，都不能合理地被视为G_1 和G_2之间最大的共享区域。第二项要求确保任何解也必须满足组合意义上的极大性标准——即使满足第一项条件，如果G_M 的任何接口仍可通过进一步复合而合并或闭合，同时又保持为G_1 和G_2的子分量，那么仍可能违反第二项要求。这一额外步骤是必要的，目的是将每个解二部图中包含额外任意区域/位置/链接所产生的理论上无限多种变体限制在有限范围内，同时确保G_M 仍符合极大双图的概念；我们希望“饱和”每个发现的解，使得对G_M 进行任何进一步的非恒等组合，都不再能使其成为有效解。在确定最优嵌入后，通过计算三元关系（G_1, G_2, G_M）的相对推出 (RPO)来处理这一问题。由于双图文献已说明如何在给定一对双图及其元素间映射的情况下计算相对位置 [20], ，我们不再赘述，而是着重介绍如何利用这一方法计算所需映射，将其作为前置步骤。图4展示了一个MCB问题实例，以及G_1 和G_2 的相应分解，以求得公共双图。

类似于将MCIS视为子图同构问题（SIP）判定问题的一种优化问题变体，我们将MCB定义为大图匹配的优化变体。我们还提出，可以将问题实例作为带有额外约束的MCIS来处理，并通过预处理和后处理来应对超边及组合规则带来的复杂性，从而求解MCB；以往的工作也曾采用这种方法，将大图匹配实例编码为SIP问题 [2]。

4.1 性质

通过这一定义，我们可以直观地推断出 MCB 的一些性质，这些性质对于任何实例都必须成立。

- **恒等性**：$(G,M) \in \mathrm{MCB}(G,G)$ 对于任意大图 G 均成立。两个等价大图之间的最大公共大图也将是一个等价关系。当G内部存在对称性时，还可能存在额外的解映射。

- **逆规则**：$(G_M,M) \in \mathrm{MCB}(G1,G2) \longleftrightarrow (G_M,M^{-1}) \in \mathrm{MCB}(G2,G1)$ 对于任意大图对 $G1$ 和$G2$ 均成立。交换输入图的顺序仍应产生相同的解集，但$G1$ 与 $G2$ 之间的所有映射都将被反转。

- **传递性**：解 $(G_M,M) \in \mathrm{MCB}(G3,\mathrm{MCB}(G1,G2))$ 表示三个大图G_{1-3}之间最大的共享区域，与运算顺序无关（传递性）。
- **匹配**：$(S_{\text{大图}} \neq /0) \in$匹配$(P,\ T) \longrightarrow (P,\ S_{\text{大图}}) \in \mathrm{MCB}(P,\ T)$——如果目标图$T$中存在模式双图$P$ 的完整匹配，那么P 也将是二者之间具有相同嵌入的最大公共大图。

5 使用 McSplit 编码求解 MCB

接下来，我们将介绍如何调整 McSplit，以支持将大图编码为有向/带标签标准图。我们假设实例均为非平凡实例，即至少包含一个实体，且不包含空闲边/区域/站点。我们方法的可靠性和完备性的完整证明见附录 A。

首先，我们仅考虑位置图的独立编码，以及对 McSplit 数组结构所做的修改，这些修改使我们能够强制生成遵循双图组合规则的有效解。随后，我们引入链接图展平函数，用于对实体的端口及其关系进行建模，并进一步调整细化过程和奖励函数以支持这些操作。接着，分别计算位置和链接各自的相对位置，以确保组合饱和。图 5 以可视化方式展示了这一过程的概览。

5.1 位置图编码

两个G_1 和G_2 的位置图，都是通过丢弃位置和区域，以及分别与它们相关的所有父关系来编码的。在执行底层的 MCIS 搜索循环时，我们不会立即考虑G_1 和G_2 中这些位置和区域的放置方式，因为无论是否放置位置或区域，都不会使候选最大公共双图中原本有效的一组实体映射失效；在最坏情况下，为了确保结果在其编码形式中总是出现在G_1 和G_2 中，我们只需在构建解二部图时，将一个位置设为所有实体$v \in V_{G_M}$的子节点，并将一个区域设为每个顶层实体的父节点。因此，

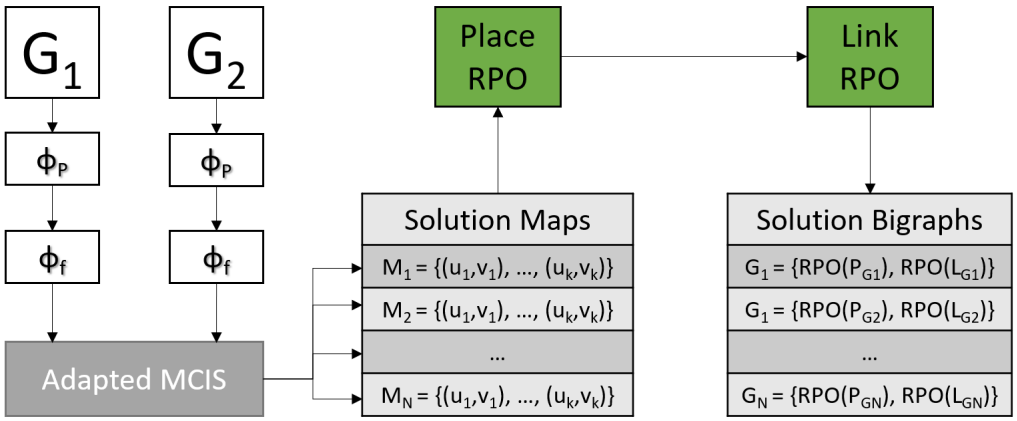


图 5：MCB 求解过程：使用最大公共诱导子图算法，在两个编码后的双图中找到最大的公共区域，然后细化其接口，以生成一组最小绑定的解双图。

我们首先假设，作为基准，这种放宽后的结构——其中所有实体都同时邻接内接口和外接口——将是完成 MCIS 搜索后返回的解二部图的形式，并允许 RPO 函数通过在可能的情况下闭合并合并所有区域和站点来细化解的接口。控制也编码为节点标签，以强制执行相应的双图约束，即相同类型必须相互匹配。

