arXiv:2601.00957v1 [math.CO] 2 Jan 2026

Tyshkevich 图分解的算法应用：入门与工具包

Christine T. Cheng 和 Chelsea Ann Lambert

电气工程与计算机科学系 威斯康星大学密尔沃基分校
ccheng@uwm.edu; lambe222@uwm.edu

摘要

完全由其度序列确定的图称为单图。2000 年，Regina Tyshkevich 发表了一篇关于单图的重要论文 [17]。这篇论文包含两部分：一项分解定理，阐述如何将每个图分解为一系列基本图；以及对所有基本单图的完整分类。二者共同揭示了每个单图是如何构造出来的。

我们对 Tyshkevich 的结果作非正式概述，并展示这些结果如何使我们能够在线性时间内计算单图的各种图参数。我们还提供了一个工具包（https://chelseal11.github.io/tyshkevich_decomposition_toolkit/），其中实现了本文介绍的算法。

1 引言

每个图 G 都有一个度序列 D_G，按从大到小的顺序列出各顶点的度。例如，图1中的树 T的度序列是 (3, 2, 1, 1, 1) $=(3,2,1^3)$。一般来说，一个图的度序列并不能唯一确定该图；两个或更多个图可能具有相同的度序列。以 8-圈 C_8为例。它与 $C_5 \cup C_3$ 和 $2C_4$具有相同的度序列，即 (2,2,2,2,2, 2, 2, 2) $=(2^8)$。

完全由其度序列确定的图称为单图。当 G是单图且 $D_G = D_{G'}$时，那么 G 和 G' 同构。单图的一些例子包括完全图、mK_2（m 个 K_2的副本）、C_5 及其补图。单图看起来似乎很特殊，因此数量不多。但事实上，具有 n个顶点 [17]的（无标号）单图有 2.3^{n-2} 到 2.6^n 个!

关于单图最重要的论文之一，是 Regina Tyshkevich 于 2000 年发表在《离散数学》期刊上的《图序列和单图的分解》[17]。自 20 世纪 70 年代末以来，Tyshkevich 一直与几位合著者合作研究单图（参见 [17] 中的参考文献 15 至 23）。这些论文以俄语撰写，在西方鲜为人知。《分解》论文的发表，终于让更多人了解了 Tyshkevich 的工作。

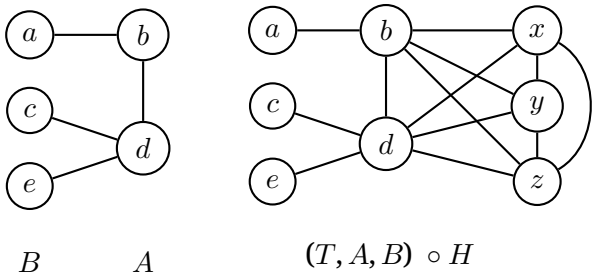


图1：左侧的树 T 的度序列为 (3, 2, 1, 1, 1) $= (3, 2, 1^3)$。它有一个 KS- 分割 $(\{b, d\}, \{a, c, e\})$。它与另一个图 H（一个 3-圈）通过复合运算组合，得到图 $(T, A, B) \circ H$。

Tyshkevich 的论文分为两部分：（i）一条分解定理，描述如何将每个图拆解为一列基本分量；（ii）对所有基本单图分量的完整分类。两者结合，揭示了每个单图是如何生成的。这篇文章旨在非正式地概述 Tyshkevich 的结果，并展示如何利用这些结果计算各种图参数。

这些算法是以下思路的一种变体：要计算参数 $f(G)$，先确定每个基本分支 G_i 的 $f(G_i)$，再将它们组合起来以得到 $f(G)$。在第 4 节中，我们介绍 Whitman [18] 的一项结果，并将其推广，以证明当 G 是单图时，团数 $\omega(G)$、独立数 $\alpha(G)$、顶点覆盖数 $\beta(G)$ 和色数 $\chi(G)$ 均可在线性时间内计算。在第 5 节中，我们将说明如何利用 Tyshkevich 分解定理的一个修改版本来刻画图的自同构群。因此，当 G 是单图时，G 的区分数和固定数（这两个参数与图的对称性破缺有关）也可在线性时间内计算。我们还配备了一个工具包，使读者更容易理解相关内容。该工具包将在第 6 节中介绍。

