arXiv:2304.14295v1 [cs.CC] 27 Apr 2023

通过重配置进行解发现

Michael R. Fellows　Mario Grobler　卑尔根大学，挪威　不来梅大学，德国michael.fellows@uib.no　grobler@uni-bremen.de
Nicole Megow　Amer E. Mouawad　不来梅大学，德国　贝鲁特美国大学，黎巴嫩
nicole.megow@uni-bremen.de　aa368@aub.edu.lb
Vijayaragunathan Ramamoorthi　Frances A.Rosamond　不来梅大学，德国　卑尔根大学，挪威
vira@uni-bremen.de　Frances.Rosamond@uib.noDaniel Schmand　Sebastian Siebertz　不来梅大学，德国　不来梅大学，德国
schmand@uni-bremen.de　siebertz@uni-bremen.de

2023年4月28日

摘要

现实世界应用和系统的动态变化，要求我们能够通过以受限方式修改系统当前状态，高效地改进不可行解或恢复受损解。我们提出了一种新的通过重配置进行解发现 框架，用于从给定状态出发，执行一系列小规模修改，为给定问题构造可行解。该框架整合并形式化了经典局部搜索、再优化和组合重配置的不同方面。我们以多个基础组合问题为例说明该框架，包括顶点覆盖、独立集、支配集和着色。我们研究这些问题的解发现变体的经典复杂性和参数化复杂性，并探讨易处理实例与难处理实例之间的界限。

1 引言

在许多决策问题的动态现实世界应用中，必须从某个预先确定的系统状态出发寻找可行解。这不同于经典的学术问题，在后者中，我们可以从头开始计算一个可行解。然而，从头构建新解时，我们无法控制当前状态与新的目标状态之间的差异。在本文中，我们构建了一个新的框架，旨在通过有限数量的“微小”修改步骤，将某个可能不可行的解修改为可行解。

以频率分配问题为例 [1, 41] ，其中智能体通过无线消息传输进行通信。每个智能体都可以在一个固定频率上进行广播。要求任意两个相邻的智能体使用不同的频率，否则它们的信号会相互干扰，而这种情况必须避免。目标是分配给定数量的频率，使系统不发生干扰。这个问题可以方便地建模为图上的顶点着色问题。图中的顶点表示智能体；如果两个智能体彼此接近，且对应智能体使用的频率可能相互干扰，则在相应的两个顶点之间连接一条边。随后，频率对应于分配给顶点的颜色，因为相邻顶点必须获得不同的颜色。在图着色问题的解发现变体中，给定一个使用固定数量颜色的图的非合法着色，以及一个固定预算。目标是将给定的非合法着色转换为合法着色，同时确保重新着色步骤的数量不超过预算（重新着色步骤存在多种定义；我们将在后文讨论其中最常见的定义）。上述智能体通信示例对应于这样一种情形：智能体已经被分配了会造成一定干扰的频率。这种情况可能是由于运行中的系统发生了各种变化，例如引入新的智能体、位置发生变化，或广播范围扩大。重新从头计算合法着色可能并不可取，因为这可能会导致系统发生大量变化。根据给定的不可行指派，可能只需对着色进行少量受控修改，就能满足可行性约束。然而，一项修改可能会触发其他修改，因此问题在于，是否能够以“低成本”将系统重配置为有效状态。

作为我们框架的第二个动机，考虑这样一个例子：一组移动智能体负责监测一个国家（由一个图表示）的部分区域，以发现安全威胁，例如自然灾害。每个智能体负责监测该国家的某个子集（若干城市），而这些城市以一种反映网络不同部分之间依赖关系或邻接关系的方式相互连接。支配集中的智能体代表能够监测所有相关区域的智能体。通过在该图中识别一个较小的支配集，就可以确定并部署少量负责维护国家安全的智能体。在现实世界中，智能体可能会从系统中退役或被添加到系统中，感兴趣的监测区域（即城市）也可能随时间发生变化。鉴于移动智能体的成本很可能较高（可以将智能体想象成移动监测车辆），因此，在考虑转换步骤成本的同时，能够在修改后的系统中计算出一个新的支配集是理想的。

我们的新框架对静态（判定）问题的这类解发现版本进行建模。解发现问题与其他将一个解转换为另一个解的方法在风格上相似，例如局部搜索、再优化、动态算法以及组合重构。

局部搜索是一种算法范式，其基础是通过在定义明确的邻域中搜索更优解（通常由某些局部变化定义），对（可能不可行的）解进行迭代改进。局部搜索过程在邻域中不存在改进解时终止。与之密切相关的是再优化框架：在给定相邻问题实

例的解时，寻找问题实例的解。局部搜索和再优化已被证明在理论和实践中的许多问题上都非常强大；例如参见 [2, 36, 8]。与这些模型不同，在解发现问题中，只要我们能在有界数量的步骤内到达一个可行解，重配置步骤序列就不一定需要在每一步都找到改进解。

在动态（图）算法领域中，现实世界系统的动态性通过图中顶点/边的一系列添加/删除操作进行建模。目标是以远高于静态算法从头计算一个解的效率，输出更新后的解。动态图算法已得到广泛研究，也包括图着色 [5],最大独立集 [4], 和最小顶点覆盖 [62, 6]。动态算法的（参数化）复杂性也已得到研究，例如见 [43, 3, 53]。与解发现框架相比，系统变化以一种非常具体的方式发生，而重配置步骤仅受到计算时间（以及空间）的间接限制，并不受具体重配置预算的限制。

在组合重配置 框架 [68, 61], 中，我们同时给定（针对一个固定图的）初始解和目标解，并旨在通过执行一系列重配置步骤，将初始解转换为目标解。这些步骤选自定义明确的受限步骤集合，通常通过允许的令牌移动进行描述。然而，在解发现问题中，目标可行解是未知的（甚至可能不存在），目标是通过任意的中间配置（可能不可行）找到任意目标可行解，其数量受预算限制。

1.1 通过重配置进行解发现

我们聚焦于基础图问题：给定一个图 G 和一个整数 k，目标是为实例 (G, k) 找到一个可行解，例如，找到大小至多为 k的顶点覆盖、独立集或支配集，或者使用至多 k种颜色进行合法顶点着色。这些是人工智能中的重要组合问题，参见例如 [16, 44, 45,58, 64, 69], ，它们在现实世界的问题中有大量应用，包括特征选择 [71],、调度、规划、资源分配 [7], 、频率分配 [1], 、网络安全 [17],、传感器系统 [50],等。

我们介绍这类问题的解发现变体：给定一个图 G、一个大小为 k 的初始配置（不一定是可行解）以及一个预算 $b \in \mathbb{N}$。目标是使用至多 b个修改步骤，将初始配置转化为可行解。受重配置框架的启发，我们主要关注沿图的边进行更改的修改步骤（令牌滑动）。我们还简要讨论文献中研究过的其他修改步骤（令牌添加/移除或交换 和跳跃）。

如前所述，我们的框架受到这样一类方法的启发：从预定义系统状态出发寻找可行解，例如局部搜索、动态算法和再优化中的方法；但与组合重配置中的常见做法一样，它限制了局部修改步骤。上一节介绍了两个应用，其中解如何通过沿图中的边移动实体来修改，受到自然限制。更一般地说，我们对这类应用进行建模：物品/智能体只能沿给定物理网络中的链路移动，例如道路网络或计算机网络；或者运动轨迹受到限制。

我们将框架应用于顶点覆盖、独立集、支配集和着色问题，以此展示该框架。这些都是核心问题，已从局部搜索、再优化和组合重配置的角度得到广泛研究。为陈述我们的结果，我们假设读者对图论、上述问题以及（参数化）复杂性有所了解。以下各节将在符号首次使用前正式定义所有符号。

设 Π 是一个顶点子集问题，即定义在无向图上的问题，其解由一个顶点子集构成。Π-发现问题定义如下：给定一个图 G、一个大小为 k 的子集 $S \subseteq V(G)$（此时它不一定是 Π 的解），以及一个预算 b（正整数）。假设 S 中的每个顶点都包含一个令牌（对应一名智能体）。目标是判断能否通过至多 b 次移动 S 上的令牌，使最终得到的集合成为 Π 的解。

根据底层问题的不同，重配置领域中已经建立了令牌移动的不同定义。也就是说，对于令牌配置 Q，$Q' \subseteq V(G)$，我们有以下模型。在 令牌滑动模型 中，如果 $u \in Q$, $v \notin Q$, $v \in Q'$, $u \notin Q'$，且 $\{u, v\} \in E(G)$，则称一个令牌从 $u \in V(G)$滑动 到 $v \in V(G)$。在 令牌跳跃模型 中，如果 $u \in Q$, $v \notin Q$, $v \in Q'$，且 $u \notin Q'$，则称一个令牌从 $u \in V(G)$ 跳跃 到 $v \in V(G)$。最后，在 令牌添加/移除模型 中，如果 $v \notin Q$ 且 $v \in Q'$，则向顶点 $v \in V(G)$添加 一个令牌。类似地，如果 $v \in Q$ 且 $v \notin Q'$，则从顶点 $v \in V(G)$移除 一个令牌。

值得注意的是，我们重点研究顶点子集问题中的滑动，因为我们关于Π-发现的大多数结果都基于此。将 Π 分别实例化为顶点覆盖、独立集或支配集，即可得到顶点覆盖发现（VCD）、独立集发现（ISD）或支配集发现（DSD）问题。与顶点子集发现问题类似，我们在第6节定义着色发现问题。

1.2 我们的结果

我们研究上述解发现问题的经典复杂性和参数化复杂性，并证明以下结果。

1. **顶点覆盖发现：** 我们证明，即使在最大度数为四的平面图上，该问题仍为 NP 完全；即使在 2-退化二分图上，对于参数 b，该问题也是 W[1]-困难。另一方面，我们证明，对于树宽参数，该问题属于 XP（也就是说，在每个固定的有界树宽类上都可在多项式时间内求解）；对于一般图上的 k 参数，该问题属于 FPT；而当限制在结构上处处稀疏的图类时，对于 b 参数，该问题属于 FPT。

2. **独立集发现：** 我们证明，即使在最大度数为四的平面图上，该问题仍为 NP 完全；对于参数 $k + b$，即使在排除 $\{C_4$、……、$C_p\}$ 作为诱导子图的图上，该问题仍为 W[1]-难（对任意常数 p）；对于参数 b，即使在 2-退化二分图上，该问题仍为 W[1]-难。从积极的一面来看，我们证明了该问题关于树宽参数属于 XP；关于参数 k，在 d-退化图类和处处稀疏图类上属于 FPT；此外，关于参数 b，在结构上处处稀疏的图类上属于 FPT。

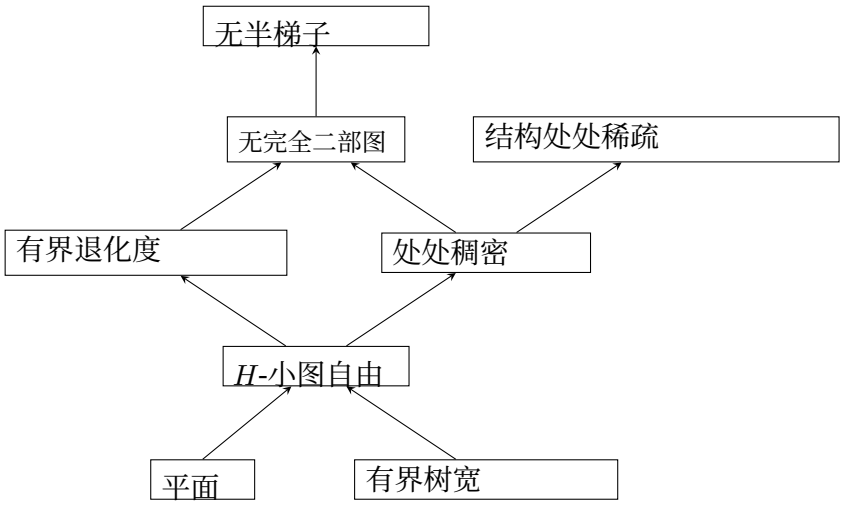

图 1：所考虑的图类。箭头表示包含关系。

3. **支配集发现：** 我们证明，即使在最大度数为五的平面图上，该问题仍为 NP 完全；对于参数 $k+b$，即使在二分图上，该问题仍为 W[1]-难；对于参数 b，即使在 2-退化图上，该问题仍为 W[1]-难。从积极的一面来看，我们证明了该问题关于树宽参数属于 XP；关于参数 k，在无完全二部图且无半梯子的图上属于 FPT；此外，关于参数 b，在结构上处处稀疏的图类上属于 FPT。

4. **着色发现：** 我们证明，对于每个 $k \geq 3$，即使在平面二分图上，该问题也是 NP 完全的（也就是说，对于参数 k 是 para-NP 困难的）；对于参数树宽，它是 W[1] 困难的；对于一般图上的参数 $k+b$，它是 W[1] 困难的。在积极的一面，我们证明，在结构上处处稀疏的图类上，该问题对于 $k=2$ 是多项式时间可解的，并且对于 $k+b$ 参数化后属于 FPT。

我们的结果几乎完整地刻画了小图封闭类以及单调类（即子图封闭类）图类上这些问题的复杂性。关于多项式时间可解的情形，注意到，一个图类具有有界树宽，当且仅当它排除某个平面图作为小图 [63]。因此，对于顶点子集发现问题，在平面图上的困难性意味着：对于小图封闭图类，只有具有有界树宽的图类才有望存在高效算法，而这正是我们所建立的结果。对于任意 $k \geq 3$，着色发现在平面图上以及在具有有界树宽的图类上都是 NP 完全的；因此，我们关于 $k=2$ 的可处理性结果是我们所能期望的最佳结果。

退化性和处处稠密性是非常一般的图稀疏性概念。除了这些类在理论上的意义之外，许多现实世界网络也被证明是退化的或处处稠密的，例如若干社会网络（如疾病传播网络）、生物网络（如基因相互作用或脑网络）以及信息学网络（如自治系统），参见例如 [15, 26, 33]。退化类和处处稠密类都是无完全二部图的。要找到既非退化、

也非处处稠密的无完全二部图类并不容易——一个有趣的例子是所有围长至少为 5、平均度为 $O(n^{1/5})$ 的 n-顶点图所构成的类，参见 [65]。结构上处处稀疏类是处处稠密类的稠密扩展，而半梯形自由类是无完全二部图类的稠密扩展。总体概览请参见图 1。

我们针对顶点子集发现问题及参数 b，对图的单调类给出了几乎完整的分类；我们证明了退化类上的困难性，以及（结构）处处稀疏类上的可处理性。对于参数 k，顶点覆盖发现问题在所有图上都是可处理的。类似地，对于参数 k，独立集发现问题在退化类和（结构）处处稀疏类上都具有可处理性，而其在无完全二部图类上是否具有可处理性仍是一个开放问题。支配集发现问题甚至在无半梯子类上也是可处理的，因此在无完全二部图类上也具有可处理性。