应用编码函数 ϕ_P 后，$G_1 = (V_{G_1},\ ctrl_{G_1},\ prnt_{G_1}) : i \to j$和 $G_2 = (V_{G_2},\ ctrl_{G_2},\ prnt_{G_2}) : k \to l$ 的编码形式如下图对所示：

$$\phi_P(G_1) = \{V_{G_1}, (u,v) \in prnt_{G_1}^{-1} \mid v \neq i, u \neq j\}$$

$$\phi_P(G_2) = \{V_{G_2}, (u,v) \in prnt_{G_2}^{-1} \mid v \neq k, u \neq l\}$$

5.1.1 确保复合有效

为位置图实现 MCB 的下一步，是修改 McSplit，以确保解二部图 G_M 始终符合双图的组合性质，因此能够分别以 $G_1 = C_1 \circ G_M \circ D_1$ 和 $C_2 \circ G_M \circ D_2$ 的形式，作为 G_1 和 G_2 的分量存在。为确保这一点，每对已分配实体都必须彼此直接邻接（即形成连通子图），或者完全传递不相交，即任何一方都不是另一方的后代（即形成张量积）。我们首先考虑用于求解最大公共连通诱导子图的 McSplit 算法。该算法通过仅从与当前解子图至少有一条邻接边的标签类中选择顶点，来确保解是连通的。然而，我们希望以一种方式放宽这一限制，同时仍允许选择不相交的顶点，只要这些顶点无法从任何当前已分配顶点传递到达，也无法传递到达任何当前已分配顶点。在 CSP 格式中，可将以下内容添加到约束列表中：

$$\{\forall u, v \in G_1 \mid \{\forall v' \in prnt_{G_1}^{-1}(v) \mid \texttt{match}(v') = \emptyset\} \wedge (u,v) \in prnt_{G_1}^{+} \wedge \texttt{match}(u) \neq \emptyset\} \to \texttt{match}(v) = \emptyset$$

当 match$(u) = v \leftrightarrow (u,v) \in M$ 时，G_2，也适用相同的约束，区别在于所有对 match 函数的调用都替换为 match^{-1}（其中 match$^{-1}(u) = v \leftrightarrow (v,u) \in M$），以确保对称性。

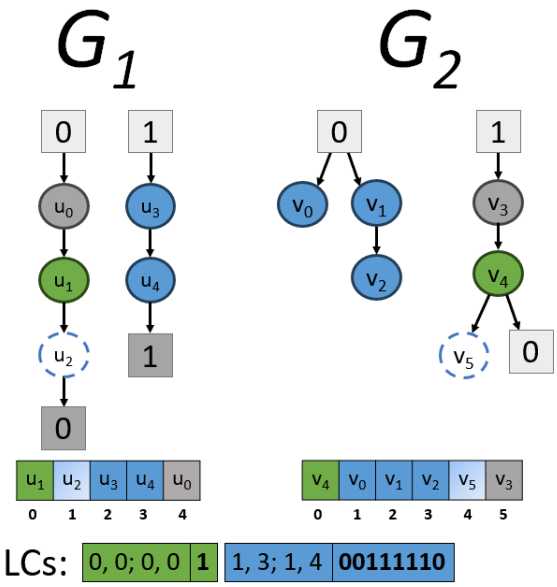


图 6：对一对位置图应用改进后的 McSplit 算法，此时已建立映射 $u_0 \to v_3$。在下一赋值步骤中，实体 u_2 和 v_5 将被禁止选择，因为它们分别与 u_0 和 v_3 传递相邻，但不直接相邻。此处允许选择蓝色标签类中的顶点；在默认的 Connected McSplit 中，这些顶点无法被选择，因为它们可以分别作为 u_1 和 v_4 的张量积存在。为简洁起见，本例未显示控制。

此方法还要求我们在开始搜索之前，先为 G_1 和 G_2 中各自所有顶点对构建两个后代映射，分别记为 τ_1 和 τ_2；当且仅当实体 u,v 传递相邻时，$\tau_{\{1,2\}}(u,v)$ 才为真。这样，求解器便可判断属于“无邻接”标签类的实体是否仍可被选中进行赋值。

在变量选择步骤中，可以调整 Connected McSplit 来适应这一点：将每个标签类中的布尔标志（用于表示其节点是否与候选解节点相邻）替换为位集变量，并将其定义为 A(l)。该变量表示其成员实体的实体选择可见性：如果标签类 l 与当前解二部图相邻，A(l) 就只包含一个值为 1（真）的位，因为可以推断所有成员顶点也都相邻。然而，如果标签类与解二部图不相交，那么该位集将以一个 0 开头，后面跟着 $(|\{u \in G_1 \wedge LC(u) = l\}| + |\{v \in G_2 \wedge LC(v) = l\})$ 个位，其中 $A(l)[k+1]$ 表示该类中的第 k 个实体当前是否是有效选择。在变量选择过程中，如果 $A(LC(v))[0] = 0$ 且 $A(LC(v))[v+1] = 0$，则禁止将顶点 v 赋值，因为它当前对传播器不可见。图 6 展示了给定状态下这一改进方法的示例实例，其中每个 LC 旁边都显示了其 A(l) 值。为简洁起见，此方法中的条件可见性约束不适用于实体的扁平化端口（第 5.2 节），因此只有在实体本身完成赋值后，才能对其端口顶点赋值。

我们还将 $A'(l)$ 定义为标签类 l 在上一个赋值步骤中的位集值。算法 1 展示了执行赋值（$u \in V_{G_1} \to v \in V_{G_2}$）时标签类位集的细化过程，该过程在划分过程之前进行。可以看出，与 McSplit 的主要标签类细化过程类似，我们的位集结构可以在 $O(n+m)$ 时间内完成细化

算法 1细化LC可见性 (int u, int v)

对所有 $l \in LC$ **执行**
  $\mathbf{S_1} \leftarrow \{w \in V_{G_1} \wedge lc(w) = l\}$
  $\mathbf{S_2} \leftarrow \{w \in V_{G_2} \wedge lc(w) = l\}$
  如果 $A'(l)[0] = 1$ **或** $(|S_1| > 0$ **且** $A_{V_1}[S_1[0]][u] > 0)$ **或** $(|S_2| > 0$ **且** $A_{V_2}[S_2[0]][v] > 0)$ **则**
    $A(l) \leftarrow 1$
    继续
  结束 if
  $A(l)[0] \leftarrow 0$
  如果 $\ell(l) \neq$ **link 则**
    对所有 $u' \in S_1$ **执行**
      如果 $A'(l)[u'+1] = 1$ **并且** $R[u][u'] = 1$ **则**
        $A(l)[u'+1] \leftarrow 0$
      如果结束
    循环结束
    对于所有 $v' \in S_1$ **执行**
      如果 $A'(l)[v'+1] = 1$ **并且** $R[v][v'] = 1$ **则**
        $A(l)[v'+1] \leftarrow 0$
      结束 if
    结束 for
  结束 if
结束 for

过程是通过检查每个尚未分配的实体与该步骤中已分配到每个大图中的实体之间的关系来实现的。划分过程本身也作了修改：在适当情况下，每个实体对应的标签类位值会随其一同重新定位，以保持状态之间的一致性。

除了这项额外的细化过程（用于确保组合完整性）外，我们的编码函数足以找出一对位置图的实体之间最大公共大图的所有有效赋值。图中被舍弃的区域和位置稍后会被重新添加，并在适当情况下进行闭合或合并，作为后处理相对推余函数的一部分，以确保最大性。

5.2 链接图编码

现在我们考虑将链接图加入编码中。我们通过构造适当的链接扁平化函数来实现这一点。

给定一个大图 B^{L}：$(V_B, E_B, ctrl_B, link_B)$：$X \to Y$，以及其位置图 D $\phi_{\{P,T\}}(D^{\mathsf{P}})$：$(V_D, E_D)$ 的编码，其中（其中 $V_B = V_D$），我们将扁平化函数定义如下：

$$\phi_f : \phi_{\{P,T\}}(D^{\mathsf{P}}) \times B^{\mathsf{L}} \mapsto (V, E)$$

所得扁平化图的顶点可以描述为：

$$V = V_D \uplus P_B \uplus \widehat{E_B}, \quad \widehat{E_B} = \{e \in E_B \mid link_B(p) = e, p \notin X\}$$