我们注意到，其他研究人员也研究了单图上的算法问题。Calam- oneri 和 Petreschi [5] 使用上述模板设计了一种近似算法，用于计算单图的 $L(2, 1)$-标记。所使用的颜色数至多为 $1.5 \times OPT$。Nakahata [16]利用相同的思想证明了单图的团宽度至多为 4。第 4 节和第 5 节中的算法之所以值得关注，是因为它们能够在线性时间内输出精确答案。

2 分解定理

分裂图在 Tyshkevich 分解定理中发挥着重要作用。如果一个图 G 的顶点集可以划分为两个集合，使得第一组顶点诱导一个团，而第二组顶点诱导一个稳定集或独立集，则称该图为分裂图。这两个集合构成所谓的 KS-划分，其中 G 是该图。K-部分表示团，S-部分表示稳定集。

考虑图1中的树。它有一个 KS-划分（$\{b, d\}$，$\{a, c, e\}$），因此它是一个分裂图。事实上，这是这棵树唯一的 KS-划分。另一方面，完全图 K_n 有两种

KS-划分类型：第一种将整个顶点集放在 K部分，并将空集作为 S部分；第二种将除一个顶点外的所有顶点放在 K部分，并将一个顶点放在 S部分。Hammer 和 Simeone [13] 描述了分裂图的 KS-划分。

定理2.1。 [13] 对于分裂图 G 的任意 KS-划分 (A, B)，以下情况恰有一种成立：

- $|A| = \omega(G)$ 和 $|B| = \alpha(G)$
- $|A| = \omega(G)$ 和 $|B| = \alpha(G) - 1$ （称为 K-最大）
- $|A| = \omega(G) - 1$ 和 $|B| = \alpha(G)$（称为 S-最大）

其中 $\omega(G)$ 和 $\alpha(G)$ 分别是 G 的团数和独立数。此外，在一个K-最大划分中，存在一个顶点 $u \in A$，使得 $B \cup \{u\}$ 诱导出一个稳定集；而在一个 S-最大划分中，存在一个顶点 $v \in B$，使得 $A \cup \{v\}$ 诱导出一个完全图。

具有第一类 KS-划分的分裂图称为平衡的；否则称为非平衡的。事实证明，平衡分裂图只有一个 KS-划分 [9], ，而非平衡分裂图则有一个 K-最大划分和一个 S-最大划分，这是因为存在顶点u 和 v，它们称为摆动顶点。因此，图1中的树是一个平衡分裂图，而K_n 则是一个非平衡分裂图。

命题2.2. [9] 平衡分裂图 G 有唯一的 KS-划分。

对于分裂图 G，我们通常需要明确所关注的 KS-划分，因此将该图表示为 (G, A, B)，其中 A 是 K-部分， B 是 S-部分，二者构成 KS-划分。(G, A, B)的度序列也表示为配对度序列 $(d_1, \ldots, d_r; d_{r+1}, \ldots, d_n)$，以区分 A 和 B 中顶点的度。

* * * * *

设 (G, A, B) 是一个分裂图。设 H 是另一个图。Tyshkevich 将 (G, A, B) 与 H 的复合定义为图 $(G, A, B) \circ H$。其顶点集为 $V(G) \cup V(H)$，边集为$E(G) \cup E(H) \cup \{uv | u \in A, v \in V(H)\}$。也就是说，$(G, A, B) \circ H$ 是图 $G \cup H$ 与 A 中的顶点和 H中的顶点之间的边所构成的图。图1给出了一个例子。

复合运算可应用于两个分裂图（G', A', B'）$\circ$（G, A, B），从而得到另一个分裂图（G'', $A \cup A', B \cup B'$）。它也可以多次应用以生成新的图；例如，（G_k, A_k, B_k）$\circ$（$G_{k-1}, A_{k-1}, B_{k-1}$）$\circ \ldots \circ$（$G_1, A_1, B_1$）$\circ G_0$。有趣的是，复合运算可以按任意顺序进行，因为该运算满足结合律。

一种重要的图类可以用这种方式描述，即阈值图。当每个 G_i（包括 G_0）都是单顶点图时，所形成的图就是阈值图。为简便起见，我们将单顶点图记为 $\{v\}$。将其视为分裂图且顶点位于 K 部分时，我们记作 $(\{v\}, \{v\}, \emptyset)$；其配对度序列为 $(0; \emptyset)$。另一方面，当顶点位于 S 部分时，我们记作 $(\{v\}, \emptyset, \{v\})$，其度序列为 $(\emptyset, 0)$。例如，

$$(\{v_4\}, \{v_4\}, \emptyset) \circ (\{v_3\}, \{v_3\}, \emptyset) \circ (\{v_2\}, \emptyset, \{v_2\}) \circ (\{v_1\}, \emptyset, \{v_1\}) \circ \{v_0\}$$