关于着色发现，仍有更多问题尚未解决。目前尚不清楚，对于参数 $k+b$，该问题在 d-退化图上是否具有可处理性；对于参数 b，在平面图上是否具有可处理性；以及对于参数 k，在树宽有界的类上是否具有可处理性。

2 预备知识

我们用 $\mathbb{N}$ 表示自然数的集合。对于 $k \in \mathbb{N}$，我们定义 $[k] = \{1, 2, \ldots, k\}$。

图 我们考虑有限、简单、无自环的无向图。对于图 G，我们分别用 $V(G)$ 和 $E(G)$ 表示 G 的顶点集和边集。若两个顶点 $u, v \in V(G)$ 满足 $\{u, v\} \in E(G)$，则称它们相邻或互为邻居。顶点 u 和 v 称为边 $\{u, v\}$ 的端点。顶点 v 的度等于其邻居 v 的数量。如果图中的所有顶点的度都不超过 d，则称该图的最大度数为 d。

由两两不同的顶点组成的序列 $v_1, \ldots, v_q$ 是长度为 $q-1$ 的路径，当且仅当对于所有 $1 \leq i < q$，均有 $\{v_i, v_{i+1}\} \in E(G)$。由两两不同的顶点组成的序列 $v_1, \ldots, v_q$ 是长度为 q 的环，当且仅当对于所有 $1 \leq i < q$，均有 $\{v_i, v_{i+1}\} \in E(G)$，且 $\{v_q, v_1\} \in E(G)$。我们用 P_q 表示长度为 q 的路径，用 C_q 表示长度为 q 的环。

参数化复杂性 参数化问题是一种语言 $L \subseteq \Sigma^* \times \mathbb{N}$，其中 Σ 是一个固定的有限字母表。对于实例 (x, κ)，$\in \Sigma^* \times \mathbb{N}$，$\kappa$ 被称为参数。如果存在一种算法，对于输入 (x, κ)，能在时间 $f(\kappa) \cdot |(x, \kappa)|^c$ 内判定 $(x, \kappa) \in L$，其中 f 是可计算函数，c 为常数，则问题 L 被称为固定参数可处理问题，简称 FPT。同样地，如果该问题可以由非确定性算法在相同的时间界内求解，则它属于 para-NP。L 是分片多项式的，简称 XP，如果存在一种算法，能在时间 $f(\kappa) \cdot |(x, \kappa)|^{g(\kappa)}$ 内判定 (x, κ) 是否属于 L，其中 f, g 为可计算函数。

“*W-层级*”是由参数化复杂度类 FPT $\subseteq$ W[1] $\subseteq \ldots \subseteq$ W[t] $\subseteq \ldots \subseteq$ para-NP $\cap$ XP 构成的集合。FPT $\neq$ para-NP 当且仅当 P $\neq$ NP，这是一个标准假设。此外，FPT $\subseteq$ W[1] 的包含关系被猜想为严格包含（并且在假设指数时间假设成立时，已知该猜想成立）。因此，在参数化情境下证明难解性，通常通过证明某个 W-困难问题可以固定参数可解归约到该问题来实现。

设 $L, L' \subseteq \Sigma^* \times \mathbb{N}$ 为参数化问题。从 L 到 L' 的参数化归约 是一种算法：给定 L 的一个实例 (x, κ)，输出 L' 的一个实例 (x', κ')，使得 $(x, \kappa) \in L \Leftrightarrow (x', \kappa') \in L'$，$\kappa' \leq g(\kappa)$，其中 g 是某个可计算函数；并且该算法的运行时间有界于 $f(\kappa) \cdot |(x, \kappa)|^c$，其中 f 是某个可计算函数， c 是常数。

有关参数化复杂性的详尽背景，请参阅教科书 [22, 29, 35] 。

3 顶点覆盖发现

在 NP 完全顶点覆盖（VC）问题 [37], 中，给定一个图 G 和一个整数 k，问题是判断 G 是否存在一个大小至多为 k 的顶点覆盖 C，即一个至多包含 k 个顶点的集合 C，使得对于每条边 $\{u、v\} \in E（G）$，都有 $\{u、v\} \cap C \neq \emptyset$。

3.1 相关工作

顶点覆盖问题存在一个 2-近似算法；假设唯一游戏猜想 [51]，对于任意 $\varepsilon > 0$，该问题难以在因子（$2 - \varepsilon$）以内近似。对于一个整数 λ，如果两个集合 A 和 B λ 的对称差大小至多为 λ，则称它们是 λ-接近的。通过与 λ-接近的解进行交换来执行局部搜索，当 $\lambda \in \mathsf{O}（1/poly（\varepsilon）$ 时，可以得到一个适用于具有次指数扩张 [42] 的图类的顶点覆盖（$1 + \varepsilon$）-近似算法。顶点覆盖问题是固定参数可处理问题 [29] 的教科书范例。该问题的动态变体也曾在 [3] 等文献中得到研究。

顶点覆盖问题在组合重配置框架下也得到了深入研究 [68, 61]。回顾一下，在（图的顶点子集）问题的重配置变体中，给定一个图 G 以及两个可行解 S（源）和 T （目标），目标是判断能否通过“较小的”重配置步骤将 S 变为 T，同时保持可行性（有时还会限制允许的重配置步骤数）。

与判定变体不同，已知在一般图上，顶点覆盖重配置（VCR）在令牌滑动、令牌跳跃和令牌添加/移除模型下均为 PSPACE-完全 [46, 48]。即使对于有界带宽/路径宽度/树宽等非常受限的图类，结论依然成立 [70]。另一方面，目前已知的多项式时间算法仅适用于树等非常简单的图类 [25]。如果考察问题的参数化复杂性，则可以得到更多积极结果（具体结果因模型而异）。更多详情请参阅 [14] 。

3.2 记号与定义

在顶点覆盖发现（VCD）问题中，给定一个图 G、一个起始配置 S 和一个正整数预算 b。目标是判断是否能够从 S 出发，使用至多 b 次令牌滑动，到达 G 的一个顶点覆盖（其中禁止多个令牌占据同一个顶点）。形式上，VCD 的一个实例是一个元组 $((G, S, b), \kappa)$，其中 κ 是参数。为便于阅读，我们将简写为 (G, S, b)，并在正文中明确说明参数。我们始终用 k 表示 S 的大小，即 $k = |S|$；但在第6节中， k 表示颜色数量。参数 κ 可以取 k、 b、 $k + b$，也可以取结构参数，例如输入图的树宽。

3.3 难解性

请注意，对于所有（图）顶点子集问题的解发现变体，我们总可以假设 $b \leq n^2$，其中 n 是输入图中的顶点数。这是因为每个令牌都必须经过一条长度至多为 n 的路径。因此，这类问题的所有解发现变体确实都属于 NP。我们首先证明，即使在限制非常严格的图类上，顶点覆盖发现问题仍然是 NP 完全的。

> 定理 3.1。 在最大度数为四的平面图类上，顶点覆盖发现问题是 NP 完全的。

证明。 我们将已知为 NP 完全的最大度数为三的平面图上的顶点覆盖问题归约而来 [55]。给定顶点覆盖问题的一个实例（G，κ），其中 G 是一个最大度数为三的平面图，我们如下构造一个 VCD 实例。我们创建一个新图 H，它由 G 的一个副本组成。然后，对于每个顶点 $v \in V$（H），我们创建一条由三个顶点组成的新路径 $\{x_v, y_v, z_v\}$，并将 v 连接到 x_v。我们选择 $S = \{x_v, y_v \mid v \in V(G)\}$，并设置预算 $b = \kappa$。注意，$k = |S| = 2|V(G)|$。由此完成 VCD 实例（H，S，b）的构造。根据该构造，H 是平面的，且最大度数为四。我们证明，（G，κ）是顶点覆盖问题的一个是实例，当且仅当（H，S，b）是 VCD 的一个是实例。

首先，假设 G 有一个大小至多为 κ 的顶点覆盖 C。那么，在 H 中，我们可以将 x_v 上的每个令牌直接滑动到 v，其中 $v \in C$。由于 C 的大小至多为 $\kappa = b$，因此我们至多需要进行 b 次滑动。为说明所得集合是 H 的一个顶点覆盖，注意到每条边 $\{v, x_v\}$ 仍由 v 或 x_v 覆盖，而边 $\{x_v, y_v\}$ 和 $\{y_v, z_v\}$ 仍由 y_v 覆盖。此外，H 的所有其他边也都被覆盖，因为 C 是 G 的一个顶点覆盖。

反过来，假设 (H, S, b) 是 VCD 的一个是实例。由于 $b = \kappa$，我们知道至多有 κ 个令牌可以移动，即放置在 $H[V(G)]$ 中。此外，由于所得配置必须是 H 的一个顶点覆盖，并且 $H[V(G)]$ 是 H 的一个诱导子图，因此，对应于 G 中各顶点的那些顶点上的令牌必须构成 G 的一个顶点覆盖，且大小至多为 κ，正如所需。 □

我们的下一个结果表明，从参数化视角来看，即使在 2-退化二分图上，仅以预算 b 为参数，顶点覆盖发现问题仍然是困难的。回顾一下，如果图 G 的每个子图 H 都包含一个度数至多为 d 的顶点，那么称图 G 为 d-退化图。

> 定理 3.2. 即使将问题限制在 2-退化二分图这一类图上，顶点覆盖发现问题以预算 b 为参数时仍是 W[1]-困难的。

证明。我们给出一个从团问题出发的参数化归约。设 (G, κ) 是团问题的一个实例。我们按如下方式构造图 H：从 G 出发，首先定义顶点集 $V(H)$。$V(H)$ 被划分为六个顶点集 $V_1, \ldots, V_6$。令 $V_1 = \{r_i \mid i \in [\binom{\kappa}{2}]\}$，且 $V_2 = \{z_i \mid i \in [\binom{\kappa}{2}]\}$。对于 $m = |E(G)|$，集合 V_3 由 $2m\binom{\kappa}{2}$ 个顶点组成，分为 m 个集合 V_3^e，每个 $e \in E(G)$ 对应一个集合。对于每个 $e \in E(G)$，令 $V_3^e = \{g_e^i, h_e^i \mid i \in [\binom{\kappa}{2}]\}$。令 $V_4 = \{y_e \mid e \in E(G)\}$ $V_5 = \{x_u \mid u \in V(G)\}$，且 $V_6 = \{s_u \mid u \in V(G)\}$。现在定义边集 $E(H)$。对于每个 $i \in [\binom{\kappa}{2}]$，我们在顶点 $r_i \in V_1$ 和

$z_i \in V_2$ 之间添加一条边。接下来，对于每个 $i \in [\binom{\kappa}{2}]$ 和 $e \in E(G)$，顶点 $g_e^i \in V_3^e$ 与顶点 $z_i \in V_2$、$h_e^i \in V_3^e$ 和 $y_e \in V_4$ 相邻。对于每条边 $e = \{u, v\} \in E(G)$，顶点 $y_e \in V_4$ 与 x_u, $x_v \in V_5$ 相邻。最后，对于每个 $u \in V(G)$，我们在顶点 $x_u \in V_5$ 和 $s_u \in V_6$ 之间添加一条边。至此，图 H 的构造完成。我们定义大小为 $k = m + n + m\binom{\kappa}{2}$ 的初始配置 $S = V_6 \cup V_4 \cup \{g_e^i \in V_3 \mid e \in E(G)\ i \in [\binom{\kappa}{2}]\}$，并令$b = 2\binom{\kappa}{2} + \kappa$（见图2）。

我们首先证明图 H 确实是 2-退化的。回顾一下，如果图 H 的每个诱导子图 H 都包含一个度至多为 d 的顶点，则称图 H 是 d-退化的。考虑 H 的任意诱导子图 H'。如果 H' 包含来自 V_1、V_3 或 V_6 的顶点，那么结论显然成立，因为 V_1 中的顶点和 V_6 中的顶点在 H 中的度为一，而 V_3 中的顶点在 H 中的度为三，但其中一个邻居是度为一的顶点。不难看出，不包含 $V_1 \cup V_3 \cup V_6$ 中顶点的 H 的任何诱导子图，要么包含来自 V_2 的孤立顶点，要么包含来自 V_4 的度为二的顶点，要么包含来自 V_5 的孤立顶点（当不包含 V_4 中的任何顶点时）。

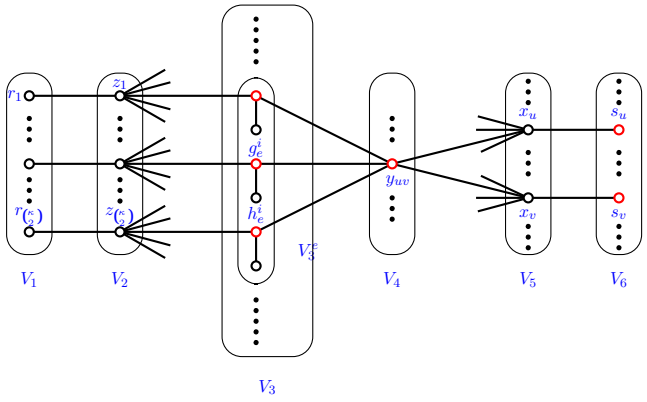

图2：W[1]-困难性归约的示意图。红色顶点表示初始配置。

现在我们从正向开始证明该归约的正确性。设 C 是 G 中大小为 κ 的团。令 $S' = (S \cup V_2 \cup \{x_u \mid u \in C\}) \setminus (\{y_{uv} \mid u, v \in C\} \cup \{s_u \mid u \in C\})$。我们用至多 b 次令牌滑动，从 S 得到 $S' \subseteq V(H)$。我们将任意 $\binom{\kappa}{2}$ 个顶点上的令牌从 V_3 滑动到 V_2。然后，我们将 $\{y_{uv} \mid u, v \in C\}$ 上的令牌滑动到 V_3 中交换后的 $\binom{\kappa}{2}$ 个顶点上。接下来，我们将 κ 个令牌从 $\{s_u \mid u \in C\}$ 滑动到 $\{x_u \mid u \in C\}$。由于 $V_2 \cup \{g_e^i \in V_3 \mid e \in E(G) i \in [\binom{\kappa}{2}]\} \subseteq S'$，关联于这些顶点的边均已覆盖。对于每个 $u \in V(G)$，$x_u \in V_5$ 或 $s_u \in V_6$ 必在 S' 中。因此，集合 V_5 与 V_6 之间的边由 S' 覆盖。最后，V_4 中不属于 S' 的顶点与 V_5 之间的边，由对应于团 C 的顶点覆盖，因为 V_4 中移动的顶点对应于团 C 的边。

对于反向，我们考察 H 中顶点覆盖发现问题的可行解的性质。集合 S 不包含来自 $V_1 \cup V_2$ 的任何顶点。由于 V_1 与 V_2 之间的边是匹配边，我们需要将 $\binom{\kappa}{2}$ 个令牌从 S 移动到 V_2。这些令牌必须来自 V_3。对于每个 $i \in [\binom{\kappa}{2}]$ 和每个 $e \in E(G)$，g_e^i 或 h_e^i 必须属于某个可行解。因此，我们必须将 $\binom{\kappa}{2}$ 个令牌从 V_4 移动到 V_3。此时我们只剩下 κ 的预算来覆盖这些 $\binom{\kappa}{2}$ 个顶点之间的边。只有当 $\binom{\kappa}{2}$ 条边构成一个团时，这才可能实现。