其中，$\widehat{\ } E_B$ 是B^{L}中的闭合链接集合，P_B 是B^{L}的端口，并添加一个额外的闭包节点来表示每个闭合链接。我们将大图的边标识符重新用作扁平化图中的顶点标识符。

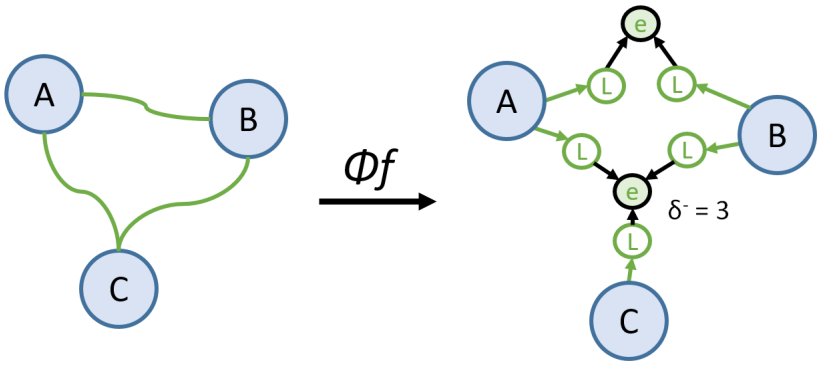


图7：MCB 算法中链接扁平化的示例——主搜索循环不考虑超边连接，而是在找到解后重新添加并合并。

我们将结果边集描述如下：

$$E = E_D \uplus \{(v,p) \mid p = (v,i) \in P_B\} \uplus \{(p,e) \mid e \in \widehat{E_B}, link_B(p) = e\}$$

这是用于大图匹配的基于团的链接扁平化函数的一个约束较少的变体 [2], ，其中我们不再在共享同一超边的端口之间构建团。这样一来，扁平化图就能表示输入链接图对的一种形式：所有端口都已释放，且每个端口都只有一条连接通向大图接口；这相当于我们在位置图中丢弃所有站点和区域的过程。由此得到的链接图中，接口组件之间不存在约束，表示在找到有效映射集时，可以成为G_1 和G_2 子结构的最小可能结构。与位置图类似，搜索结束后再细化并合并所有可能的链接集合（保留G_M 作为G_1 和G_2的子大图）。图7展示了修改后的扁平化函数的直观应用。

对闭合节点施加额外的度相等约束，以确保任何公共超边在G_1 和G_2中都具有同构邻接集。在 McSplit 中，可以利用现有的标签类结构预先实现这一约束：在搜索之前，按入度进一步划分表示所有 $\ell =$**闭合**节点的标签类。为保留解二部图中超边的结构，我们还对闭合节点施加约束：只有在其所有父端口节点都已分配后，闭合节点才可用于变量选择。这可以防止解中出现仍与接口相连的“闭合”链接边，而我们希望禁止这种情况。可以通过施加以下附加约束来体现这一点：

$$\{\forall e \in E_{G_1},\ p = (v,i) \in P_{G_1} \mid (p,e) \in link_{G_1} \wedge \texttt{match}(p) = \emptyset\} \rightarrow \texttt{match}(p) = \emptyset$$

这可以通过在分配端口时检查相邻的闭包，并在其所有父端口均已分配后将其切换为可见来实现；使用标签类中现有的位集结构，即可区分可见和不可见的闭包顶点。为此还需进行一项调整：闭包标签类的位集始终将首位设为 0，因为仅与当前选定子结构邻接已不足以保证该顶点在选择时可见。在 G_1，上施加这一约束，并结合度约束以确保同类匹配，也足以保证 G_2 中的闭包满足这一条件，因为共享同一标签类即可保证它们各自的邻域相匹配。

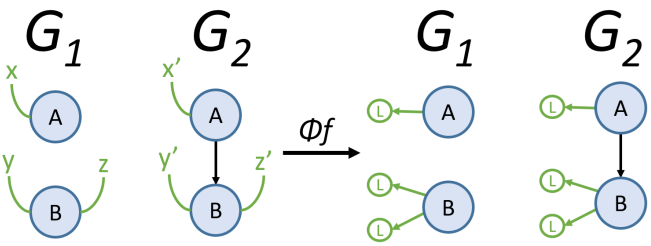


图 8：一个扁平化的 MCB 实例示例。由于计入了端口顶点，且未先修改评分函数，返回的 MCIS 解集与 MCB 的预期解集不匹配。

5.2.1 修改奖励函数

这种将端口表示为扁平化顶点的方法，使候选解的编码形式大小与其支撑大小之间出现了差异。图 8 展示了一个例子：其中给出了一个由一对双图构成的 MCB 实例，以及对 G_1 和 G_2 进行编码和扁平化后对应的 MCIS 实例。这个 MCB 实例有两个大小相同的解，大小均为 1，即 $(A \rightarrow A)$ 和 $(B \rightarrow B)$。然而，在编码后的 MCIS 实例中，只会返回 $(B \rightarrow B)$ 这一解，因为它的端口也会计入映射大小。这意味着，MCIS 算法会将 $(B \rightarrow B)$ 及其端口的映射视为大小为 3 的解，而将 $(A \rightarrow A)$ 及其唯一端口的映射所得的解大小则为 2。因此，我们希望修改 McSplit，使其忽略端口顶点的分配，尤其是在计算候选解当前得分时。不过，包含已分配实体但其端口尚未分配的情况绝不可能出现在有效解中，因为该实体的所有组成部分都必须与实体本身一同匹配（由于匹配标签的元数值始终相同，所有端口都应始终可供匹配）。考虑到这一点，我们引入一个新的评分函数，用于确定解的最优性，定义如下：

$$score = \begin{cases} -1 & \text{if } \exists \{\ p = (v,i) \in P_{G_1} \mid \texttt{match}(p) = \emptyset \ \wedge \texttt{match}(v) \neq \emptyset\} \\ |G_M| & \text{otherwise} \end{cases}$$

其中，P_{G_1} 是大图 G_1，的扁平化端口集合，而 $|G_M|$是公共双图的支撑大小。已分配实体和元数的对称性意味着，只需检查其中一个输入双图即可验证有效性。此外，我们还修改 McSplit 界函数，使其忽略所有包含端口节点的标签类，因为这些标签类不会对解的评分产生贡献：

$$bound = |G_M| + \sum_{l \in LC} min(|(u \in \{V_{G_1} \uplus E_{G_1}\} \wedge LC(u') = l \ \wedge \ u \neq (p,i) \in P_{G_1})|,$$

$$|(v \in \{V_{G_2} \uplus E_{G_2}\} \wedge LC(v') = l \ \wedge \ v \neq (q,j) \in P_{G_2})|)$$