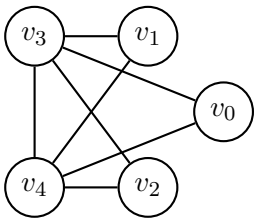


图2：阈值图 $(\{v_4\},\{v_4\},\emptyset)\circ(\{v_3\},\{v_3\},\emptyset)\circ(\{v_2\},\emptyset,\{v_2\})\circ(\{v_1\},\emptyset,\{v_1\})\circ\{v_0\}$。

得到图2所示的图。

* * * * *

Tyshkevich 将图定义为：若一个图可以通过某个分裂图与另一个图复合得到，则称它为可分解图；否则，它就是不可分解图。

如何判断一个图是否可分解？答案原来藏在图的度序列中！我们再来看看复合运算。在 $(G, A, B)\circ H$ 中，A中的所有顶点都与 H中的所有顶点相邻。因此，

- 当 $u\in A$, $deg(u)\geq |V(H)|+|A|-1$ 时，因为 u 也与 A中的其他所有顶点相邻；
- 当 $v\in V(H)$, $|A|\leq deg(v)\leq |A|+|V(H)|-1$ 时，因为 v 在 H中的邻居数量介于 0 和 $|V(H)|-1$之间；
- 当 $w\in B$, $0\leq deg(w)\leq |A|$ 时，因为 w 在 G 中的邻居数量介于 0 和 $|A|$之间。

由此可知，在 $(G, A, B)\circ H$ 的度序列中，A 中顶点的度排在最前面，其后是 H 中顶点的度，最后是 B 中顶点的度。换言之，如果 $(G, A, B)\circ H$ 的度序列为 $(d_1, d_2, \ldots, d_N)$，那么存在两个指标 p 和 q，使得 $p+q<N$，从而

$$(d_1,\ldots,d_p),(d_{p+1},\ldots,d_{N-q}) \text{ and } (d_{N-q+1},\ldots,d_N)$$

分别是 A、H 和 B 中顶点的度。

沿用上述记号，我们还注意到，在 $(G, A, B)\circ H$ 中，顶点 $u\in A$ 与 (i) A 中的 $p-1$ 个其他顶点、(ii) H 中的所有顶点，以及 (iii) B 中与其相邻的顶点相邻。因此，若 v_i 是度为 d_i 的顶点，且 $deg_B(v_i)$ 表示 v_i 在 B 中的邻居数，

$$\begin{aligned}\sum_{i=1}^{p} d_i &= \sum_{i=1}^{p}[(p-1)+(N-p-q)+deg_B(v_i)]\\ &= p(p-1)+p(N-p-q)+\sum_{i=1}^{p} deg_B(v_i)\\ &= p(N-q-1)+\sum_{i=N-q+1}^{N} d_i.\end{aligned}$$

Tyshkevich 指出，这些观察对于证明一个图是可分解的，既是必要条件，也是充分条件。

定理 2.3. [17] 一个度序列为（$d_1, d_2, \ldots, d_N$）的图 G^* 可分解，当且仅当存在指标 p, q，满足 $0 < p + q < N$，使得

$$\sum_{i=1}^{p} d_i = p(N - q - 1) + \sum_{i=N-q+1}^{N} d_i.$$

如果 G^* 可分解，那么 $G^* = (G, A, B) \circ H$，其中（$d_1, \ldots, d_p$）、（$d_{p+1}, \ldots, d_{N-q}$）和（$d_{N-q+1}, \ldots, d_N$）分别是 A、H 和 B 中顶点的度。

* * * * *

当 G^* 是多个图的复合时，由于复合运算满足结合律，将 G^* 表示为两个图的复合有多种方式。也就是说，有许多指标对（p, q）满足定理 2.3 中的条件。但是，如果选择 p 值最小的那一对，那么（G, A, B）必定是一个不可分解图。然后我们可以进一步分解 H。如此一来，我们就能将 G^* 分解为不可分解图。这引出了 Tyshkevich 的主要结果。