□

3.4 可处理性

现在我们将展示，在考虑不同参数时，顶点覆盖发现问题变得固定参数可处理。不同于定理 3.2，下一个结果表明，将令牌数量 k 而非预算 b 作为参数，可以使该问题变得可处理。

> 定理 3.3。当以令牌数量 k作为参数时，顶点覆盖发现问题是固定参数可处理的。此外，该问题可以在时间 $O(2^k \cdot n^3)$ 内求解。

证明。我们的算法包含一个枚举步骤，用于枚举 G 的所有大小至多为 k的极小顶点覆盖。然后，对于每个大小至多为 k的顶点覆盖 C，我们在二分图上求解一个最小权完美匹配（MWPM）实例。

为了枚举所有大小至多为 k的顶点覆盖，我们采用标准搜索树算法，该算法逐步构造极小顶点覆盖，并在其叶节点中生成所有大小至多为 k 的极小顶点覆盖。考虑一个 VC 实例。为了构建搜索树，在每个非叶节点处，算法选择一条尚未被覆盖的边，并通过两种可能的方式覆盖该边进行分支：在每个分支中分别将该边的一个端点加入覆盖。由于我们不关注基数超过 k的顶点覆盖，因此无需搜索树中深度超过 k 的部分，从而得到叶节点数量上界 2^k，以及枚举时间上界 $O^*(2^k)$。

接下来需要检查的是，是否存在大小至多为 k 的顶点覆盖 C，能够通过至多 b 次令牌滑动从 S 达到。为此，我们执行如下步骤。对于每个集合 C，我们构造一个完全加权二分图 H，其中 H 的二部划分由 (S, C) 构成。也就是说，我们将 S 的顶点置于一侧，将 C 的顶点置于另一侧。对于每一对顶点 $u \in S$ 和 $v \in C$，我们将边 $\{u, v\} \in E(H)$ 的权重设为 u 到 v 在 G 中的一条最短路径上的边数（当 u 和 v 属于不同的连通分量时，我们将权重设为 $m+b+1$）。现在，我们在 H 中应用一个求解最小权完美匹配的算法。如果完美匹配的权重不超过 b，那么我们就可以相应地移动令牌。否则，我们继续处理下一个顶点覆盖。要理解其正确性，请注意：如果令牌 t_1 为到达其目标所要经过的路径上还有另一个令牌 t_2，我们可以交换这两个令牌的目标，然后继续移动 t_2。可以验证，移动次数不会超过完美匹配的权重。另一方面，发现问题的任何解都必须将令牌移动到目标位置，其效果不可能优于最小权完美匹配中最短路径长度之和。这一观察最早见于 [18]。众所周知，使用 Edmonds 花算法 [31] 等方法，可以在 $O(n^3)$ 时间内求解最小权完美匹配问题。

□

接下来，我们证明，当以输入图的树宽为参数时，VCD 属于 XP。特别地，对于每个固定的有界树宽类，该问题都是多项式时间可解的。注意，在下面的定理中，k 可以为 O(n)，因此 $2^{\mathcal{O}(t \log k)} \in n^{\mathcal{O}(t)}$。我们指出，对于参数树宽，该问题甚至可能属于 FPT，但我们既无法证明这一结论，也无法证明其 W[1]-困难性。

> **定理 3.4。** 顶点覆盖发现问题可以在 $2^{\mathcal{O}(t \log k)} \cdot n^{\mathcal{O}(1)}$ 时间内求解，其中 t 表示输入图的树宽。

证明。令（G，S，b）为一个 VCD 问题实例。我们首先计算 G 的一个宽度为 O(t) 的优美树分解（T，（X_i）$_{i \in V(T)}$）。这可以在 $2^{\mathcal{O}(t)} \cdot n^2$ 时间内完成；关于树宽和优美树分解的未定义概念，我们参阅教科书 [22]。我们将 G_i 记为由以 i 为根的子树中各袋内的所有顶点诱导的子图。对于每个节点 $i \in V(T)$，我们计算一个表 $\mathtt{Sol}_i$。表中的条目由四元组索引。我们将这些四元组称为状态。因此，节点 i 处的一个状态 s 是一个元组（q，ℓ，$\mathtt{A}$，$\mathtt{B}$），其中

- $0 \leq q \leq k$ 是一个整数，用于指定子问题 G_i 所需顶点覆盖解的大小（令牌数量），
- $0 \leq \ell \leq b$ 是一个整数，用于指定子问题 G_i 中允许的滑动步数，
- $\mathtt{A}$：$X_i \to \{0, 1\}$ 是 X_i 中顶点的指示函数（标记 X_i 中必须属于某个解的顶点），并且
- $\mathtt{B}$：$X_i \to [-k, k]$ 是一个函数，用于指定 X_i 中每个顶点上滑动的令牌数量，其中负值表示有更多令牌从 G_i 滑向图中的其余部分，而正值表示相反的情况。设 $\gamma^+ = \{u \in X_i \mid \mathtt{B}(u) > 0\}$ 和 $\gamma^- = \{u \in X_i \mid \mathtt{B}(u) < 0\}$。对于每个顶点 $u \in \gamma^+$，我们可以使用 $\mathtt{B}(u)$ 个在 u 处可用的额外令牌。假定这些令牌在顶点 u 处可用，并且它们从 G_i 外部的顶点滑入。进一步地，对于每个顶点 $u \in \gamma^-$，我们必须将 $\mathtt{B}(u)$ 个令牌滑到顶点 u，这些令牌的作用是之后可以滑到 G_i 外部的顶点。

我们用 $\mathtt{S}_i$ 表示节点 i 的所有状态的集合。对于状态 $s = (q, \ell, \mathtt{A}, \mathtt{B}) \in \mathtt{S}_i$，当且仅当以下条件成立时，$\mathtt{Sol}_i[s]$ 设为 1：存在一个大小为 q 的集合 $C \subseteq V(G_i)$，使得

- C 是 G_i 的顶点覆盖，其中 $C \cap X_i = \mathtt{A}^{-1}(1)$，并且

使用至多 ℓ 个滑动步数、从每个顶点 $u \in \gamma^+$ 使用 $\mathtt{B}(u)$ 个令牌，并将 $\mathtt{B}(u)$ 个令牌滑动到每个顶点 $u \in \gamma^-$，可以从 $(S \cap V(G_i)) \cup \gamma^+$ 到达 $C \cup \gamma^-$。

直观地说，函数 $\mathtt{B}$ 确定了两个量：

- 可从 G_i 外部到达分隔集的额外令牌数量；以及
- 需要到达分隔集的所需令牌数量，以便它们最终滑动到 G_i之外的顶点。

我们计算所有可能的函数 $\mathtt{B}$、函数 $\mathtt{A}$、解的大小 $0 \le q \le k$ 以及滑动步数 $0 \le \ell \le b$ 的解。

局部无效状态。 如果存在一个顶点 $u \in X_i$，使得 $N[u] \subseteq V(G_i)$ 和 $\mathtt{B}(u) \neq 0$，则称状态 s 为局部无效。对于任意满足 $N[u] \subseteq V(G_i)$ 的顶点 u， $\mathtt{B}(u)$ 必须为 0，因为令牌无法从顶点 u 滑动到 G_i之外的顶点。因此，对于局部无效状态 s，我们设 $\mathtt{Sol}_i[s] = 0$。此外，这种状态的存在对于父节点中的任何状态都不再有用。如果一个状态不是局部无效状态，则它是局部有效状态。因此，在动态规划计算中，我们只考虑局部有效状态。

对于每个节点 $i \in V(T)$ 和状态 $s \in \mathtt{S}_i$，我们展示如何根据 T 中节点 i 的类型计算 $\mathtt{Sol}_i[s]$。

叶节点。 设 i 为袋 $X_i = \emptyset$的叶节点。状态集 $\mathtt{S}_i$ 是一个单元素集合，其中唯一的状态为 $s = (0, 0, \emptyset \to \{0, 1\}\ \emptyset \to [-k, k])$，且 $\mathtt{Sol}_i[s] = 1$。

引入节点。设 i 是一个引入节点，其子节点为 j，且 $X_i = X_j \cup \{v\}$，其中 $v \notin X_j$为某个值。注意， $N[v] \cap V(G_i) \subseteq X_i$。设 $N_v = N(v) \cap X_i$。若 $\mathtt{A}(v) = 0$且$N(v) \cap X_i \not\subseteq \mathtt{A}^{-1}(1)$，则我们令 $\mathtt{Sol}_i[s] = 0$。也就是说，关联于 v的边要么由 v 覆盖，要么由 v的邻居覆盖。因此，要么 v 被包含在解中，要么v 在 X_i 中的所有邻居都被包含在解中。处理节点 j的表项时，顶点 v 将被视为外部顶点。我们考虑令牌经过v滑动的所有可能方式。设 $F = \{f : N_v \to [-k, k] \mid \sum_{u \in N_v} f(u) = \mathtt{B}(v)\}$。函数 $f \in F$定义了每个邻居 $u \in N_v$给予（或从中取得）的令牌数量。对于每个 $f \in F$，我们定义 $\mathtt{B}_f : X_j \to [-k, k]$，使得对于每个 $u \in X_j$，都有

$$\mathtt{B}_f(u) = \begin{cases} \mathtt{B}(u) + f(u) & \text{if } u \in N_v, \\ \mathtt{B}(u) & \text{otherwise.} \end{cases}$$

对于每个 $f \in F$，令 $den(f) = \sum_{u \in N_v} |f(u)|$且 $s_f = (q - \mathtt{A}(v)\ \ell - den(f)\ \mathtt{A}_j,\ \mathtt{B}_f)$。值 $den(f)$ 表示将令牌滑动经过或滑动至 v 所需的滑动步数，而$\mathtt{A}_j$ 表示限制在集合 X_j上的函数 $\mathtt{A}$。最后，状态 s 的表值可计算为

$$\mathtt{Sol}_i[s] = \bigvee_{f \in F} \mathtt{Sol}_j[s_f].$$

所需时 间受 $O(|F| \cdot t) = 2^{\mathcal{O}(t \log k)}$ 的上界。

遗忘节点。 设 i 是一个遗忘节点，其子节点为 j，且满足 $X_i = X_j \setminus \{v\}$，其中 $v \in X_j$。注意到 $N[v] \subseteq V(G_i) = V(G_j)$。因此，对于任意局部有效状态 $s' = (q', \ell', \mathtt{A}', \mathtt{B}') \in \mathtt{S}_j, \mathtt{B}'(v) = 0$。定义 $\mathtt{B}_j : X_j \to [-k, k]$，使得对于每个 $u \in X_j$，都有

$$\mathtt{B}_j(u) = \begin{cases} 0 & \text{if } u = v, \\ \mathtt{B}(u) & \text{otherwise.} \end{cases}$$

接下来，我们尝试两种可能性：将顶点 v 纳入目标解，或将其排除。对于 $c \in \{0, 1\}$，定义 $\mathtt{A}_c\colon X_j \to \{0, 1\}$，使得对于每个 $u \in X_j$，都有

$$\mathtt{A}_c(u) = \begin{cases} c & \text{if } u = v, \\ \mathtt{A}(u) & \text{otherwise.} \end{cases}$$

我们将状态 s 的值 $\mathtt{Sol}_i[s]$ 计算为

$$\mathtt{Sol}_i[s] = \bigvee_{c \in \{0,1\}} \mathtt{Sol}_j[(q, \ell, \mathtt{A}_c, \mathtt{B}_j)].$$

值 $\mathtt{Sol}_i[s]$ 可以在常数（O(1)）时间内计算得出。

连接节点。设 i 为一个连接节点，其子节点为 j 和 h，满足 $X_i = X_j = X_h$。令 $F = \{f : X_i \to [-k, k]\}$。目标解的大小为 q，其中 $\mathtt{A}^{-1}$ 个顶点包含在 X_i 中。因此，我们尝试从 $V(G_i) \setminus X_i$ 中获取 $q - \mathtt{A}^{-1}$ 个额外顶点的所有可能方式。对于每个 $0 \le q' \le q - \mathtt{A}^{-1}$，令 $q_j = q' + \mathtt{A}^{-1}$ 和 $q_h = q - q'$。

类似地，我们总共有 ℓ 个滑动步数的预算。因此，我们尝试将 ℓ 分配到分别以 j 和 h 为根的两个子树中的所有可能方式，其中将 ℓ' 分配给以 j 为根的子树，将 $\ell - \ell'$ 分配给以 h 为根的子树。接下来，我们定义 $\mathtt{B}$ 关于 j 和 h 的函数计算。对于每个 $f \in F$，我们定义 $\mathtt{B}_j^f\colon X_j \to [-k, k]$ 和 $\mathtt{B}_h^f\colon X_h \to [-k, k]$，如下所示。对于每个 $u \in X_j$、$\mathtt{B}_j^f(u) = \mathtt{B}(u) + f(u)$，以及每个 $u \in X_h$、$\mathtt{B}_h^f(u) = \mathtt{B}(u) - f(u)$。函数 $f \in F$ 定义了在两个子树（分别以 j 和 h 为根）之间移动的令牌数量。正号表示令牌从以 j 为根的子树移动到以 h 为根的子树，负号则相反。我们将状态 s 的值 $\mathtt{Sol}_i[s]$ 计算为

$$\mathtt{Sol}_i[s] = \bigvee_{\substack{0 \le q' \le q - \mathtt{A}^{-1}(1), \\ 0 \le \ell' \le \ell, \\ f \in F}} (\mathtt{Sol}_j[q_j, \ell', \mathtt{A}, \mathtt{B}_j^f] \wedge \mathtt{Sol}_h[q_h, \ell - \ell', \mathtt{A}, \mathtt{B}_h^f]).$$

计算 $\mathtt{Sol}_i[s]$ 的值所需的时间为 $O(|F| \cdot k \cdot b) = 2^{\mathcal{O}(t \log k)}$。

给定一个有 n 个顶点的图 G，G 的树分解有 $O(n)$ 个节点。对于树分解中的每个节点 i，表 $\mathtt{Sol}_i$ 存储了 $2^{\mathcal{O}(t \log k)}$ 个状态的值。计算一个状态的表项需要 $2^{\mathcal{O}(t \log k)} \cdot n^{\mathcal{O}(1)}$ 的时间。因此，计算树分解中所有节点的表需要 $2^{\mathcal{O}(t \log k)} \cdot n^{\mathcal{O}(1)}$ 的时间。令 r 为根节点。我们从根节点处的表项中得到最终答案。也就是说，(G, S, b) 是 VCD 的一个是实例，当且仅当对于某个 $\ell \le b$，$\mathtt{Sol}_r[(k, \ell, \emptyset \to \{0, 1\}\ \emptyset \to [-k, k])] = 1$。 □