其中 u' 和 v' 是表示支撑元素u 和 v 的编码顶点。由于顶点标签类型在初始化期间被划分为不同的标签类，因此只需检查任意成员元素的顶点标签值，即可轻松确定标签类 l 是否为“端口”类；如果顶点 $\ell(v)$ =**link** 的标签函数如此，则在求和时跳过 l 。虽然限制上界并非确保解的正确性所必需的步骤，但只要可能，我们都希望这样做，以确保

获得最佳性能并尽量减少执行冗余搜索树遍历所花费的时间，同时确保不会因此错误地过滤掉任何有效解。

6 实现

我们展示了一个用于改编版 MCB 算法的原型求解器。该求解器通过修改 Trimble 提供的 McSplit 变体实现，该变体使用 Python 编写，最初是作为对 *NetworkX* Python 图形库的贡献而创建的 [26]。此变体实现了 McSplit 的简化且未经优化的版本，不使用第 3 节所述的高效标签类结构，而是通过创建新的 Python 列表结构来模拟其行为，这些列表结构存储每个细化步骤中各顶点所属的标签类。尽管与使用更高效语言构建的 McSplit 优化实现相比，该实现的效率较低，但通过这一简化实现进行实验，可以更容易地设计和构建用于建模 MCB 的必要约束；随后，还可以使用手工构造的实例以及一组后来用于基准测试的大图匹配实例，对其正确性进行评估。

该实现还使用 McSplit↓ 版本的算法，其中求解器将实例视为一系列判定问题。从 $n = min(|G_1|, |G_2|)$ 开始，搜索循环尝试寻找大小为n的最大公共子图；当返回不可满足时，再将n 逐次减 1，直到找到至少一个解。界函数也经过修改：当界严格小于目标时触发回溯，而不是在界小于或等于目标时触发。研究发现，在所评估的多种基准测试实例中，McSplit↓ 的整体表现略优于默认的 McSplit [27]。

我们已公开提供代码 [6]。在实验过程中，我们发现，将变量选择启发式限制为：在选择一个元数为 n 的实体后，接下来的 n 步中只允许对端口进行赋值，以强制满足“实体不能有未赋值端口”这一约束，能够缩短求解时间。为此，我们使用一个全局的 端口_锁定变量进行跟踪：每当选择一个元数为 n的实体时，就将该变量设为 n；每次选择一个端口顶点时，则将其减 1。只要 端口_锁定 > 0，当前赋值集合的分数就设为零，并将其视为无效解。由于端口存在连通性约束，只有在实体顶点本身被选中后，该实体的端口顶点才会显示为可供选择，从而确保这一逻辑足以恰当地引导变量选择顺序。

6.1 评估

我们基于 Milner 的电话会议模型示例 [20]，从生成的大图匹配实例中选取 2000 个，对原型工具进行初步评估。由于 MCTS 的输入是反应规则红ex和代理状态，这与大图匹配类似；并且两者都会搜索模式的发生，以执行重写并扩展各自的迁移系统，因此这些测试实例适用于此场景。我们将模式规则（生成大小为 $|V| = 6$ 的编码图）与不同大小的目标代理进行匹配，其编码图大小从 59 到 1581 不等。我们记录调用底层 MCIS 搜索函数处理编码图所花费的时间，不计输入/输出时间；此外，还统计搜索后构建每个解映射所对应的 RPO 大图所花费的时间。为确保 MCB 实现的可靠性，我们进行验证：对于大图匹配中至少存在一个完整发生的每个实例，其所有发生也都存在于 MCB 的解集中——这是我们在第 4.1 节指出的一项必要属性。

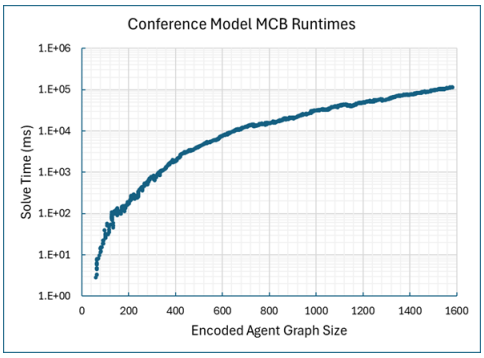


图9：在包含 2000 个实例的测试套件中，比较各会议模型红ex/代理对的求解时间与编码后的代理图大小（范围为 59 至 1581）的关系。

我们还通过验证输入图顺序颠倒时，整套实例始终返回相同的解集（映射取反），来确认 MCB 的对称性。我们公开提供完整的原始数据集 [4]。

图9显示了我们的原型工具所达到的求解时间，并将其与代理的编码图大小进行了比较。从这一比较中可以看出，我们的工具能够在一秒内求解图大小约为 300 以内的编码目标状态，这表明使用此实现来求解小型至中型代理的双模拟颇具前景。在 2000 个实例中，我们的原型工具能够在一秒内求解约 400 个实例，进一步证明了计算较小规模模型的最小上下文转移系统的可行性。虽然这还包括构建每个解的 RPO 大图所花费的额外时间，但在所有实例中，这部分时间仅占总求解时间的 $< 1\%$，因此在此情境下可忽略不计。这主要是因为反应规则红ex较小，不过在实际情境中，这些模式规则通常也应当较小（例如，BigraphER 套件提供的所有真实世界示例双图反应系统中，最大的平均模式匹配大小为 9）。不过，我们指出，作为未来优化求解器全面评估的一部分，记录所提供的模式双图逐渐增大（直至达到或甚至超过目标图大小）对性能的影响，或许值得关注。值得注意的是，对于会产生无限迁移系统的模型，无法验证双图之间的双模拟，因为这意味着每个代理都可能演化出无限多个状态——在这种情况下，模型至多只能判定一对代理“在 N 步以内双模拟”。因此，实际中预计代理不会增长到异常庞大的规模。

7 结论

本文定义了最大公共大图问题，并将其视为基于优化的大图匹配推广：其解遵循大图的 RPO 性质，以确保复合的最大性。我们还说明了如何利用该问题求出使反应规则得以适用的最小复合的逆，这正是最小上下文转移系统中必须解决的关键问题；此外，我们还确定了用于保证上下文与智能体之间接口兼容性的额外约束。随后，我们基于 McSplit 算法构建了一个用于求解 MCB 的原型回溯求解器。尽管目前这一实现还只是未经优化的原型，但在一组选定的

规则智能体输入对上的评估结果表明，从理论上讲，它适用于较小规模的模型。

现在我们已经介绍了该问题及其应用场景，接下来可以沿着这一方向开展进一步研究：构建并改进这一方法，使 McSplit 最大公共双图算法能够处理实际为代理/规则对寻找所有最小上下文时所施加的额外约束。例如，MCB 解不一定能够提供有效的 MCTS：接口上的不匹配可能会阻止缺失上下文被组合起来，而理想情况下，这种情况应在搜索传播过程中被检测到并排除。随后，还可以将一个更优化的底层求解器集成到 BigraphER 这样的双图工具包中。该求解器能够使用位并行结构并充分发挥 McSplit 的全部能力，从而可以通过为一组用于建模现实世界应用场景的双图反应系统构建 MCTS，开展进一步评估。