定理 2.4。（Tyshkevich规范分解定理 [17]）每个图G^* 都可以表示为一个复合

$$G^* = (G_k, A_k, B_k) \circ (G_{k-1}, A_{k-1}, B_{k-1}) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$$

其中每个 G_i 都是不可分解图。该分解是规范的；也就是说，设

$$G' = (G'_\ell, A'_\ell, B'_\ell) \circ (G'_{\ell-1}, A'_{\ell-1}, B'_{k-1}) \circ \ldots \circ (G'_1, A'_1, B'_1) \circ G'_0.$$

那么 $G^* \cong G'$ 当且仅当 $k = \ell$、$G_0 \cong G'_0$，并且 $(G_i, A_i, B_i) \cong (G'_i, A'_i, B'_i)$ 对于 $i = 1, \ldots, k$。

利用定理 2.4，我们提出了一种名为 Decompose(G^*) 的线性时间算法 [8]，该算法输出一个栈，栈顶包含 G_0 的（配对）度序列，其后依次是 G_1、……、G_k的配对度序列。

3 不可分解单图

在她的论文第二部分中，Tyshkevich指出了定理 2.4 对单图的意义。

定理 3.1. [17] G 是单图，当且仅当 G 的规范分解中的每个不可分解分量也都是单图。

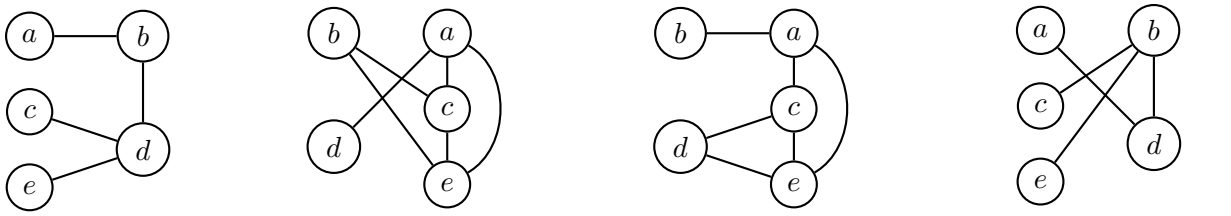


图3：左侧给出了树 T及其 $A = \{b, d\}$ 和 $B = \{a, c, e\}$，右侧展示了 (T, A, B) 的补图、逆以及补图的逆。

同样重要的是，Tyshkevich 描述了所有不可分解单图。证明相当冗长，长达二十页，需要梳理许多情形。Tyshkevich 之所以能将不可分解单图的清单控制在合理范围内，是因为她利用了图的补图和逆图。

回顾一下，$G, \overline{G}$ 的补图与 G 具有相同的顶点集，并且当且仅当两个顶点在 G中不相邻时，它们在补图中相邻。当 G 是一个具有 KS-划分 (A, B) 的分裂图时，(G, A, B) 的补图为 $(G, A, B) = (\overline{G}, B, A)$。其逆为 $(G, A, B)^I = (G^I, B, A)$，其中 G^I由 G 删除边 $\{aa' : a, a' \in A\}$ 并添加边 $\{bb' : b, b' \in B\}$ 得到。随后可以将这两种操作结合起来，得到 (G, A, B) 的逆的补图；它也等于 (G, A, B) 的补图的逆。图3展示了这两种图操作。

不可分解非分裂单图可以用四种类型的图来描述。前两种是 C_5 和 mK_2。第三种是 $U_2(m, \ell)$，即由 mK_2 与星图 $K_{1,\ell}$的不交并构成的图。第四种是 $U_3(m)$，即由中心顶点 mK_2 和 P_3构成的图。中心顶点与 mK_2的所有顶点相邻，形成 m 个三角形；它还与P_3的端点相邻，从而形成一个四圈。这些非分裂单图见表 1。

定理 3.2。 [17] 设 G是一个顶点数不少于 2 的不可分解非分裂单图。那么， G 或 $\overline{G}$ 是以下图之一： C_5、带有 $m \geq 2$的 mK_2 、带有 $m \geq 1$的 $U_2(m, \ell)$、 $\ell \geq 2$ ，或带有 $m \geq 1$的 $U_3(m)$。