现在我们考虑参数 b 在（结构上）处处稀疏的图类上的情形。处处稀疏图类这一概念，是对图稀疏性的若干既有概念的共同推广，例如平面图、排除（拓扑）小图的图、具有（局部）有界树宽的图，或最大度有界的图。已知，对于每个处处稀疏图类，都存在一个函数 f，使得每个可由一阶公式 φ 定义的性质都能在时间 $f(|\varphi|, \varepsilon) \cdot n^{1+\varepsilon}$ 内判定，

对于任意 $\varepsilon > 0$ [39]。最近，这一可处理性结果被推广到结构上处处稀疏类 [30], ，这类图构成了处处稀疏类的稠密版本。我们利用这一元定理，证明了存在一种固定参数可处理算法，用于求解顶点覆盖发现问题；该问题以 b 为参数，并限制在结构上处处稀疏的图类中。

> 定理 3.5。 当以 b 为参数，并限制在结构处处稀疏的图类中时，顶点覆盖发现问题是固定参数可处理的。

证明。我们将 VCD 归约为一阶模型检验问题；已知该问题在结构处处稀疏图类上是固定参数可处理的 [30]。只需证明：我们可以构造一个一阶公式 φ ，用于表达 VCD 实例存在解，并且对于某个函数 f有 $|\varphi| = f(b)$。给定一个 VCD 实例 (G, S, b)，我们用一个一元谓词标记集合 S ，也就是说，这是一个一元关系，当且仅当 $v \in S$时， $S(v)$成立。这样，输入配置便可供一阶公式使用。构造 φ 的过程如下。注意，我们需要表达：在 S 中存在至多 $q \leq \min\{k, b\}$ 个互不相同的顶点，它们能够通过总长度至多为 b 的路径到达 q 个彼此不同且不属于 S 的其他顶点，并且所得集合是 G的顶点覆盖。为表达这一点，我们对至多 $q \leq \min\{k, b\}$ 个属于 S的互不相同顶点 $u_1, \ldots, u_q$进行存在量化，并对 q个属于 $V(G) \setminus S$的互不相同顶点 $v_1, \ldots, v_q$进行存在量化。现在，对于满足 $\sum_{1 \leq i \leq q} \ell_i \leq b$ 的所有 $\ell_1, \ldots, \ell_q$，我们验证（再次通过存在量化）从 u_i 到v_i 存在一条长度为 ℓ_i的路径。我们对 $\ell_1, \ldots, \ell_q$的所有此类选择作一个大析取。最后，还需检查集合 $(S \setminus \{u_i \mid i \in [q]\}) \cup \{v_i \mid i \in [q]\}$ 是该图的顶点覆盖。为此，可以检查图中的每条边是否至少有一个端点属于 $S \setminus \{u_1, \ldots, u_q\}$，或属于 $\{v_1, \ldots, v_q\}$。公式的形式化表达略去，因为它不会带来更多见解。

□

4 独立集发现

在独立集（IS）问题中，给定一个图 G 和一个整数 k，问题是判断 G 是否包含大小至少为 k 的独立集，其中独立集是由两两不相邻的顶点组成的集合。

4.1 相关工作

独立集问题是 NP 完全的 [49], ，即使要在任意 $\varepsilon > 0$ [72]下以 $n^{1-\varepsilon}$的因子进行近似，它仍然是 NP 完全的。以解的大小 k 为参数的独立集问题是 W[1]-完全的，因此被认为不是固定参数可处理的 [28]。

因此，对于一般图，不能指望局部搜索方法改进上述给出的近似因子。然而，在实践中，我们经常处理属于特殊图类的图，例如平面图，以及更一般的具有次指数扩张的图类；已知在这些图类中，局部搜索能够带来更好的近似算法，甚至是多项式时间近似方案（PTAS），例如参见 [42]。

独立集问题也是组合重配置框架下研究最为广泛的问题之一 [68, 61]。就经典复杂性而言，独立集重配置问题和顶点覆盖重配置问题的结果是可以相互转换的；我们只需对顶点集取补集即可（回顾一下，每条边至少有一个端点属于顶点覆盖，因此剩余的顶点构成一个独立集）。如果考虑问题的参数化复杂性，则可以得到更多积极的结果（这与顶点覆盖重配置问题的结果大不相同） [60]。

4.2 记号与定义

在独立集发现（ISD）问题中，给定一个图 G、一个由 k 个令牌组成的初始配置 S 以及一个正整数预算 b。目标是判断是否可以从 S 出发，经过至多 b 次令牌滑动，达到一个大小为 G 的独立集（我们不允许两个或更多令牌占据同一个顶点）。同样地，我们用 $(G,\ S,\ b)$ 表示 ISD 的一个实例，并在正文中明确所考虑的参数。

注意，对于独立集发现问题，令牌跳跃模型和令牌添加/移除模型可归结为以下问题。如果 $k \leq b$ （对于令牌跳跃）或 $k \leq 2b$（对于令牌添加/移除），那么问题就是判断是否存在一个大小为 k的解，因为在这种情况下，我们只需将令牌逐个移动到该解中的位置。换句话说，该问题归结为经典的独立集问题。如果 $b \leq k$ （对于令牌跳跃）或$2b \leq k$ （对于令牌添加/移除），那么问题就是判断是否存在一个与初始配置的对称差至多为 $2b$的解。这个问题已在独立集的局部搜索版本中得到研究。因此，我们专注于令牌滑动模型。

4.3 难解性

我们首先证明，即使将问题限制在最大度数为四的平面图上，独立集发现问题仍然是 NP 完全的。

> 定理 4.1。 在最大度数为四的平面图类上，独立集发现问题是 NP 完全的。

证明。我们从最大度数为三的平面图上的 IS 进行归约；已知该问题是 NP 完全的 [59]。给定 IS 的一个实例（G,κ），其中 G 是一个最大度数为三的平面图，我们如下构造 ISD 的一个实例。我们创建一个新图 H，其初始状态为 G 的一个副本。然后，对于每个顶点 $v \in V$（H），我们在五个顶点w_v、 x_v、 c_v、 y_v、 z_v上创建一条新路径，并将 v与 c_v相连。我们选择 $S = \{$ c_v、 x_v、 $y_v \mid v \in V$（G）$\}$，并设置预算 $b = 2n - \kappa$，其中 $n = |V$（G）$|$。注意， $k = 3|V$（G）$|$。至此，实例（H、 S、 b）的构造完成。容易观察到，图 H是平面的，且最大度数为四。我们证明， （G、 κ）是 IS 的是实例，当且仅当（H、 S、 b）是 ISD 的是实例。

首先假设 G 有一个大小至少为 κ的独立集 I。然后，在 H 中，我们可以将 c_v 上的每个令牌滑动到 v，其中 $v \in I$。对于所有其他顶点 $v \notin I$，我们将 x_v 上的每个令牌滑动到w_v，并将 y_v 上的每个令牌滑动到 z_v。注意，对于每个顶点 $v \notin I$上的路径，我们需要预算 2 进行修复，而对于顶点 $v \in I$上的路径，我们只需要预算 1。由于 I的大小至少为 κ，因此我们至多需要 $2n - \kappa = b$ 次滑动。为了证明所得集合是

一个大小为 H的独立集，请注意，对于顶点 $v \in I$ 上的每条路径，我们都将令牌从 c_v 移动到了其自身 v。由于 I 是一个独立集，且 x_v 和 y_v 唯一发生冲突的邻居分别是c_v，因此这些路径上的令牌构成一个独立集。顶点 $v \notin I$的路径上的令牌同样构成一个独立集。由于 w_v 的唯一邻居是 x_v，且令牌已从 x_v 移动到 w_v，因此不存在冲突。对于 y_v 和 z_v也是如此。由于 c_v 的邻居 x_v 和y_v 已被腾空，且 v 自身没有令牌，即这些路径构成一个独立集。

对于逆向过程，假设 (H, S, b) 是 ISD 的是实例。令 I 为所得的独立集。我们需要证明 $|I \cap V(G)| \geq \kappa$，这随后对应于 G 中的一个独立集。反设 $|I \cap V(G)| = \ell < \kappa$。这意味着 $3n - \ell$ 个令牌仍位于路径构件上。由于每个路径构件至多包含 3 个独立顶点，且 I 是一个独立集，因此至少有 $n - \ell$ 个路径构件包含 3 个令牌。要在不将这 3 个令牌移出路径的情况下保持它们彼此独立，至少需要进行 2 次滑动。因此，这些滑动至少需要 $2n - 2\ell$ 的预算。此外，$V(G)$ 上的每个 ℓ 个令牌都至少需要进行一次滑动。总计需要 $2n - \ell > 2n - \kappa$ 的预算，矛盾。

□

接下来我们将说明，从参数化视角来看，当考虑参数 $k + b$ 时，该问题同样是困难的。

> 定理 4.2。 当以 $k + b$ 为参数时，即使在排除 $\{C_4, \ldots\ldots, C_p\}$ 作为诱导子图的图上，对于任意常数 p，独立集发现问题也是 W[1]-困难的。

证明。 我们从多彩独立集（MIS）问题进行参数化归约。已知对于任意常数 p [10]，该问题在排除 $\{C_4, \ldots\ldots, C_p\}$ 作为诱导子图的图上是 W[1]-困难的。回顾一下，在 MIS 问题中，给定一个图 G 和一个整数 κ，其中 $V(G)$ 被划分为 κ 个团 V_1、V_2、……、V_κ；目标是寻找一个大小为 κ 的多彩独立集，即对于 $i \in [\kappa]$，从每个集合 V_i 中恰好包含一个顶点的独立集。给定 MIS 的一个实例（G，κ），我们按如下方式构造 ISD 的一个实例（H，S，b）。首先，令 H 为 G 的一个副本。然后，对于每个 $i \in [\kappa]$，我们在两个新顶点 $\{u_i, w_i\}$ 上添加一条边，并令 u_i 与 V_i 中的所有顶点相邻。最后，我们选择$S = \{u_i \mid i \in [\kappa]\} \cup \{w_i \mid i \in [\kappa]\}$，并令 $b = \kappa$。注意到 $k = |S| = 2\kappa$。

假设 G 有一个大小为 κ 的多彩独立集。令 $I = \{v_1、\ldots\ldots、v_\kappa\}$ 表示这样一个集合，其中 $v_i \in V_i$。然后，我们可以通过按需将 u_i 上的每个令牌滑动到顶点 v_i，来解决该发现实例。反过来，由于我们需要滑动顶点 u_i 上的所有令牌，并且每个集合 V_i 只能包含一个令牌，因此只有当 G 存在大小为 κ 的多彩独立集时，这才有可能。

□

接下来，我们将进一步研究独立集发现问题在仅以 b 或 k （而非 $k+b$）为参数，且限制在特殊图类时的复杂度。结果表明，当以 b 为参数且限制在 2-退化二分图类时，该问题仍然是 W[1]-困难的。

> 定理 4.3。 即使将预算 b 作为参数，独立集发现问题在 2-退化二分图类上仍是 W[1]-难的。

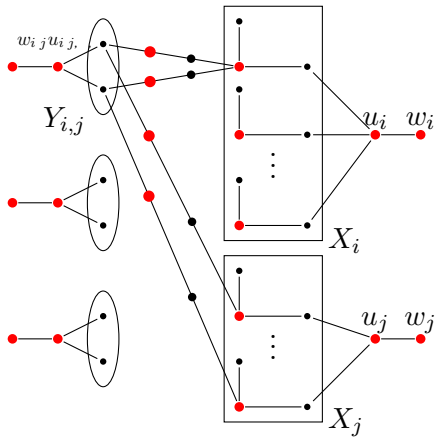

图3：2-退化二分图上独立集发现的 W[1]-困难性归约示意图。

证明。我们给出一个从多色团问题出发的参数化归约；这是一个众所周知的 W[1]-难问题 [22]。回顾一下，在多色团问题中，给定一个图 G 和一个整数 κ，其中 $V(G)$ 被划分为 κ 个独立集$V_1, V_2, \ldots, V_\kappa$，目标是寻找一个大小为 κ的多色团，即每个集合 V_i中恰好包含一个顶点的团，其中 $i \in [\kappa]$。给定一个多色团实例 (G, κ)，我们按如下方式构造一个 ISD 实例 (H, S, b)。首先描述图 H。对于每个顶点集 V_i，我们创建一个新集合 X_i，其中每个顶点 $v \in V_i$ 都替换为一条含 3 个顶点的路径，记为 x_v、 y_v和 z_v。此外，我们添加一个顶点 u_i ，并将其与所有顶点 z_v相连，其中 $v \in V_i$。最后，我们添加一个顶点 w_i ，它仅与顶点 u_i相邻。对于 $i < j \in [\kappa]$，我们用$E_{i,j}$ 表示图 G中连接 V_i 内顶点与 V_j内顶点的边集。对于每个 $E_{i,j}$，我们创建一个新的顶点集，记为 $Y_{i,j}$，其中每条边 $\{u, v\} \in E_{i,j}$对应一个顶点 e_{uv} 。此外，我们添加一条边（由两个新顶点构成）$\{w_{i,j}, u_{i,j}\}$，其中 $u_{i,j}$ 也与 $Y_{i,j}$ 中的每个顶点相邻（但不属于 $Y_{i,j}$）。对于每个顶点 $e_{uv} \in Y_{i,j}$, $i < j \in [\kappa]$，我们通过一条由两个新顶点构成的路径将 e_{uv} 连接到 y_u ，并通过一条由两个新顶点构成的路径将 e_{uv} 连接到 y_v 。令$X = \bigcup_{i \in [\kappa]} X_i$ 以及 $Y = \bigcup_{i, j \in [\kappa]} Y_{i,j}$。我们用 W表示从 Y 到 X 的路径上所有与 Y 中的某个顶点距离为 1 的顶点所构成的集合，并用 Z 表示这些路径上所有与 Y中的某个顶点距离为 2 的顶点所构成的集合。最后，令$S = W \cup \{u_i \mid i \in [\kappa]\} \cup \{w_i \mid i \in [\kappa]\} \cup \{u_{i,j} \mid i < j \in [\kappa]\} \cup \{w_{i,j} \mid i < j \in [\kappa]\} \cup \{y_v \mid v \in V(G)\}$，并令 $b = 3\binom{\kappa}{2} + 2\kappa$ 和 $k = 2\kappa + 2\binom{\kappa}{2} + n + 2m$ （见图3）。

根据构造，不难看出图 H 确实是二分图。要证明图 H 是 2-退化的，只需注意，删除所有 y_v 顶点、 u_i 顶点和$u_{i,j}$顶点后，我们得到一个最大度数为 2 的图，且被删除的顶点之间互不相邻。因此， H的每个子图要么含有一个形如 w_i、 $w_{i,j}$、 x_v、 z_v的顶点，或者含有一个属于 $W \cup Y$的顶点，这些顶点的度都至多为 2；要么不存在这样的顶点，这意味着每个其他顶点的度都至多为 2。我们断言，$(G$、 $\kappa)$ 是多色团的一个是实例，当且仅当 $(H$、 S、 $b)$ 是 ISD 的一个是实例。