这种方法也有望通过并行化得到改进。尽管高效的子图同构求解器能够有效处理包含数千个顶点的子图同构问题实例，但最先进的最大公共子图算法在无标记图上通常只能在计算上处理约 35 个顶点，超过这一规模后，在一般情况下便变得不切实际 [15], 因此，我们可以预期，MCB（以及后续的 MCTS 构建）在计算上的开销将高于在原始转移系统中执行大图匹配。这促使我们思考：在可行的情况下，同时为一个 MCTS 求解多个 MCB 实例，并将它们分配到独立的进程线程中。如果将专用的 MCB 求解器集成到 BigraphER 之类的工具包中，就可以以与求解器无关的方式支持转移系统的并行扩展，而无需关注所构建的转移系统类型（原始转移系统或最小上下文转移系统），也无需关注底层依赖的求解器类型。

参考文献

[1] Faeq Alrimawi、Liliana Pasquale 和 Bashar Nuseibeh（2019）：智能空间中安全事件的自动化管理。*IEEE Access*7，第111513–111527页，doi:10.1109/ACCESS.2019.2934221。[2] Blair Archibald、Kyle Burns、Ciaran McCreesh 和 Michele Sevegnani（2021）：通过子图同构实现实用双图。载于：约束编程原理与实践——第*27*届国际会议，*CP 2021*，doi:10.4230/LIPIcs.CP.2021.15。[3] Horst Bunke（1997）：图编辑距离与最大公共子图之间的关系。*PatternRecognition Letters*18(8)，第689–694页，doi:10.1016/S0167-8655(97)00060-3。[4] Kyle Burns（2024）：测试实例与结果：双图匹配的高效可扩展算法（博士论文），doi:10.5281/zenodo.10578633。[5] Kyle Burns（2025）：双图匹配的高效可扩展算法。博士论文，格拉斯哥大学，doi:10.5525/gla.thesis.85146。[6] Kyle Burns 和 James Trimble：最大公共双图源代码。可访问：https://github.com/KyleBurns/MaximumCommonBigraph/tree/main。[7] Andrea Calabrese、Lorenzo Cardone、Salvatore Licata、Marco Porro 和 Stefano Quer（2023）：改进最大公共子图计算的网页抓取算法。载于：第*18*届国际软件技术会议论文集——*ICSOFT*，INSTICC，SciTePress，第197–206页，doi:10.5220/0012130800003538。[8] Muffy Calder 和 Michele Sevegnani（2014）：使用共享随机双图对 *IEEE 802.11* 载波侦听多路访问/冲突避免请求发送/允许发送进行建模。*Formal Asp. Comput.*26(3)，第537–561页，doi:10.1007/s00165-012-0270-3。[9] Alessio Chiapperini、Marino Miculan 和 Marco Peressotti（2022）：计算有向双图的（最优）嵌入。221，第102842页，doi:10.1016/j.scico.2022.102842。[10] Piotr Cybulski 和 Zbigniew Zieli´nski（2021）：使用跟踪双图反应系统对无人机群行为建模。*Sensors*21(2)，第622页，doi:10.3390/s21020622。

[11] Amal Gassara、Ismael Bouassida Rodriguez、Mohamed Jmaiel 和 Khalil Drira（2019）：执行大图反应系统。《离散应用数学》253，第73–92页，doi:10.1016/j.dam.2018.07.006。第14届科隆-特温特图与组合优化研讨会（CTW 2016）。[12] Dominik Grzelak（2024）：BiGGer：使用 Java 编写的、用于 GrGen.NET 中大图重写的模型转换工具。《开源软件期刊》9(98)，第6491页，doi:10.21105/joss.06491。[13] Dominik Grzelak（2025）：无人机集群编程：迈向安全的即插即用模块化。载于：第1届德国机器人会议（GRC）论文集，德国机器人研究所，德国纽伦堡。可访问：https://ras.papercept.net/conferences/conferences/GRC25/program/ GRC25_ContentListWeb_1.html#thga_58。独立视频投稿。[14] Dominik Grzelak 和 Markus Hamann（2025）：通过 GrGen.NET 改进大图重写以实现高效系统仿真。《Formal Asp. Comput.》，doi:10.1145/3736704。[15] Ruth Hoffmann、Ciaran McCreesh 和 Craig Reilly（2017）：介于子图同构与最大公共子图之间。《AAAI人工智能会议论文集》31(1)，doi:10.1609/aaai.v31i1.11137。[16] Mete Kemertas 和 Tristan Aumentado-Armstrong（2024）：迈向稳健的双模拟度量学习。载于：第35届神经信息处理系统国际会议论文集，NIPS '21，Curran Associates Inc.，美国纽约州雷德胡克，doi:10.48550/arXiv.2110.14096。[17] Jean Krivine、Robin Milner 和 Angelo Troina（2008）：随机大图。《电子理论与计算机科学札记》218，第73–96页，doi:10.1016/j.entcs.2008.10.006。[18] Ciaran Mccreesh、Samba Ndojh Ndiaye、Patrick Prosser 和 Christine Solnon（2016）：最大公共（连通）子图问题的团与约束模型。载于：第22届约束编程原理与实践国际会议（CP），《计算机科学讲义》9892，施普林格，法国图卢兹，第350–368页，doi:10.1007/978-3-319-44953-1_23。[19] Ciaran McCreesh、Patrick Prosser 和 James Trimble（2017）：最大公共子图问题的划分算法。载于：第二十六届国际人工智能联合会议论文集，IJCAI-17，第712–719页，doi:10.24963/ijcai.2017/99。[20] Robin Milner（2009）：《通信主体的空间与运动》。剑桥大学出版社。[21] Jannik Rau、David Richerby 和 Ansgar Scherp（2023）：为大图计算 k-双模拟：比较与效率分析"。载于：《图变换》，Springer Nature Switzerland，查姆，第223–242页，doi:10.1007/978-3-031-36709-0_12。[22] Michele Sevegnani（2012）：带共享的大图及其在无线网络中的应用。博士论文，格拉斯哥大学。可访问：https://theses.gla.ac.uk/3742/。[23] Michele Sevegnani 和 Muffy Calder（2015）：带共享的大图。《理论计算机科学》577，第43–73页，doi:10.1016/j.tcs.2015.02.011。[24] Michele Sevegnani 和 Muffy Calder（2016）：BigraphER：大图的重写与分析引擎。载于：《计算机辅助验证——第28届国际会议，CAV 2016》论文集，第二部分，第494–501页，doi:10.1007/978-3-319-41540-6 27。_[25]Michele Sevegnani、Milan Kabác、Muffy Calder 和 Julie A. McCann（2018）：大规模传感器网络基础设施的建模与验证。载于：第23届复杂计算机系统工程国际会议，ICECCS 2018，IEEE计算机学会，第71–81页，doi:10.1109/ICECCS2018.2018.00016。[26] James Trimble：NetworkX——"添加最大公共诱导子图"分支。可访问：https: // github.com/networkx/networkx/pull/4221。[27] James Trimble（2023）：诱导子图问题的划分算法。博士论文，格拉斯哥大学，doi:10.5525/gla.thesis.83350。[28] Yuepeng Wang、Isil Dillig、Shuvendu K. Lahiri 和 William R. Cook（2017）：验证数据库驱动应用程序的等价性。《ACM程序设计语言会议录》2(POPL)，doi:10.1145/3158144。