另一方面，不可分解分裂单图可用五种类型的图来描述。第一种是单顶点图。第二种是 $S(p, q)$，即由 q 个星图 $K_{1,p}$ 组成的图，其 q 中心构成一个完全图。第三种是 $S_2(p_1, q_1, p_2, q_2, \cdots, p_m, q_m)$，它是 $S(p, q)$ 的推广；它由 q_i 个 K_{1,p_i} 组成，其中 $i = 1, \ldots, m$，且所有星图的中心构成一个完全图。第四种图是 $S_3(p, q_1, q_2)$，由 $S(p, q_1)$ 和 $S(p + 1, q_2)$ 组成，并且它们所有星图的中心两两相邻。此外，还有一个顶点 e，它仅与 $S(p, q_i)$ 中星图的中心相邻。最后一种是 $S_4(p, q)$，由 $S_3(p, 2, q)$ 以及一个顶点 f组成；该顶点与 $S_3(p, 2, q)$ 中除 e 之外的所有顶点相邻。这些分裂单图见表2。

定理 3.3。 [1 7] 设 G 为一个不可分解分裂单图。那么 G、 $\overline{G}$、 G^I 或 $\overline{G^I}$ 是以下 e

之一：一个单顶点图， $S(p,q)$（其中 $p \geq 1$、 $q \geq 2$、 S_2），$p_1, q_1, p_2, q_2, \cdots, p_m, q_m$（其中 $m \geq 2$和 $p_1 > p_2 > \ldots > p_m$）， $S_3(p, q_1, q_2)$（其中 $p \geq 1$、 $q_1 \geq 2$、 $q_2 \geq 1$），或 $S_4(p,q)$（其中 $p \geq 1$、 $q \geq 1$）。

当 $\overline{G} = H$ 时， $G = \overline{H}$。类似地，当 $G^I = H$ 时， $G = H^I$。因此，定理 3.2 和定理 3.3 还可以表述为：当 G 是不可分解非分裂单图时， G 同构于表 1 中四种类型的图之一或它们的补图。另一方面，当 G 是不可分解分裂图时， G 同构于表 5 中五种类型的图之一、它们的补图、逆图或逆图的补图。因此，只要 G 是不可分解单图，我们就可以称其类型（以及相关参数）。例如， C_4 同构于 $\overline{2K_2}$，因此其类型是 mK_2 的补图，其中 $m = 2$。另一方面，图 1 中树的类型是 $S_2(p_1, q_1, p_2, q_2)$，其中 $p_1 = 2, q_1 = p_2 = q_2 = 1$。当 G 是单顶点分裂图时，如果该顶点分别位于分裂图的 K 部分或 S 部分，我们就称其类型为 K_1 或 S_1。

定理 3.1 至定理 3.3 有一个微妙但重要的推论：单图及其不可分解分量可以在线性时间内识别。高效识别单图并非新成果；Kleitman 和 Li [15] 以及 Borri 等人 [3] 已经设计出了线性时间算法。新颖之处在于，它们的不可分解分量也能被快速识别。

定理 3.4. 设 G 为一个图。存在一种线性时间算法，可以判定 G 是否为单图。此外，当 G是单图时，该算法会输出 G每个不可分解分量的类型（及相关参数）。

该定理成立，是因为表1和表2中的图，其度序列都具有非常明确的模式。我们在附录中完整介绍名为 IsUnigraph(G) 的线性时间算法。该算法基于我们在 [8], 中的工作，这项工作与 Calamoneri 和 Peterschi [5]的工作类似。

Tyskevich 的结果也可用于生成单图。在她的研究之前，人们并不清楚该如何创建，例如一百个并非来自同一子类（如阈值图）的单图。如今我们知道了，因为她已经确定了用于组装单图的基本部件。下面展示的是具有十个顶点的单图。有些本身就是不可分解的，或者由一个不可分解单图与任意数量的不可分解分裂单图复合而成。

- 8^{10} 来自 $\overline{5K_2}$
- (3$_1$,1$_9$) 来自 $U_2(3, 3)$
- (8$_4$,5$_4$, 2$_2$) 来自 $S(2, 2)^I \circ 2K_2$
- (8$_4$,7$_1$, 6$_2$, 3$_3$) 来自 $\overline{S_3(1, 2, 1)} \circ S_1 \circ S_1$
- (7$_4$,6$_1$, 5$_1$, 3$_3$, 0$_1$) 来自 $S_1 \circ \overline{S_3(1, 2, 1)} \circ S_1$
- (9$_1$、7$_1$、6$_1$、4$_5$、1$_2$) 来自 $K_1 \circ S_1 \circ S_1 \circ K_1 \circ U_3(1)$