首先假设 $(G、k)$ 是一个是实例，并令 $C = \{v_1、v_2、\ldots v_\kappa\}$ 表示 G 中的一个多色团，其中每个顶点 v_i 对于 $i \in [\kappa]$ 都属于 V_i。我们构造一个滑动序列，将 S 转换为一个独立集，具体如下。对于每个 i，我们将位于 y_{v_i} 上的令牌滑动到 x_{v_i}，然后将位于 u_i 上的令牌滑动到 z_{v_i}。这总共需要 2κ 次滑动。接下来，对于每一对 $i、j$，其中 $i < j$，我们将位于 $u_{i、j}$ 上的令牌滑动到顶点 $e_{v_iv_j}$，然后将 W 中的两个令牌滑动到它们在 Z 中的邻居。这总共需要 $3\binom{\kappa}{2}$ 次滑动，从而得到总计 $b = 3\binom{\kappa}{2} + 2\kappa$ 次滑动。由于 C 是 G 中的一个多色团，因此得到的配置确实是 H 的一个独立集。

对于反向方向，假设 (H, S, b) 是 ISD 的是实例。由于 u_i 和 w_i 上有两个相邻令牌，对于每个 $i \in [\kappa]$，我们知道每个 i 至少需要一次移动。此外，由于每个顶点 y_v 都包含一个令牌，我们知道每个 i 还需要一次额外滑动。因此，对于形如 $\{u_i, w_i\}$ 的边，我们至少需要 2κ 次滑动。类似地，对于每一对 $i、j$（其中 $i < j$），$w_{i、j}$ 以及 $u_{i、j}$ 上分别有两个相邻令牌。此外，W 中的所有顶点都包含令牌。因此，对于每一对 $i、j$（其中 $i < j$），我们至少需要三次滑动；对于形如 $\{u_{i,j}, w_{i,j}\}$ 的边，总共需要 $3\binom{\kappa}{2}$ 次滑动。因此，必须存在 κ 个顶点 y_{v_i} 和 $\binom{\kappa}{2}$ 个顶点 $e_{v_iv_j}$ 与这些顶点相邻，才能成功地将 u_i 和 $u_{i、j}$ 个顶点上的令牌滑离其邻居 w_i 和 $w_{i、j}$（这些邻居包含令牌）。

□

4.4 可处理性

我们关于以 k 为参数的独立集发现所取得的正面结果，利用了 [56] 中引入的独立集覆盖族这一概念。我们对这类族作出形式化定义，并介绍了针对不同图类提取它们的各种算法。

定义 4.1. 对于图 G 和 $k \geq 1$，若由 G 的独立集组成的一个族满足以下条件，则称其为 (G, k) 的独立集覆盖族，记作 $\mathrm{F}(G, k)$：对于 G 中任意大小至多为 k 的独立集 I，都存在 $J \in \mathrm{F}(G, k)$，使得 $I \subseteq J$。

定理 4.4 ([56]). 给定一个 d-退化图类或处处稀疏图类 $\mathscr{C}$、一个图 $G \in \mathscr{C}$ 以及 $k \geq 1$，存在一种确定性算法，其运行时间为 $f(k) \cdot n^{\mathcal{O}(1)}$，并输出一个大小至多为 $g(k) \cdot n^{\mathcal{O}(1)}$ 的 (G, k) 独立集覆盖族。其中，$f(k)$ 和 $g(k)$ 取决于 k 和图类 $\mathscr{C}$，但与图的大小无关。

现在我们可以证明以下定理：

定理 4.5。当参数化为 k 时，对于每个能够容纳大小为 $g(k)$ 的独立集覆盖族的图类 $\mathscr{C}$，独立集发现问题都是固定参数可处理的，其中该独立集覆盖族可在 $f(k) \cdot n^{\mathcal{O}(1)}$ 时间内计算，而 f 和 g 是可计算函数。特别地，该问题在 d-退化类和处处稀疏图类上是固定参数可处理的。

证明。给定一个 ISD 实例 (G, S, b)，其中 $G \in \mathscr{C}$，我们首先计算一个大小为 $g(k) \cdot n^{\mathcal{O}(1)}$、用时 $f(k) \cdot n^{\mathcal{O}(1)}$ 的独立集覆盖族 $\mathrm{F}(G, k)$，这由假设可知是可行的（对于 d-退化的处处稀疏图类，也可由定理 4.4 得出）。令 $\mathrm{F}(G, k) = \{J_1, J_2, \ldots\}$ 表示所得的族。令 $J \in \mathrm{F}(G, k)$。我们构造一个完全加权

二分图 $H_{S,J}$，如下所示。令 S 为一侧的顶点，令 J 为另一侧的顶点。我们将每条边 $\{u,v\}$ 的权重设为 u 到 v 在 G 中的一条最短路径上的边数（当 u 和 v 属于不同的连通分量时，我们将权重设为 $m+b+1$）。

接下来要证明，(G,S,b) 是一个是实例，当且仅当存在一个 $J \in \mathsf{F}(G,k)$，使得 $|J| \geq k$，并且 $H_{S,J}$ 中的最小权完美匹配的权重至多为 b。假设前述结论成立，只需遍历每个大小至少为 k 的 J，构造图 $H_{S,J}$，然后计算 $H_{S,J}$ 中的最小权完美匹配，即可得到该算法。如果找到权重至多为 b 的匹配，则这是一个是实例；否则，这是一个否实例。

假设 (G,S,b) 是一个是实例。那么，存在一个大小为 k 的独立集 I，它可以通过至多 b 次令牌滑动从 S 到达。根据独立集覆盖族的定义，存在 $J \in \mathsf{F}(G,k)$，使得 $I \subseteq J$。此外，由于 I 可以通过至多 b 次滑动从 S 到达，$H_{S,I}$ 中完美匹配的权重必定至多为 b。因此，$H_{S,J}$ 中最小权完美匹配的权重至多为 b，证毕。

现在假设存在一个大小至少为 k 的 $J \in \mathsf{F}(G,k)$，使得 $H_{S,J}$ 中的最小权完美匹配的权重至多为 b。回顾独立集覆盖族的定义可知，J 是 G 中的一个独立集。因此，J 的任意大小为 k 的子集 I 都是 G 中大小为 k 的独立集。令 I 表示在最小权完美匹配中与 S 中某个顶点匹配的顶点集合。正如我们刚才所述，I 是 G 中大小为 k 的独立集。因此，还需证明我们可以使用至多 b 次滑动从 S 到达 I。如果令牌 t_1 为到达其目的地所经过的路径上有另一个令牌 t_2，我们可以让这两个令牌交换目的地，然后继续移动 t_2。可以验证，移动次数不超过完美匹配的权重。

众所周知，可以使用 Blossom 算法 [31] 或匈牙利算法 [54] 在 $O(n^3)$ 时间内求解最小权完美匹配。□

采用与有界树宽图上的顶点覆盖发现问题（定理 3.4）相同的动态规划技术，并归约到结构上处处稀疏类上的一阶模型检测（定理 3.5），我们可以得到以下结果：

> 定理 4.6。独立集发现问题可在时间 $2^{\mathcal{O}(t \log k)} \cdot n^{\mathcal{O}(1)}$ 内求解，其中 t 表示输入图的树宽。

> 定理 4.7。当参数为预算 b 且限定在结构上处处稀疏的图类中时，独立集发现问题是固定参数可处理的。

5 支配集发现

在 NP 完全的支配集（DS）问题中 [37],，给定一个图 G 和一个整数 k，问题是判断 G 是否存在一个大小至多为 k 的支配集 D，即一个至多包含 k 个顶点的集合 D，使得 $V(G)$ 中的每个顶点要么属于 D，要么在 D 中有一个邻居。

5.1 相关工作

通过简单的贪心算法 [47, 57]，支配集问题可以近似到 $\log n$ 倍，而且除非 P $=$ NP [27]，否则这一因子在渐近意义上无法改进。通过与 λ-近解进行交换的局部搜索，对于 $\lambda \in$ O$(1/poly(\varepsilon))$，可在具有次指数扩张 [42] 的类上得到 $(1+\varepsilon)$-近似。该问题是 W[2]-完全的 [28]。该问题的动态变体已在例如 [3] 中得到研究。

对于支配集重配置问题，已知在令牌跳跃模型下，该问题在分裂图、二分图、有界带宽图以及最大度数为六的平面图上是 PSPACE-完全的 [40]。另一方面，该问题在树、区间图和余图上可在线性时间内求解。Bonamy 等人 [9] 表明，在令牌滑动模型下，支配集重配置问题在分裂图、二分图和有界树宽图上是 PSPACE-完全的，在对偶弦图和余图上则可在多项式时间内求解。Bousquet 和 Joffard [13] 表明，该问题在圆弧图上可在多项式时间内求解，在圆图上则是 PSPACE-完全的。

5.2 记号与定义

在支配集发现（DSD）问题中，给定一个图 G、一个由 k 个令牌组成的初始配置 S 以及一个正整数预算 b。目标是判断我们能否从 S出发，经过至多 b 次令牌滑动，到达图 G 的一个大小为 k 的支配集。我们将支配集发现的一个实例记为 $(G,\ S,\ b)$，并在正文中明确给出参数。

5.3 难解性

我们首先证明，即使限定在最大度数为五的平面图类中，该问题仍然是 NP 完全的。

> 定理 5.1。 在最大度数为五的平面图类中，支配集发现问题是 NP 完全的。

证明。 我们从平面图上最大度数为三的支配集（DS）问题归约而来，该问题已知是 NP 完全的 [37]。给定一个 DS 实例 (G, κ)，其中 G 是最大度数为三的平面图，我们按如下方式构造一个 DSD 实例。我们创建一个新图 H，它由 G的一份副本构成。然后，对于每个顶点 $v \in V(H)$，我们创建一条由顶点 $\{w_v, x_v, y_v, z_v\}$ 构成的新路径，并将 v 连接到 w_v。接着，我们再创建一个新顶点 u_v，并将 u_v同时连接到 v和 x_v。我们选择 $S = \{x_v, y_v, \mid v \in V(G)\}$，并设定预算$b = 2\kappa$。注意， $k = 2|V(G)|$。至此，实例 (H, S, b) 的构造完成。由构造可知，图 H 是平面图，且最大度数为五。

假设 G 有一个大小至多为 κ的支配集 D。那么，在 H 中，我们可以将 x_v 上的每个令牌滑动到 v，其中 $v \in D$。由于 D 的大小至多为 κ，我们至多需要进行 2κ 次滑动。要说明所得集合是 H的一个支配集，只需注意，顶点对 w_v 和 u_v中的每一对都由 x_v 或 v支配。此外，顶点对 y_v 和 z_v 中的每一对都由 y_v支配。顶点 x_v 要么由 x_v 支配，要么由 y_v 支配（取决于是否 $v \in D$）。由于 D是 G的一个支配集，所有顶点 v 也都受到支配，正如所需。

对于反向方向，假设 (H, S, b) 是 DSD 的一个是实例。注意，将 x_v 上的令牌移动到 w_v 或 u_v 都会留下一个未被支配的顶点。此外，为了支配所有顶点 $\{u_v, w_v, x_v, y_v, z_v\}$，我们至少需要两个令牌。这意味着，在所得配置中，对于每个 v，我们要么在 v 上有一个令牌，要么在 x_v上有一个令牌；第二个令牌要么位于 y_v 上（如果 v 上有令牌且 x_v上没有令牌），要么可能位于 e_v 上（如果 x_v 上的令牌没有滑动到 v）。由于 $b = 2\kappa$ 且从 x_v 到 v 的距离为二，至多有 κ 个令牌可以滑动到对应于 G中顶点的顶点。这些顶点必须构成 G的一个支配集，正如所需。 □

接下来我们证明，当以 $k + b$ 为参数时，该问题从参数化视角来看同样是困难的。

定理 5.2。 即使限制在二分图类上，支配集发现问题在以 $k + b$ 为参数时仍是 W[2]-困难的。

证明。我们给出一个从 DS 问题出发的参数化归约；已知该问题在一般图上是 W[2]-困难的。给定一个 DS 实例 (G, κ)，我们按如下方式构造一个 DSD 实例 (H, S, b)。首先， H 包含 G的顶点集的两个副本。我们将这两个集合分别记为 L 和 R。在 R 中，我们将每个顶点 $v \in L$ 与 $N[v] = \{v' \in R \mid \{v, v'\} \in E(H)\} \cup \{v\}$ 相连，也就是说，将 v 与其在 R 中闭邻域内的每个顶点相连。然后，我们令 $k = \kappa + 1$，并添加 k 个新顶点 $v_1, \ldots, v_k$，使得每个 v_i 都与 L 中的所有顶点相邻。我们再添加一个顶点 u，使 u 有 $\kappa + 1$ 个悬挂邻居 $\{w_1, \ldots, w_{\kappa+1}\}$ 与之相连。我们令 u 与 L 中的所有顶点相邻。我们选择 $S = \{u, v_1, v_2, \ldots, v_k\}$（因此 $k = \kappa + 1$），并令 $b = \kappa$。至此，构造完成。图 H 确实是二分图，其二部划分为 $(L \cup \{w_i \mid i \in [\kappa + 1]\}\ R \cup \{u\} \cup \{v_i \mid i \in \kappa\})$。

假设 G 有一个大小为 κ的支配集。令 D 表示这样一个集合。于是，我们可以通过将 v_i 上的每个令牌滑动到 L中 D 的某个顶点副本上，来求解该发现实例。这些顶点将支配 $R \cup \{v_i \mid i \in \kappa\}$，而顶点 u 支配 $L \cup \{w_i \mid i \in [\kappa + 1]\}$。反过来，不失一般性地，假设在所得配置中，$u$ 上有一个令牌，而 R上没有令牌。这是因为，如果 u 上没有令牌，就需要将 $\kappa + 1 > b$ 个令牌移动到其邻居 $\{w_1, \ldots, w_{\kappa+1}\}$ 上，而这是不可能的。此外， R 中的每个令牌只能支配自身，因为 L 已经被 u支配。因此， R 中的每个令牌都可以改为以更少的移动次数放置在 L 中同一顶点的副本上，并且可能支配更多顶点。综上，我们可以假设所得配置包含 u上的一个令牌，以及 L 中的至多 κ 个令牌，而这些令牌必须支配 R的所有顶点，正如所需。

□

对于下一个结果，我们采用从顶点覆盖到支配集的标准归约，将 2-退化二分图上的顶点覆盖发现归约为 2-退化图上的支配集发现。

定理 5.3： 即使限制在 2-退化图类上，支配集发现问题在以预算 b 为参数化时也是 W[1]-困难的。

证明。 给定一个 VCD 实例（G， S， b），我们构造一个 DSD 实例（H， S， b），其中 H 通过对每条边 $\{u, v\} \in E(G)$ 添加一个新顶点 e_{uv}，并将 e_{uv} 与 u 和 v 分别连接而得到。由于我们不会增加 G 中顶点之间的距离，并且不失一般性地可以假设 H 的支配集不包含细分顶点，即形如 e_{uv} 的顶点，因此结论成立。注意，我们从一个 2-退化二分图 G 开始，并且只添加度为二的顶点（形成三角形），因此 H 同样是 2-退化的（但不是二分图）。