可靠性与完备性的证明

在此，我们证明所采用方法的可靠性与完备性。通过证明 MCIS 模型识别出的任何解都对应于一个符合最大公共大图定义的解实例，我们证明了可靠性。反过来，通过证明当 MCB 的某个实例存在解时，改编算法也会在 MCIS 编码中找到相应的匹配，我们证明了完备性。

关于如何为一对双图及其间有效的实体/闭包映射集合构造 RPO，双图领域的文献中已有论述 [20], ，因此我们不再重复证明这一点；相反，只需证明改编后的 McSplit 求解器返回的赋值集合足以传递给 RPO 构造函数，从而提取出最大公共结构。基于此，我们将修改后的 McSplit 返回的每个解 M 表示为由 (u,v) 对组成的单射集，表示从 G_1 中的每个顶点 u 到 G_2，中的顶点 v 的映射；忽略后来构造的公共双图中的顶点（由 RPO 函数处理）。因此，我们希望证明，对于任何大小为m的最大公共双图嵌入：

$$S_{MCB} = \{(u_1, v_1), ..., (u_m, v_m)\}$$

相应的编码将产生一个解：

$$M = \{(u'_1, v'_1), ..., (u'_m, v'_m)\}$$

(u_k,v_k) 与 (u'_k, v'_k) 之间存在双射关系，反之亦然。为了补充我们的证明，我们提出了一些与双图复合相关的观察。

命题 1. 给定两个位置图的复合

$$G : m \to n = (A : k \to n) \circ (B : m \to k)$$

若 $u \in V_B$, $v \in V_G$ 且 $u \in prnt^{+}_{G}(v)$ 则 $v \in V_B$。

这表明，V_B 中的任意实体的后代也必须属于V_B。我们通过递归应用双图复合的定义来证明这一点 [20], : 对于某个$v' \in V_G$，若$prnt_G(v') \in V_B$，则$v' \in o \uplus V_B$；并且我们知道v'是一个实体，因此$v' \in V_B$。将此应用于u，意味着$prnt^{-1}(u) \subseteq V_B$；再将此过程递归应用于u的所有子节点，即可证明u的所有孙节点都存在于V_B 中，依此类推。如此递归重复，便可遍历u的所有后代。因此，对于任意满足$prnt^{+}_{G}(v) = u \in V_B$的v，我们可证明$v \in V_B$，符合要求。

命题 2. 给定两个位置图的复合

$$G : m \to n = (A : k \to n) \circ (B : m \to k)$$

若 $u \in V_A$, $v \in V_G$ 且$v \in prnt^{+}_{G}(u)$ 则$v \in V_A$。

这表明，V_A 中某个实体的任何祖先也必须位于 V_A 中。我们再次通过递归应用双图复合的定义来证明这一点 [20], : 对于某个 $v' \in V_A$, $prnt_G(v') = prnt\{v')_A$, 因此 $prnt\{v')_G V_A \in$。将这一结论应用于 $prnt_G$ (u)，意味着 u 的祖父节点将位于 V_A 中；并且可以在每个更高层级递归应用，从而证明 u 的所有祖先都必须存在于 V_A 中。因此，对于任何满足 $prnt^{+}_{G}(u \in V_A) = v$ 的 v，我们证明 $v \in V_A$，如所需。

命题 3. 给定三个位置图的复合

$$G : m \to n = (A : k \to n) \circ (B : l \to k) \circ (C : m \to l)$$

若$u \in V_B$，$v \in V_B$，$w \in V_G$且$w \in prnt_G^+(u)$，$v \in prnt_G^+(w)$，则$w \in V_B$。

这表示，任何存在于 V_B 中一对实体之间的实体，也必须属于 V_B。我们通过如下方式结合应用命题 13 和 14 来证明这一点：根据 1，由于 u 是 w 的祖先，且 $u \in (B \circ C)$，因此我们知道 $w \in (B \circ C)$。根据命题 2，由于 v 是 w 的后代，且 $v \in (A \circ B)$，因此我们知道 $w \in (A \circ B)$。综合起来，我们可以得出结论：w 只能存在于 B 中，因此 $w \in V_B$。作为推论，我们可以非正式地推断，对于任意两个实体（$u \in V_G$，$v \in V_G$），若 $u \in V_B$ 且 $v \in V_B$ 且 $prnt_G^+(u) = v$，则它们之间的所有实体也必定存在于 B 中。

命题 4。 给定两个链接图的复合

$$G : X \to Y = (A : Z \to Y) \circ (B : X \to Z)$$

对于任意 $p \in P_B$ 和 $e \in E_B$，当且仅当 $link_G(p) = e$ 时，$link_B(p) = e$。

这说明，B 中的端口与一条边相连，当且仅当它们在 G 中也相连。首先，根据双图复合的定义， [20], $link_B(p) = e \in E_B \implies link_G(p) = link_B(p)$，因此 $link_G(p) = e$。其次，我们通过假设存在 $p \in P_B$、$e \in E_B$，使得 $link_G(p) = e$ 但 $link_B(p) \neq e$，来证明 $link_B(p) = e \impliedby link_G(p) = e$。反之，这意味着 $link_B(p) = x \in (Z \uplus E_B)/\{e\}$。如果情况确实如此，那么根据链接图构造，当 $x \in E_B$ 时，$link_G(p) = x \neq e$，这将导致矛盾；或者当 $x \in Z$ 时，$link_G(p) = link_A(x)$。然而，由于 $link_A(x)$ 存在于 A 中，因此它不可能是 $e \in E_B$，这与我们的假设相矛盾，并通过反证法证明了我们的命题。作为推论，我们可以得出 $|link_G^{-1}(e) \cap P_G| = |link_B^{-1}(e) \cap P_B|$。

A.1 可靠性

给定一个 MCB 实例 (G_1, G_2)，其 McSplit编码为 $(\phi_f(\phi_P(G_1)) \phi f (\phi_P(G_2)))$，且存在一个大小为 m 的解，其形式为单射映射 $M = \{(u'_1, v'_1) ..., (u'_m, v'_m)\}$。我们希望证明，这对应于一个 MCB 解 $G_1 = C_1 \circ (G_M \otimes id) \circ D_1, G_2 = C_2 \circ (G_M \otimes id) \circ D_2,$，其形式为单射嵌入映射 $S_{MCB} = \{(u_1, v_1) ..., (u_m, v_m)\}$，从支撑元素 $u \in G_1$ 的一个子集映射到 G_2，中的支撑元素 $v \in$，其中 $u'_k \to v'_k$ 是支撑元素 $u_k \to v_k$的编码形式。我们希望证明两个关键性质：首先，由 M 构造的大图 G_M 是 G_1 和 G_2 的公共大图。其次，不存在另一组大小为 $> m$ 的赋值 M'，使其同样是 G_1和 G_2的公共大图。