□

5.4 可处理性

我们关于以 k 为参数的支配集发现所得到的正面结果，利用了 k-支配核这一概念。据我们所知，这一概念最早可追溯至 Dawar 和 Kreutzer [23]。

定义 5.1。 设 G 是一个图，且 $k \geq 1$。若集合 $C \subseteq V(G)$ 满足：每个大小至多为 k 、支配 C 的集合也支配 G，则称其为 k-支配核 。

然而，一般图中不存在有界大小的支配核；不过，许多重要图类中确实存在这类支配核，例如，最一般地说，对于不含半梯的图也是如此 [52, 67]，[32]。

> 定理 5.4 ([32])。 设 $\mathscr{C}$ 是一个半梯指标有界的图类。那么存在一个函数 f，使得对于每个图 $G \in \mathscr{C}$ 和 $k \in \mathbb{N}$，我们都能在多项式时间内计算出 G 的一个 k-支配核，其大小为 $f(k)$。

> 定理 5.5。 当以 k 为参数且限制在无半梯子图上时，支配集发现问题是固定参数可处理的。

证明。令 (G, S, b) 表示 DSD 的一个实例。首先，我们计算一个大小为 $f(k)$ 的支配核心 C，这得益于定理 5.4。然后，我们计算关于 C 的投影类。也就是说，我们计算一个由顶点集合/类组成的族，使得任意两个顶点 $u, v \in V(G) \setminus C$ 属于同一个集合/类，当且仅当 $N(u) \cap C = N(v) \cap C$。投影类的数量也仅由 k 界定；事实上，一个显然的上界是 $2^{f(k)}$，但已有更好的上界。根据支配核心的定义，不失一般性地，我们可以假设 G 的任意大小至多为 k 的极小支配集，每个投影类至多包含一个顶点。因此，我们现在可以将每个投影类视为单个顶点，枚举所有大小至多为 k 的极小支配集。由于 $|C|$ 和投影类的数量均由 k 的某个函数界定，因此可以在由 k 的某个函数界定的时间内完成这种暴力枚举。令 D 表示一个支配集，它由来自 C 的顶点以及代表投影类的顶点组成。现在我们构造如下二分图 H。一侧是 S 的顶点，另一侧是 D 的顶点。我们将每个 $u \in S$ 与一个顶点 $v \in D \cap C$ 相连，并将从 u 到 v 的最短路径长度赋为边权。我们将每个 $u \in S$ 与一个顶点 $v \in D \setminus C$ 相连，并将从 u 到对应投影类中任意顶点的最短路径长度赋为边权；若 u 属于该投影类，则边权为零。证明的其余部分与 ISD 或 VCD 的情形完全相同，即，我们寻找权重至多为 b 的完美匹配；该匹配的存在意味着可以沿着匹配顶点对之间最短路径上的边滑动令牌。

□

同样，采用与有界树宽图上的顶点覆盖发现问题（定理 3.4）所用的相同动态规划技术，并结合在结构上处处稀疏类上进行一阶模型检测的归约（定理 3.5），我们得到以下结果：

> 定理 5.6。支配集发现问题可以在时间$2^{\mathcal{O}(t \text{ 对数 } k)} \cdot n^{\mathcal{O}(1)}$内求解，其中 t 表示输入图的树宽。

> 定理 5.7。当以 b 为参数，并限制在结构上处处稀疏的图类中时，支配集发现问题是固定参数可处理的。

6 着色发现

到目前为止，所有考虑的基础问题的可行解都是满足某些特定问题属性的顶点子集。事实上，我们的框架更加丰富，可以应用于更广泛的问题类别。我们转向非常基础的着色问题，以此展示我们的解发现框架的丰富性；在该问题中，可能的解不再是顶点子集。

图 G 的一个 k-**着色**是一个映射 $\varphi\colon V(G) \to [k]$。如果对于任意 $\{u, v\} \in E(G)$，都有 $\varphi(u) \neq \varphi(v)$，则称 k-着色是**正规的**。在 NP 完全的着色问题中 [37]，，给定一个图 G 和一个整数 k，目标是判断 G 是否存在一个正规的 k-着色。

6.1 相关工作

着色问题的重配置变体是组合重配置文献中的一个核心问题 [68, 61]。对于可行（及不可行）着色之间邻接关系的定义有很多种。我们考虑的重配置步骤（参见令牌模型）定义如下。在**颜色翻转模型**中，每一步都可以将任意顶点的颜色改为颜色集 $[k] = \{1, \ldots, k\}$ 中的任意颜色。在**颜色交换模型**中，(u, v)-颜色交换允许在任意两个顶点 $u \in V(G)$ 和 $v \in V(G)$ 之间交换颜色，其中 $u \neq v$。最后，在**颜色滑动模型**中，(u, v)-颜色滑动是沿图的一条边进行的 (u, v)-颜色交换。换言之，在颜色滑动中，只有当 $\{u, v\} \in E(G)$ 时，我们才能执行 (u, v)-颜色交换。

颜色翻转模型是着色问题重配置变体中研究最为广泛的模型之一。颇为出人意料的是，在该模型中，对于 $k \leq 3$ [19]，着色重配置问题可在多项式时间内求解；而对于 $k \geq 4$ [12]，该问题是 PSPACE-完全的。颜色滑动模型在 [11] 中进行了研究。据我们所知，颜色交换模型尚未在重配置框架中得到考察。

着色问题的发现变体已经在颜色翻转模型（k-Fix [38]）和颜色交换模型（k-Swap [24]）中以不同名称得到研究。Garnero 等人 [38] 表明，对于颜色翻转，发现变体是 NP 完全的，即使在二分平面图上也是如此。此外，他们还表明，该问题在以 b 为参数时是 W[1]-困难的，即使在二分图上也是如此；而在以$k + b$为参数时，它是固定参数可处理的。有趣的是，在颜色交换模型中，后者并不成立；在该模型下，以 b [24] 为参数时，无论固定 $k \geq 3$ 为何，该问题都是 W[1]-困难的。研究还表明，这两个

问题 k-Fix 和 k-Swap 在以输入图的树宽为参数时都是 W[1]-困难的。最后，研究表明， k-Fix 问题，即颜色翻转模型中的发现问题，在使用 $k \leq 2$ 种颜色时可以多项式时间求解，但颜色交换和颜色滑动模型中的这一问题仍未解决 [38]。与我们的工作相关的还有色数恶行这一概念 [21]。鉴于以上内容，除非另有说明，我们主要考虑颜色滑动模型中的着色发现。

6.2 记号与定义

在着色发现（CD）问题中，给定图 $G = (V, E)$ 的一个（非合法） k-着色 $\varphi\colon V \to [k]$以及一个预算 $b \in \mathbb{N}$。任务是判断是否可以通过至多 b 个重新着色步骤，将给定着色转换为一个合法的 k-着色。我们用（G, φ, b）表示着色发现的实例，并在正文中明确参数。如果希望得到一个使用（至多） q 种颜色的着色，我们还可以添加一个额外的输入参数 q 。不过，这与我们所关注的颜色滑动模型无关；颜色滑动会保持图中的颜色数量不变。

6.3 难解性

即使要证明颜色滑动模型中的问题属于 NP，也并非易事。我们将证明以下关键性质：要将正确的颜色滑动到一个顶点，沿边至多需要进行 $2n$ 次颜色滑动。因此，我们可以假设 $b \leq 2n^2$ ，并且重新着色序列是一个可在多项式时间内检查的证书。

命题 6.1。 在颜色滑动模型下，着色发现属于 NP。

证明。 我们证明，可以假设 b 在 n 中有多项式界，并使用重新着色序列作为证书。事实上，为了确定一个顶点的颜色，沿边至多需要进行 $2n$ 次颜色滑动。为此，考虑一条从 u 到 v 的路径，我们的目标是使用 v 的颜色为 u 着色。设该路径为（u, w_1, w_2, ..., w_ℓ, v）。我们通过以下滑动序列交换 u 和 v 的颜色：

1. 首先依次滑动（u, w_1）、（w_1, w_2）、（w_2, w_3）...，（w_ℓ, v）。完成这些滑动后，v 获得了 u 的原始颜色，路径上的其他所有顶点则获得其后继顶点的原始颜色。2. 随后，执行滑动（$w_{\ell-1}$, w_ℓ）、（$w_{\ell-2}$, $w_{\ell-1}$）...，（w_1, w_2）、（u, w_1）。完成这些滑动后，u 获得 v 的原始颜色， v 仍保有 u 的原始颜色，其他所有顶点都重新着色为其原始颜色。

因此，如果存在重新着色序列，至多需要 $2n^2$ 次颜色滑动。 □

我们证明，在所有三种模型中， $k \geq 3$ 着色发现问题都是 NP 完全的，即使在平面二分图上也是如此。对于颜色翻转模型和颜色交换模型，这一点最近已经得到证明 [38, 24]。我们的证明统一了所有三种模型下的困难性归约。我们通过从列表 k-着色（LikC）问题进行归约来证明 NP 困难性。该问题给定一个图 $G = (V, E)$ 以及分配给每个顶点 $u \in V(G)$ 的颜色列表 $L(u) \subseteq [k]$ ，询问是否存在一种合法着色 φ ，使得对于 G 的每个顶点 u ， G 都具有 $\varphi(u) \in L(u)$。已知 LikC 问题即使对于平面二分三正则图 [20] 也仍然是 NP 完全的 $k \geq 3$ 。

定理 6.1。 在所有三种模型下，使用 $k \geq 3$ 种颜色的着色发现问题都是 NP 完全的，即使在平面二分图类上也是如此。

证明。对于 $k \geq 3$，设 $(G, L = (L_u \subseteq [k] \mid u \in V(G)))$ 是 LikC 问题的一个实例，其中图 G 是平面二分图。我们构造一个 CD 实例 (H, φ, b)，使得$b = n(n+1)$，其中 H 是平面二分图，φ 是其顶点的一个初始（非合法） k-着色。定义函数 $f\colon [k] \to [k]$，使得 $f(c) = c+1$ 对 $c < k$ 成立，且$f(k) = 1$。我们从图 H 开始，它是 G的一个副本。 H 中的每个顶点都着色为 k。对于每个顶点 $u \in V(G)$，我们向H中对应的顶点 u添加 $k-1$ 个邻居 u_1、.....、u_{k-1}。对于每个 $i \in [k-1]$，新顶点 u_i 分别着色为颜色 i。对于每种颜色$c \in L(u)$，我们向 u 添加 n 个邻居，将它们记为 $\{x_c^i \mid i \in [n]\}$，并赋予它们颜色 c。每个顶点 x_z^i 都有一个邻居 y_z^i，其着色为颜色 $f(c)$。对于每种颜色$c \notin L(u)$，我们向 u添加 $b+1$ 个邻居，将它们记为 $\{x_c^i \mid i \in [b+1]\}$，并赋予它们颜色 c。至此，图 H的构造完成。图 H 是平面二分图，因为 G 是平面二分图。图 4 展示了 $k = 3$ 以及颜色集 {绿色, 蓝色, 红色}对应的构件。

现在我们证明该归约的正确性。首先，我们证明如果 $(G,\ L)$ 是 LikC 问题的一个是实例，那么 $(H,\ \varphi,\ b)$ 是 CD问题的一个是实例。设 $\psi\colon\ V(G) \to [k]$ 是实例 $(G,\ L)$ 的一个可行解。我们说明如何从（不可行的）着色 φ 开始，至多经过 b 步，为 H 找到一个合法着色。对于每个顶点 $u \in V(G)$，如果 $\psi(u) \neq$ k，我们就交换 u 和 $u_{\psi(u)}$ 之间的颜色。该交换确保 $H[V(G)]$ 得到合法着色。此外，对于每个顶点 $u \in V(G)$，以及 $i \in [k-1]$，顶点 u_i 在局部上与合法着色一致，即由这些顶点诱导的子图中不存在单色边。进一步地，对于每个 $i \in [n]$，我们交换 $x^i_{\psi(u)}$ 和 $y^i_{\psi(u)}$之间的颜色。这样，对于 $V(G)$，每个顶点至多进行 $1+n$ 次颜色交换，即可得到一个合法的 k-着色。

我们通过证明 CD 问题的一个是实例对应于 LikC 问题的一个是实例来完成证明。根据定义，如果能使构造图中的着色合法，我们也能得到图 G 的合法 k着色。我们只需证明，对于某个不在列表 $L(u)$ 中的顶点 u，我们不会使用颜色 c。然而，如果 $\varphi(u) = c$，我们就必须对所有顶点 x_c^i 重新着色 $i \in [b+1]$，这与预算 b 矛盾。

注意，给定 LikC 的一个是实例，在颜色滑动、颜色交换和颜色翻转模型中，预算 $n(n+1)$ 都足够。反过来，在颜色翻转和颜色交换模型中，我们无法在预算为 b 的情况下对所有顶点 x_c^i 重新着色 $i \in [b+1]$。 □

现在，我们考虑 Coloring Discovery 问题关于参数 k、 b 和树宽的参数化复杂性。下一个结果可由定理 6.1 和命题 6.1 直接推出。

推论 6.1。 由 $k \geq 3$ 参数化的着色发现问题在三种模型中均为 para-NP完全，即使限制在平面二分图上也是如此。

定理 6.2。 在颜色滑动模型中，以 $k+b$ 为参数的着色发现是 W[1]-困难的。

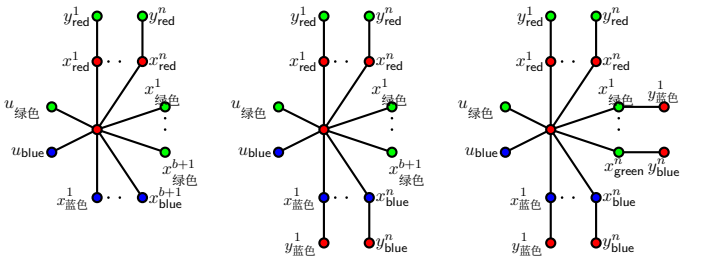

图 4：针对不同顶点列表大小的构件，以及 $k = 3$

证明。我们从多彩独立集问题归约而来；已知该问题是 W[1]-完全的。设 (G, κ) 是该问题的一个实例，其中 $V(G)$ 被划分为独立集 $V_1, \ldots, V_\kappa$。不失一般性，我们假设每个集合 V_i 的大小至少为 $3\kappa + 1$；否则，我们可以适当地添加顶点。我们按如下方式构造一个 CD 实例 (H, φ, b)。首先，令 H 为 G 的一个副本。然后，对于 $1 \leq i \leq \kappa$，我们添加新顶点 u_i；对于 $1 \leq j \leq \kappa + 1$，我们添加新顶点w_j。对于 $1 \leq i \leq \kappa$，我们将 u_i 与 w_j 相连，并将 $1 \leq j \leq \kappa + 1$ 相连。我们还添加边，将 $1 \leq j \leq \kappa + 1$ 中的 w_j 个顶点构成一个大小为 $\kappa + 1$ 的团。对于每个 $i \in [\kappa]$，我们将 u_i 与 V_i 中的每个顶点相连。我们用颜色 i 为 V_i 中的所有顶点着色，用颜色 $\kappa + 1$ 为顶点 u_i,$i \in [\kappa]$ 着色，用颜色 $\kappa + 2$ 为顶点 w_j, $j \in [\kappa]$ 着色，并用颜色 $\kappa + 1$ 为顶点 $w_{\kappa+1}$ 着色。因此，我们有 $k = \kappa + 2$，并将预算 b 设为 2κ。