A.1.1 共性

我们通过从编码形式构造输入大图 G_K，$K = \{1, 2\}$，开始证明共性。对于位置图 $P = (V_K, ctrl_K, prnt_K)$，我们按如下方式进行：

$$\begin{aligned} &V_K = V_{\phi_P(G_K)} \\ &\{\forall v_i \in V_K \mid ctrl_K(v_i) = \ell(v'_i)\} \\ &\{\forall (v_i, v_j) \in V_K \mid (v'_i, v'_j) \in E_{\phi_P(G_K)} \to prnt(v_j) = v_i\} \end{aligned}$$

随后可以按如下方式构造链接图 $L = (V_K,\ E_K,\ ctrl_K,\ link_K)$:

$$
\begin{aligned}
&V_K = \{g' \in \phi_f(G_K) \mid \ell(g') \notin \{\textbf{link}, \textbf{closure}\}\}\\
&P_K = \{g' \in \phi_f(G_K) \mid \ell(g') = \textbf{link}\}\\
&E_K = \{g' \in \phi_f(G_K) \mid \ell(g') = \textbf{closure}\}\\
&\{\forall g \in V_K \mid ctrl_K(g) = \ell(g')\}\\
&\{\forall (g'_a, g'_b) \in E_{\phi_f(G_K)} \mid \ell(g'_a) = \textbf{link}, \ell(g'_b) = \textbf{closure}\} \rightarrow link_G(g_a \in P_G) = g_b \in E_G
\end{aligned}
$$

我们注意到，在初始编码过程中，G_1 和G_2 的接口相关结构信息会丢失。然而，如前所述，这些接口组件不会影响赋值集合M 能否生成有效最大公共大图，因为最初假定G_M 的接口处于完全开放状态，之后由 RPO 函数负责约束接口组件——因此，它们只会影响公共大图 RPO 的结构，所以在证明M的可靠性时不需要考虑这些组件。利用M 和任意一个重建的输入大图（本证明中使用G_1 ），便可通过忽略所有不属于解映射（因而不可能出现在G_M中）的元素，构造出具有约束最少的接口（通过尽可能分配站点和区域，使其上下两侧均完全开放）的公共双图 G_M。

我们对位置图$P = (V_M, ctrl_M, prnt_M)$执行此操作，具体如下:

$$
\begin{aligned}
&V_M = \{u \in V_{G_1} \mid M(u) \neq \emptyset\}\\
&\{\forall v \in V_M \mid ctrl_M(v) = ctrl_{G_1}(v)\}\\
&\{\forall (v_i, v_j) \in V_M \mid prnt_{G_1}(v_j) = v_i \rightarrow prnt_M(v_j) = v_i\}\\
&\{\forall v \in V_M \mid prnt_M(v) \notin V_M \rightarrow prnt_M(v) = r \in n\}\\
&\{\forall v \in V_M \mid (s \in m) \in prnt_M^{-1}(v)\}
\end{aligned}
$$

我们还为每个 $g \in P_M$ （其中 $link(g) \notin E_M$）添加一个新的外部名称作为汇。随后可按如下方式构造链接图 $L = (V_M, E_M, ctrl_M, link_M)$:

$$
\begin{aligned}
&V_M = \{u \in V_{G_1} \mid M(u') \neq \emptyset\}\\
&P_M = \{p \in P_{G_1} \mid M(p') \neq \emptyset\}\\
&E_K = \{e \in E_{G_1} \mid M(e') \neq \emptyset\}\\
&\{\forall v \in V_M \mid ctrl_M(v) = ctrl_{G_1}(v)\}\\
&\{\forall (g'_a, g'_b) \in E_{\phi_f(G_K)} \mid \ell(g'_a) = \textbf{link}, \ell(g'_b) = \textbf{closure} \rightarrow link_G(g_a \in P_G) = g_b \in E_G\}\\
&\{\forall p \in G_M \mid link_M(p) \notin E_M \rightarrow link_M(g) = y \in Y\}
\end{aligned}
$$

因此，我们能够构建公共双图 G_M。这样一来，我们便得到了所需的三元关系集合（$u \in G_1, v \in G_2, w \in G_M$），其中每个 w对应于公共元素（u,v） $\in M$的映射。

由此，我们实际上将问题归约为一对大图匹配实例：为了证明$G_1 = C_1 \circ (G_M \otimes id) \circ D_1$ 和$G_2 = C_2 \circ (G_M \otimes id) \circ G_2$，必须使$G_M$分别作为$G_1$ 和G_2 中的模式出现。对于每个输入大图，上下文和参数组件C_K和D_K，以及$K = \{1,2\}$

因此，可以按如下方式重构大图，首先从位置图开始：

$$\{\forall g_i \in V_M \mid prnt_M(g_i) = r \in n\} \rightarrow prnt_{G_K}(g_i) \in V_{C_K}, prnt_{C_K}(r) = g_i$$
$$\{\forall g_i \in V_M, g_j \in V_{G_K} \mid prnt_M^{-1}(g_i) \cap m = s, prnt_{G_K}(g_j) = g_i, g_j \notin M\} \rightarrow g_j \in V_{D_K}, prnt_{D_K}(g_j) = m$$
$$\{\forall g_i \in V_{G_K} \mid prnt_{G_K}(g_i) \in V_{D_K}\} \rightarrow g_i \in V_{D_K}, prnt_{D_K}(g_i) = prnt_{G_K}(g_i)$$
$$\{\forall g_i \in V_{G_K} \mid g_i \notin V_M, g_i \notin V_{D_K}\} \rightarrow g_i \in V_{C_K}$$
$$\{\forall g_i \in V_{C_K}\} \rightarrow prnt_{C_K}(g_i) = prnt_{G_K}(g_i)$$

由于此位置图构造已将 G_K 中的所有实体指派给C_K、G_M 或 D_K，因此我们假定在构建C_K 和 D_K 的链接时，已按如下方式完成此指派：

$$\{\forall p \in P_M \mid link_M(p) = y \in Y\} \rightarrow link_{C_K}(y) = link_M(p)$$
$$\{\forall p \in P_{D_K} \mid link_{G_K}(p) \notin D_K\} \rightarrow link_{D_J}(p) = x \in X, link_{id}(x) = x' \in Y, link_{C_K}(x') = link_M(p)$$
$$\{\forall p \in P_{C_K}\} \rightarrow link_{C_K}(p) = link_{G_K}(p)$$
$$\{\forall e \in E_{D_K}\} \rightarrow link_{D_K}^{-1}(e) = link_{G_K}^{-1}(e)$$

由此，便基于位置图中的实体和关系，以及链接图中的端口和边，构建出完整的 $G_K = C_K \circ ($ $G_M \otimes id) \circ D_K$ 分解。由于这现在为我们提供了初始的一对 MCB 组合，以及通过其 MCIS 编码和顶点指派集合 M 得到的公共双图 G_M ，因此构造性证明至此完成。

A.1.2 最大性

最大性也可以通过分析修改后的评分函数来进行构造性证明，具体如下。若当前赋值集合解码后得到不完整双图，则 M 的分数为 -1；反之，若该解在结构上健全，则 M 的分数为 $\sum_{(u',\ v')\in M} \{\ell(u') \neq \textbf{链接}\}$。
对于任意 MCB 实例，任何实例可能达到的最小得分始终为 0，其中$M = \{\}$ 和$G_M = /0$，因此不完整双图永远不会是解，我们可以假设最终结果始终满足$score(M) >= 0$ 。