令 $I = \{v_1、 v_2、 \ldots v_\kappa\}$ 表示 G 中的一个多彩独立集，其中每个顶点 v_i 属于 V_i，对于 $i \in [\kappa]$。为了从 φ 出发并使用 $b = 2\kappa$ 次滑动得到合法着色，我们按如下方式进行。首先，对于 $i \in [\kappa]$，执行滑动 $(v_i、 u_i)$。接下来，对于 $i \in [\kappa]$，执行滑动 $(u_i、 w_i)$。注意，现在 w_j、$j \in [\kappa]$ 中的每个顶点在 $[\kappa]$ 中都有不同的颜色，所有顶点 u_i、 $i \in [\kappa]$ 的颜色均为 $\kappa + 2$，顶点 $w_{\kappa+1}$ 的颜色为 $\kappa + 1$，而 I 的顶点颜色为 $\kappa + 1$，这就是 H 的一个合法着色。

对于反向证明，假设（H， φ， b）是 CD 的是实例。在 $b = 2\kappa$ 次滑动内得到合法着色的唯一方法，是给顶点 $\{w_j \mid j \in [\kappa + 1]\}$ 着色，使用 $\kappa + 1$ 种不同的颜色，因为这些顶点构成一个团。由于这些顶点各自有 κ 个邻居被着成 $\kappa + 1$，我们需要从 G 中“偷取”恰好 κ 种颜色。由于必须偷取 κ 种互不相同的颜色，每种颜色都必须来自不同的 V_i 集合 $i \in [\kappa]$。注意，将一种颜色从某个 V_i 滑动到某个顶点 w_j 需要预算为 2。此外，在得到的合法着色中，顶点 u_i 不能使用颜色 i 着色，因为 u_i 至少与 3κ 个颜色为 i 的顶点相邻（根据我们的假设，即 $|V_i| \geq 3\kappa + 1$）。此外，顶点 w_j 和 w'_j 不能获得相同的颜色，顶点 u_i 和 w_j 也不能获得相同的颜色。综合以上结论，必然存在 κ 个顶点，每个来自一个 V_i 集合，它们可以着成颜色 $\kappa + 1$，从而构成一个独立集。这对应于 G 中的一个多彩独立集。 □

定理 6.3。 参数化于输入图的树宽的着色发现在颜色滑动模型中是 W[1]-难的。

证明。我们沿用 [24] 中针对 k-Fix 和 k-Swap 问题所提出的相同策略。在预着色扩展（PrExt）问题中，给定一个图 $G = (V, E)$、一个预着色顶点集合 $W \subseteq V(G)$，以及一个预着色 $c\colon W \to [r]$，$r \geq 2$，用于对 W 中的顶点进行着色。目标是判断是否存在一个对 G 的合法 r-着色 c'，使其扩展着色 c（即，对每个 $v \in W$，$c'(v) = c(v)$）。已知当以输入图 [34] 的树宽为参数时，PrExt 是 W[1]-难的。

设 (G, W, c) 是 PrExt 的一个实例，并且不失一般性地假设 c 是 $G[W]$ 的一个合法 r-着色。我们按如下方式构造 CD 的一个实例 (H, φ, b)。首先，令 H 为 G 的一个副本，并令 $k = r$ 表示可用颜色的数量。现在，对于每个顶点 $v \in W$，我们选择 $\varphi(v) = c(v)$。对于每个顶点 $v \in W$，我们添加 $(r-1)\,n$ 个悬挂顶点，其中每组 n 个顶点分别接收 $[r] \setminus c(v)$ 中的一种颜色。注意，只要我们将预算设为 $b \leq n$，所添加的悬挂顶点就能保证 W 中的任何顶点都无法改变颜色。最后，对于每个顶点 $u \in V(G) \setminus W$，我们添加 $r-1$ 个悬挂顶点，其颜色从 1 到 $r-1$，并将顶点 u 的颜色指定为 r，即所有顶点 $u \in V(G) \setminus W$ 都接收颜色 r。我们设定 $b = n - |W|$，由此完成该构造。不难看出，这两个实例确实是等价的，结论也随之成立，因为一个图中添加度至多为一的顶点不会增加其树宽，这是一个众所周知的事实。

□

6.4 可处理性

我们考虑使用 $k = 2$ 颜色的着色发现问题。为简化记号，在本小节中我们假设颜色集为红色，绿色。已知，在颜色翻转模型 [38]下，$k = 2$ 的着色发现问题可以在多项式时间内求解。我们将证明，对于另外两种重配置模型，即颜色交换和颜色滑动，这一结论同样成立。一个图可进行 2-着色，当且仅当它是二分图。我们可以通过图搜索在多项式时间内检查给定图 G 是否为非二分图，并在这种情况下判定其为否实例。因此，下面我们将自己限制在二分图上。

定理 6.4。 使用 $k = 2$ 的着色发现问题在颜色交换模型中可以在多项式时间内求解。

证明。 对于二分输入图 G 的每个连通分量 G_i，我们找到其 (L_i, R_i) 二部划分，其中 $V(G_i)$。任何合法着色都会将 L_i 的所有顶点着为红色，并将 R_i 的所有顶点着为绿色，或者反之。

重新着色所需的预算，以及在合法着色中被着成红色和绿色的顶点数量，对于两种选择而言都可能不同。对于给定的选择，所需的交换次数和颜色数量都可以直接计算。由于一次交换总是交换红色和绿色，因此所需的总预算就是必须重新着色为绿色的红色顶点数量。根据两种可能性，我们将把连通分量 G_i 正确重新着色所需的预算分别记为 b_1^i 或 b_2^i。此外，对于每种选择，我们将连通分量 G_i 的超额定义为：最初被着成红色的顶点数减去必须被着成红色的顶点数，即，

$$e_1(G_i) = |\{v \in V(G_i) \mid \varphi(v) = \mathsf{red}\}| - |L_i|\,, \text{ and}$$
$$e_2(G_i) = |\{v \in V(G_i) \mid \varphi(v) = \mathsf{red}\}| - |R_i|\,.$$

因此，根据两种可能的选择，我们要么需要预算 b_1^i ，并得到超额 $e_1(G_i)$ ，要么需要预算 b_2^i ，且超额为 $e_2(G_i)$ 。总体而言，当且仅当存在一个选择集合，使得总超额为 0 且所需预算至多为 b 时，我们才能在给定预算下对整个图重新着色。更形式化地说，当且仅当存在一个向量 $\mathbf{x} \in \{1,2\}^q$ ，使得图 G （含有 q 个连通分量）能够被正确地重新着色，并且

$$\sum_{i=1}^{q} e_{x_i}(G_i) = 0 \text{ , and}$$
$$\sum_{i=1}^{q} b_{x_i}^i \leq b \text{ .}$$

在证明的剩余部分中，我们描述一种用于求解决策问题的动态规划。我们用一个由整数构成的三元组 $(x、\ y、\ z)$ 定义动态规划的一个状态。该三元组对应于这样的事实：存在针对连通分量 G_1、.....、G_x 的重新着色选择，使得超额之和为 y ，且所需预算为 z。我们称状态 (0、0、0) 是 可行的。基于每个连通分量的两种可能选择，我们可以从每个状态到达另外两个可行状态。如果 $(x、\ y、\ z)$ 可行，则 $(x+1、\ y+e_1(G_{x+1})\ z+b_1^{x+1})$ 和 $(x+1、\ y+e_2(G_{x+1})\ z+b_2^{x+1})$ 也可行。注意，最多存在 $|V(G)|^3$ 个可行状态，因为连通分量的数量、总超额和预算分别都不可能超过 $|V(G)|$。

通过图搜索某个状态（q、0，z），其中 q是连通分量的数量，且$z \leq b$，我们便可解决该问题。 □

> 定理 6.5。在颜色滑动模型下，具有 $k = 2$ 的着色发现问题可在多项式时间内解决。

证明。在颜色滑动模型中，我们可以独立处理各个连通分量。因此，假设 G 是连通的。首先，我们计算 G的一个二部划分（L，R）。原则上，G有两种可能的正确二着色：要么当且仅当一个顶点属于 L 时将其着为红色，要么当且仅当它属于 R时将其着为绿色。我们分别处理这两种情况，并选择成本较小的一种。对所有连通分量求和即可得到结果。现在假设 L 中的所有顶点都应着为红色。另一种情况类似。

我们将通过在调整后的网络中计算最小费用最大流来解决 CD问题。首先注意，在 L 中需要重新着色的顶点数量必须与在 R中需要重新着色的顶点数量相同。否则，该 CD实例无解。将这个数量记为 q。给定一个带有初始着色的图 $G = (V, E)$，我们按如下方式构造最小费用最大流实例。复制图 G，添加一个源 s ，并用一条成本为 0、容量为 1 的弧将其连接到所有着色为绿色、即需要重新着色的 $v \in L$ 。添加一个汇点 t，并从每个着色错误、即着色为红色的 $v \in R$向其添加入弧。这些弧的容量为 1，成本为 0。对于所有原始边 $\{u, v\}$，添加弧 (u, v) 和 (v, u)，二者的成本均为 1，容量均为 q。随后，我们计算从 s 到 t的最小费用最大流。我们将证明，当且仅当着色发现实例是是实例时，存在流值为 q 且成本至多为 b 的流。该归约还给出了构造性证明，即我们可以计算出一系列交换，从而完成重新着色。注意，计算最小费用最大流可以在多项式时间内完成 [66]。

首先，我们证明，若存在流 f，其 $|f| = q$ 且成本为 b，那么我们可以对图重新着色，并通过 b 次交换得到正确的二着色。对于给定的流 f，我们仅沿正流量边计算从 s 到 t 的最短路径。设 $(s, v_1, v_2, v_3, \ldots, v_\ell, t)$ 表示这样一条最短路径。由于它是最短路径，因此有如下性质：初始时，$v_1, v_2, v_4, v_6, \ldots, v_{\ell-2}$ 着色为绿色，而 $v_3, v_5, \ldots, v_{\ell-1}, v_\ell$ 着色为红色。否则，如果任何一个奇数的 v_i, $i \geq 3$ 着色错误，那么就存在一条从 s 到 v_i的边。同理，对于偶数的 $i \leq \ell - 2$ 和 i，如果 v_i 着色错误，那么就存在一条从 v_i 到 t的边，同样会导致矛盾。因此，我们可以如下交换颜色。首先，沿边 $(v_2, v_3) \ldots, (v_{\ell-2}, v_{\ell-1})$ 交换颜色。经过这 $\ell - 2/2$ 次交换后，节点 v_1, $v_3, v_5, \ldots, v_{\ell-1}$ 着色为红色，而 $v_2, v_4, \ldots, v_\ell$ 着色为绿色。现在，沿边 $(v_1, v_2), (v_3, v_4) \ldots, (v_{\ell-1}$, $v_\ell)$ 进行颜色交换。经过这 $\ell/2$ 次交换后，当且仅当 i 为奇数时，顶点 v_i 的颜色为红色。因此，总共进行了 $(\ell - 2)/2 + \ell/2 = \ell - 1$次交换，即路径的长度。此外，这些交换恰好改变顶点 v_1 和 v_ℓ 的颜色，所有其他顶点则保持原有颜色。因此，路径上的所有顶点在交换后都正确着色，并且 q 减少 1。我们将路径上的流量减少 1，然后继续处理下一条最短路径。总之，经过恰好 b 次交换后，我们将得到一个合法着色。

其次，我们证明，如果可以通过 b 次滑动对图进行重新着色，就能在构造的网络中得到一个流 f，其满足 $|f| = q$，且成本为 b。假设给定一个由 b 次滑动组成的滑动序列，可将图重新着色为正确的二着色。根据该序列，我们逐步沿着从 s 到 t的路径增加流量，从而构造出这个流。我们从零流开始。令 e_1、e_2、.....、e_b 表示滑动序列中的边。注意，同一条边可能在序列中出现多次。执行滑动之前，假设顶点都已分配标识符，并将每个顶点的标识符与其颜色关联起来。滑动颜色时，我们将标识符与颜色一并交换。具体来说，考虑一个顶点 v_i，其初始颜色为某种颜色，例如 L中的绿色。经过一次滑动（v_i，v_j）后，顶点 v_i 被染成红色。这个新颜色所对应的 v_i 的标识符指向最初被染成该颜色的 v_j。通过在滑动过程中追踪标识符，我们可以得到一条从每个顶点通往新颜色来源的路径。对于在滑动过程中从 v_j 获得颜色的顶点 v_i，我们将该路径记作 p（v_i，v_j）。如果 $v_i \in L$，也就是说，v_j 要么是 R中最初着色错误的节点，要么是 L中着色正确的节点。如果 $v_j \in R$，我们就停止，并沿路径（s，v_i） p（v_i，v_j）、（v_j，t）增加流量。否则，$v_j \in L$，也就是说，现在被染成红色。按照上文所述的方法追踪 v_j 的新颜色的标识符，可得到一条通往某个节点 v_ℓ 的路径。我们继续这一过程，直到到达 R中的某个节点 $v_{\ell'}$。注意，标识符各不相同，因此这一过程不会形成环。通过追踪整条拼接路径（s，v_i） p（v_i，v_j） p（v_j，v_ℓ）.....、（v'_ℓ，t），我们得到一条从 s 到 t 的路径，并沿此路径将流量增加 1。接着，我们处理 L 中下一个最初被染成绿色的节点。最终，我们得到流值为 q、成本为 b 的流，因为每条被某个流单位使用且成本为正的边，都对应于滑动过程中的一次滑动。

□

利用与定理 3.5 中所用类似的技术，我们可以证明以下结果。请注意，与定理 3.5 不同，这里我们需要将 k 作为参数的一部分，以便一阶公式能够访问实例中的所有 k 颜色。我们将这些颜色添加为一元谓词，使其能够被一阶逻辑访问。同样地，我们量化应当改变其颜色的 $q \leq \max\{k, b\}$ 个顶点，并通过长度为 $f(b)$ 的公式硬编码如何得到新着色。

定理 6.6。 在以 $k+b$ 为参数并限制于结构上处处稀疏的图类时，着色发现问题（在全部三种模型中）是固定参数可处理的。

7 结论与未来工作

我们已经证明了关于基本图问题的解发现变体的许多正面结果和困难性结果。尽管由我们的框架产生的一些问题已经在文献中得到研究（虽然使用的是不同的名称），但我们相信，从统一的视角来看待这些问题能够带来更具全局性的洞见，并有望得到更多统一的结果。