由此可推断，MCIS 解M 的得分始终等于原始 MCB 实例中G_M 的支撑大小，因为端口不计入支撑大小；在我们的 MCIS 改造中，这一点体现为不将与端口对应的扁平化连接节点计入得分函数。因此，通过这种双射，我们可以得出结论：MCIS 返回的最大得分子图必定对应于G_1 与G_2之间的最大公共结构G_M 。至此，证明完毕。

A.2 完备性

给定一个 MCB 实例（G_1,G_2），其中 $G_1 = C_1 \circ (G_M \otimes id) \circ D_1$ 且 $G_2 = C_2 \circ (G_M \otimes id) \circ D_2$,，其中 G_1 和 G_2 为实双图，G_M为共享双图；并且存在一个大小为 m 的单射嵌入映射形式的解 $S_{MCB} = \{(u_1, v_1) ..., (u_m, v_m)\}$，它将支撑元素的一个子集中的元素 $u \in G_1$ 映射到支撑元素 $v \in G_2$, 。我们希望证明，在修改后的最大公共诱导子图实例（$\phi_f(\phi_P(G_1))$ $\phi_f\phi_P(G_2)$）中，存在一个大小相同的并行解 $M = \{(u'_1, v'_1) ..., (u'_n, v'_n)\}$，其中 $u'_k \rightarrow v'_k$是支撑元素 $u_k \rightarrow v_k$的编码形式。与可靠性证明类似，我们希望证明两个关键性质：首先，编码或约束不会错误地过滤掉任何有效最大公共双图解；其次，界函数不会错误地剪枝任何最优最大公共双图解。

A.2.1 有效性

假设存在一对有效的复合映射 $G_1 = C_1 \circ (G_M \otimes \mathrm{id}) \circ D_1$ 和 $G_2 = C_2 \circ (G_M \otimes \mathrm{id}) \circ D_2$，，且它们都具有 G_M 的对应嵌入，以及一个大小为 m 的对应嵌入映射 $S_{MCB} = \{(u_1, v_1) ..., (u_m, v_m)\}$；其中，对应的 MCIS 解 $M = \{(u'_1, v'_1) ..., (u'_m, v'_m)\}$ 并非有效解。这表明我们新添加的约束中至少有一项被违反。

我们首先考察对大图进行编码的方法。根据构造，G_K、$K = \{1,2\}$ 中实体之间的所有父关系，都通过将 $\mathrm{prnt}_{G_K}(v) = u$ 编码为 E_K 中的边 (u′,v′) 得以保留，因此传统最大公共诱导子图成立。显然，由于控制通过图标记得以保留，McSplit 会在搜索前将任何不兼容的实体分开，从而确保只有兼容的实体才能彼此映射。对于扁平化边节点，这种处理也足够充分，因为对于任意 (u′,v′)，$u \in E_K \iff v \in E_K$。根据命题 4，我们知道，对于有效的双图复合，闭合节点上的度数约束和实性约束绝不会被违反，因为 $\mathrm{link}_{G_M}(p \in P_{G_M}) = e \in E_{G_M}$ 当且仅当 $\mathrm{link}_{G_K}(p) = e$。

最后，我们现在考察实体上的连通性约束。根据命题 3，我们已经证明，对于 G 中任意一对传递相邻的实体 $(u, v) \in G_M$，位于 u 和 v 之间的所有实体也必须属于 G_M，且不能被分配到 C_1, C_2, D_1, D_2 中任一者。因此，对于任何有效的公共双图，该约束始终返回 **true**。至此，McSplit 最大公共诱导子图模型中的所有额外约束都已涵盖。

我们最初关于约束违反的假设被证明会导致矛盾，因此 S_{MCB} 必定是一个有效的 MCIS 解。证明至此完成。

A.2.2 界一致性

我们通过反证法证明界一致性，具体设定如下情形。给定一个 MCB 实例和解 (G_1, G_2, G_M)，我们的 MCIS 编码应返回一个对应的解 $M = \{(u'_1, v'_1) ..., (u'_m, v'_m)\}$，其中 $\sum \{\ell(u') \neq \mathrm{link}\} = |G_M|$。然而，我们假设对于某个最终会到达 M 的部分解 $K = \{(u'_1, v'_1) ..., (u'_k, v'_k)\}$, $k < m$，界检查未能通过，剩余搜索树因此被剪枝，错误地导致 M 无法被找到。我们将界检查设定为最严格的情形，因此假设此时当前最佳得分已经是 $|G_M|$，且 M 是一个与已知最大值大小相同的有效解。

要使该界限不成立，必须满足以下条件：

$$|G_M| > |G_K| + \sum_{l \in LC_K} min(|(u \in G_1 \wedge LC_K(u) = l \ \wedge \ u \neq (p,i) \in P_{G_1})|,$$

$$|(v \in G_2 \wedge LC_K(v) = l \ \wedge \ v \neq (q,j) \in P_{G_2})|)$$

其中，所有标签类 $l \in LC_K$ 都包含 G_1 和 G_2 中的互不相交的顶点子集（但这些子集不一定合起来构成完整的顶点集合）。

根据我们的编码可以推断，$|G_K| = \sum \{\ell(u) \neq \mathrm{link}\}$。现在我们定义 $R = \{(u',v') \in (u',v') \in K \{\{M \setminus K\} \mid \ell(u) \neq \mathrm{link}\}$，它由 $\{M \setminus K\} = \{(u'_{k+1}, v'_{k+1}) ..., (u'_m, v'_m)\}$ 中所有非端口元素对（因此也包括所有支撑元素）组成；这些赋值尚未加入 K，而加入后即可达到 M。显然，$|R| = |G_M| - |G_K|$。我们将其代回被违反的界函数，如下所示：

$$|\{(u',v') \in \{\{M \setminus K\} \mid \ell(u) \neq \textbf{link}\}| > \sum_{l \in LC_K} min(|(u \in G_1 \wedge LC_K(u) = l \ \wedge \ u \neq (p,i) \in P_{G_1})|,$$

$$|(v \in G_2 \wedge LC_K(v) = l \ \wedge \ v \neq (q,j) \in P_{G_2})|)$$

简而言之，这意味着，对于 K 而言，要达到M ，尚待赋值的非端口顶点数量必须大于所有非端口标签类中可供选择的顶点组合对数量。将其解码回原始的 MCB 实例，这表明，在复合映射对 $G_1 = C_1 \circ (G_K \otimes id) \circ D_1$ 和$G_2 = C_2 \circ (G_K \otimes id) \circ D_2$ 中，将所有剩余的兼容支撑元素加入 G_K，仍不足以使其达到 $|G_M|$ 的支撑大小，因此G_M 本身不可能是最优解。这与我们最初的假设相矛盾；只有当M 一开始就不是 MCIS 的最优解时，才会出现这种情况。因此，界函数只会剪枝非最优解，证明至此完成。