未来的工作中，我们计划将我们的框架与其他已确立的框架（例如动态问题）相结合，以对更多这类决策问题的动态现实世界应用进行分类。例如，对于每个实例，可以允许有两个输入图 G 和 G'，其中 G' 通过对 G 进行少量边/顶点的添加与删除而得到。此外，我们要求初始状态 S 在 G中满足某些性质。现在问题变成：我们能否使用至多 b 个重配置步骤将 S 转换为 S'，使得 S' 在 G'中满足某些/类似的性质。这一推广增加了我们分析这些问题时的自由度，并推动了多变量分析的发展，在这种分析中，图中的变化现在也可以成为参数的一部分。

另一个可以探索的方向是放宽我们对令牌位置施加的约束。例如，对于图顶点子集问题，我们可以允许多个令牌占据同一个顶点。这样一来，初始配置的大小就可以与所需目标可行解的大小相互独立。例如，独立集发现问题可以重新表述如下：给定一个图 G、一个初始配置 S、一个整数 $k \leq |S|$ 和一个预算 b，现在的目标是确定我们是否能够从 S 出发，并且至多使用 b 步，达到大小至少为 k 的独立集。类似地，对于翻转模型中的着色发现问题，可以额外加入一个输入$q < k$，并询问我们是否能够从一个（非合法的） k-着色出发，达到一个合法的 q-着色。

参考文献

[1] Karen Aardal、Stan P. M. van Hoesel、Arie M. C. A. Koster、Carlo Mannino 和 Antonio Sassano。频率分配问题的模型与求解技术。《运筹学年鉴》，153(1):79–129，2007年。[2] Emile Aarts 和 Jan Karel Lenstra。《组合优化中的局部搜索》。Chichester：Wiley，1997年。[3] Faisal N. Abu-Khzam、Judith Egan、Michael R. Fellows、Frances A. Rosamond 和 Peter Shaw。动态问题的参数化复杂性。《理论计算机科学》，607:426–434，2015年。

[4] Sepehr Assadi、Krzysztof Onak、Baruch Schieber 和 Shay Solomon。具有次线性更新时间的全动态极大独立集。在《STOC 2018》中，第815–826页。ACM，2018年。[5] Sayan Bhattacharya、Deeparnab Chakrabarty、Monika Henzinger 和 Danupon Nanongkai。图着色的动态算法。在《SODA 2018》中，第1–20页。SIAM，2018年。[6] Sayan Bhattacharya、Monika Henzinger 和 Giuseppe F. Italiano。顶点覆盖与匹配的确定性全动态数据结构。《SIAM Journal on Computing》，47(3):859–887，2018年。[7] Yaad Blum 和 Jeffrey S. Rosenschein。多智能体图着色：帕累托效率、公平性与个体理性。在《AAAI 2008》中，第24–29页。AAAI出版社，2008年。[8] Hans-Joachim Böckenhauer、Juraj Hromkovic 和 Dennis Komm。困难优化问题的再优化。在《近似算法与元启发式手册，第二版，第1卷：方法与传统应用》中，第427–454页。Chapman and Hall/CRC，2018年。[9] Marthe Bonamy、Paul Dorbec 和 Paul Ouvrard。令牌滑动下的支配集重配置。《离散应用数学》，301:6–18，2021年。[10] Édouard Bonnet、Nicolas Bousquet、Pierre Charbit、Stéphan Thomassé 和 Rémi Watrigant。H-自由图中独立集的参数化复杂性。《Algorithmica》，82(8):2360–2394，2020年。[11] Édouard Bonnet、Tillmann Miltzow 和 Paweł Rzazewski。令牌交换及其变体的复杂性。《Algorithmica》，80(9):2656–2682，2018年。[12]Paul Bonsma 和 Luis Cereceda。寻找图着色之间的路径：PSPACE-完全性与超多项式距离。《理论计算机科学》，410(50):5215–5226，2009年。[13] Nicolas Bousquet 和 Alice Joffard。圆图与圆弧图上的支配集令牌滑动重配置。在《FCT 2021》中，第114–134页。Springer，2021年。[14] Nicolas Bousquet、Amer E. Mouawad、Naomi Nishimura 和 Sebastian Siebertz。独立集及（连通）支配集重配置问题的参数化复杂性综述。《arXiv》预印本 arXiv:2204.10526，2022年。[15] C. Titus Brown、Dominik Moritz、Michael P. O’Brien、Felix Reidl、Taylor Reiter 和 Blair D. Sullivan。使用 spacegraphcats 探索大型宏基因组组装图，揭示隐藏的序列多样性。《基因组生物学》，21:1–16，2020年。[16] Shaowei Cai、Wenying Hou、Jinkun Lin 和 Yuanjie Li。通过动态策略改进最小权顶点覆盖的局部搜索。在《IJCAI 2018》中，第1412–1418页。ijcai.org，2018年。[17] Shaowei Cai、Jinkun Lin 和 Chuan Luo。在大型稀疏图中寻找小顶点覆盖：构造、局部搜索与预处理。《人工智能研究杂志》，59:463–494，2017年。

[18] Gruia Călinescu、Adrian Dumitrescu 和 János Pach。图与网格中的重配置。*SIAM离散数学杂志*，22(1):124–138，2008年。[19] Luis Cereceda、Jan Van Den Heuvel 和 Matthew Johnson。寻找 3-着色之间的路径。图论杂志，67(1):69–82，2011年。[20] Miroslav Chlebík 和 Janka Chlebíková。低度平面二分图上的困难着色问题。离散应用数学，154(14):1960–1965，2006年。[21] Sally A. Clark、John E. Holliday、Sarah H. Holliday、Peter Johnson、Janet E. Trimm、Robert R. Rubalcaba 和 Matt Walsh。图中的色数恶行。*Congressus Nu-merantium*，182:171，2006年。[22] Marek Cygan、Fedor V. Fomin、Lukasz Kowalik、Daniel Lokshtanov、Dániel Marx、Marcin Pilipczuk、Michal Pilipczuk 和 Saket Saurabh。参数化算法。Springer，2015年。[23] Anuj Dawar 和 Stephan Kreutzer。处处稀疏类中的支配问题。载于*FSTTCS 2009*，第157–168页。Sc hloss Dagstuhl - Leibniz计算机科学中心，2009年。[24] Marzio De Biasi 和 Juho Lauri。论恢复损坏着色的复杂性。*Jour-nal of Combinatorial Optimization*，37(4):1150–1169，2019年。[25] Erik D. Demaine、Martin L. Demaine、Eli Fox-Epstein、Duc A. Hoang、Takehiro Ito、Hiro- taka Ono、Yota Otachi、Ryuhei Uehara 和 Takeshi Yamada。树上滑动令牌的多项式时间算法。载于*ISAAC 2014*，第389–400页。Springer，2014年。[26] Erik D. Demaine、Felix Reidl、Peter Rossmanith、Fernando Sánchez Villaamil、Somnath Sikdar 和 Blair D. Sullivan。复杂网络的结构稀疏性：随机模型与现实世界图中的有界扩张。计算机与系统科学杂志，105:199–241，2019年。[27] Irit Dinur 和 David Steurer。并行重复的分析方法。载于*STOC 2014*，第624–633页。ACM，2014年。[28] Rod G. Downey 和 Michael R. Fellows。固定参数可处理性与完全性 II：关于 W[1] 完全性。理论计算机科学，141(1-2):109–131，1995年。[29] Rodney G. Downey 和 Michael R. Fellows。参数化复杂性。计算机科学专著。Springer，1999年。[30] Jan Dreier、Nikolas Mählmann 和 Sebastian Siebertz。结构稀疏图类上的一阶模型检测。*arXiv*预印本 *arXiv:2302.03527*，2023年。[31] Jack Edmonds 和 Richard M. Karp。网络流问题算法效率的理论改进。*ACM*杂志，19(2):248–264，1972年。[32] Grzegorz Fabianski、Michal Pilipczuk、Sebastian Siebertz 和 Szymon Torunczyk。支配与独立性的渐进算法。载于*STACS 2019*，第27:1–27:16页。Schloss Dagstuhl - Leibniz计算机科学中心，2019年。

[33] Matthew Farrell、Timothy D. Goodrich、Nathan Lemons、Felix Reidl、Fernando Sánchez Villaamil 和 Blair D. Sullivan。随机交图的双曲性、退化性与扩张。载于*WAW 2015*，第29–41页。Springer，2015年。[34] Michael R. Fellows、Fedor V. Fomin、Daniel Lokshtanov、Frances A. Rosamond、Saket Saurabh、Stefan Szeider 和 Carsten Thomassen。关于以树宽为参数的若干彩色问题的复杂性。《信息与计算》，209(2):143–153，2011年。[35] Jörg Flum 和 Martin Grohe。参数化复杂性理论。理论计算机科学丛书。EATCS丛书。Springer，2006年。[36] Philippe Galinier 和 Alain Hertz。图着色局部搜索方法综述。《计算机与运筹研究》，33(9):2547–2562，2006年。[37] M. R. Garey 和 David S. Johnson。《计算机与不可解性：*NP*完全性理论指南》。W. H. Freeman，1979年。[38] Valentin Garnero、Konstanty Junosza-Szaniawski、Mathieu Liedloff、Pedro Montealegre 和 Paweł Rzazewski。修复图的不正规着色。理论计算机科学，711:66–78，2018年。[39] Martin Grohe、Stephan Kreutzer 和 Sebastian Siebertz。判定无处稠密图的一阶性质。载于*STOC 2014*，第89–98页。ACM，2014年。[40] Arash Haddadan、Takehiro Ito、Amer E. Mouawad、Naomi Nishimura、Hirotaka Ono、Akira Suzuki 和 Youcef Tebbal。支配集重配置的复杂性。理论计算机科学，651:37–49，2016年。[41] William K. Hale。频率分配：理论与应用。*Proceedings of the IEEE*，68:1497–1514，1980年。[42] Sariel Har-Peled 和 Kent Quanrud。针对多项式扩张图和低密度图的近似算法。*SIAM Journal on Computing*，46(6):1712–1744，2017年。[43] Sepp Hartung 和 Rolf Niedermeier。以守恒性为参数的图增量列表着色。理论计算机科学，494:86–98，2013年。[44] Emmanuel Hebrard 和 George Katsirelos。子句学习与图着色的新界限。载于*IJCAI 2019*，第6166–6170页。ijcai.org，2019年。[45] Pei Huang、Minghao Liu、Ping Wang、Wenhui Zhang、Feifei Ma 和 Jian Zhang。在图着色的引导下求解模态逻辑 S5 的可满足性问题。载于*IJCAI 2019*，第1093–1100页。ijcai.org，2019年。[46] Takehiro Ito、Erik D. Demaine、Nicholas J. A. Harvey、Christos H. Papadimitriou、Martha Sideri、Ryuhei Uehara 和 Yushi Uno。论重配置问题的复杂性。理论计算机科学，412(12-14):1054–1065，2011年。[47] David S. Johnson。组合问题的近似算法。载于*STOC 1973*，第38–49页。ACM，1973年。

[48] Marcin Kaminski、Paul Medvedev 和 Martin Milanic。独立集重配置问题的复杂性。理论计算机科学，439：9–15，2012年。[49] Richard M. Karp。组合问题之间的可归约性。载于 Raymond E. Miller、James W. Thatcher 和 Jean D. Bohlinger 编，计算机计算复杂性，第 85–103 页。Springer，1972年。[50] Vedat Kavalci、Aybars Ural 和 Orhan Dagdeviren。无线传感器网络的分布式顶点覆盖算法。*International Journal of Computer Networks & Communications*，6：95–110，2014年。[51] Subhash Khot 和 Oded Regev。顶点覆盖问题可能难以近似到 2-ε 以内。载于 *Computational Complexity Conference*，第 379 页。IEEE Computer Society，2003年。[52] Stephan Kreutzer、Roman Rabinovich 和 Sebastian Siebertz。处处不稠密图类的多项式核与宽度性质。*Transactions on Algorithms*，15(2)：1–19，2018年。[53] R. Krithika、Abhishek Sahu 和 Prafullkumar Tale。动态参数化问题。*Algorithmica*，80(9)：2637–2655，2018年。[54] Harold W. Kuhn。指派问题的匈牙利算法。*Naval researchlogistics quarterly*，2(1-2)：83–97，1955年。[55]David Lichtenstein。平面公式及其用途。*SIAM Journal on Computing*，11(2)：329–343，1982年。[56] Daniel Lokshtanov、Fahad Panolan、Saket Saurabh、Roohani Sharma 和 Meirav Zehavi。覆盖小独立集与分隔集及其在参数化算法中的应用。*ACM Transactions on Algorithms*，16(3)：32:1–32:31，2020年。[57] László Lovász。最优整数覆盖与分数覆盖之比。离散数学，13(4)：383–390，1975年。[58] Chuan Luo、Holger H. Hoos、Shaowei Cai、Qingwei Lin、Hongyu Zhang 和 Dongmei Zhang。通过高效自动配置进行最小顶点覆盖局部搜索。载于*IJCAI 2019*，第 1297–1304 页。ijcai.org，2019年。[59] Bojan Mohar。顶点图的面覆盖与亏格问题。组合理论杂志*B*辑，82(1)：102–117，2001年。[60]Amer E. Mouawad、Naomi Nishimura、Venkatesh Raman、Narges Simjour 和 Akira Suzuki。重配置问题的参数化复杂性。*Algorithmica*，78(1)：274–297，2017年。[61] Naomi Nishimura。重配置导论。算法，11(4)：52，2018年。[62] Krzysztof Onak 和 Ronitt Rubinfeld。维护大匹配与小顶点覆盖。载于*STOC 2010*，第 457–464 页。ACM，2010年。

[63] Neil Robertson 和 Paul D. Seymour。图小 minors V：排除一个平面图。组合理论杂志 *B*辑，41(1)：92–114，1986年。[64] Yunzhuang Shen、Yuan Sun、Xiaodong Li、Andrew S. Eberhard 和 Andreas T. Ernst。通过基于机器学习的定价启发式增强图着色的列生成。载于*AAAI 2022*，第 9926–9934 页。AAAI出版社，2022年。[65] Miklós Si monovits。极值图论。载于 Lowell W. Beineke 和 Robin J. Wilson 编，图论精选主题*2*，第 161–200 页。Academic Press Inc，1983年。[66] Éva Tardos。一种强多项式最小费用循环算法。*Combinatorica*，5(3)：247–256，1985年。[67] Jan Arne Telle 和 Yngve Villanger。稀疏图及更广泛图类上支配问题的 FPT 算法。理论计算机科学，770：62–68，2019年。[68] Jan van den Heuvel。变化的复杂性。组合学综述，409(2013)：127–160，2013年。[69] Yiyuan Wang、Shaowei Cai、Shiwei Pan、Ximing Li 和 Minghao Yin。加权图着色问题的归约与局部搜索。载于*AAAI 2020*，第 2433–2441 页。AAAI出版社，2020年。[70] Marcin Wrochna。有界带宽与树深度下的重配置。计算机与系统科学杂志，93：1–10，2018年。[71] Xiaojun Xie 和 Xiaolin Qin。一种基于超图最小顶点覆盖的动态特征选择算法。载于*PAKDD 2018*，第 40–51 页。Springer，2018年。[72] David Zuckerman。线性度提取器以及最大团和色数问题的不可近似性。载于*STOC 2006*，第 681–690 页。ACM，2006年。