上述图的绘图和邻接列表可以使用哈维尔-哈基米算法 [14, 12] 获得。

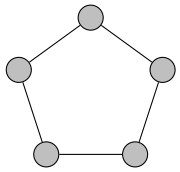
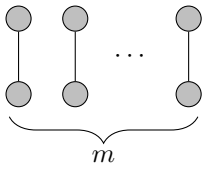
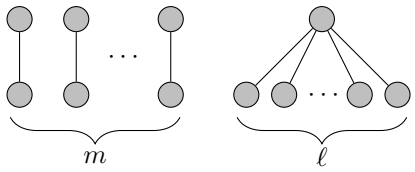
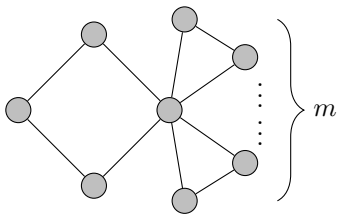
不可分解 非分裂单图 G	示意图和度序列	$\omega(G)$	$\alpha(G)$
C_5	 (2^5)	2	2
mK_2 及 $m \geq 2$	 (1^{2m})	2	m
$U_2(m, \ell) = mK_2 \cup K_{1,\ell}$ 其中 $m \geq 1, \ell \geq 2$	 $(\ell, 1^{2m+\ell})$	2	$m+\ell$
$U_3(m)$，$m \geq 1$	 $(2m+2, 2^{2m+3})$	3	$m+2$

表 1：定理 3.2 中提到的四种不可分解非分裂单图类型、它们的度序列、团数和独立数。

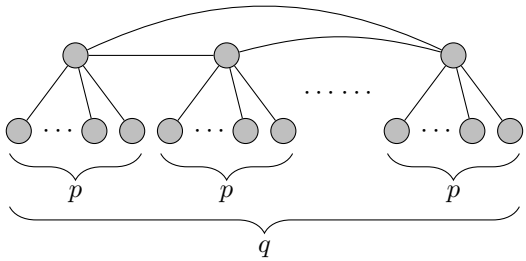
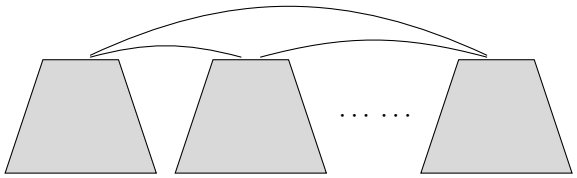
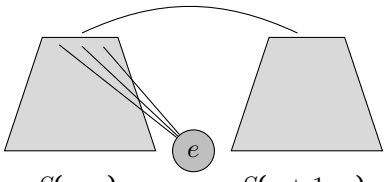
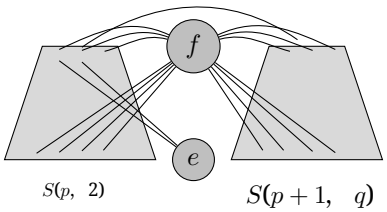
不可分解分裂单图 G	示意图与配对度序列
单顶点图	 (0; ∅) 或 (∅; 0)
$S(p,q)$, $p \geq 1$, $q \geq 2$	 $((p+q-1)^q; 1^{pq})$
$S_2(p_1,q_1,\ldots,p_m,q_m)$, $m \geq 2$, $p_1 > p_2 > \ldots > p_m \geq 1$ 和 $q_i \geq 1$ 对于 $i = 1,\ldots,m$	 $S(p_1,q_1)$ $S(p_2,q_2)$ $S(p_m,q_m)$ $((p_1+N-1)^{q_1},\cdots,(p_m+N-1)^{q_m};\ 1^{p_1q_1+\cdots+p_mq_m})$ 其中 $N=\sum_{i=1}^{m} q_i$
$S_3(p,q_1,q_2)$, 其中 $p \geq 1$、$q_1 \geq 2$、$q_2 \geq 1$	 $S(p,q_1)$ $S(p+1,q_2)$ $((p+q_1+q_2)^{q_1+q_2};\ q_1,\ 1^{pq_1+(p+1)q_2})$
$S_4(p,\ q)$, 其中有 $p \geq 1$、$q \geq 1$	 $S(p,\ 2)$ $S(p+1,\ q)$ $(2(p+q+1)+qp, (p+q+3)^{q+2}; 2^{qp+2p+q+1})$

表2：定理 3.3 中提到的五种不可分解的非分裂图及其度序列。分裂图的 K-部分位于上方，而 S-部分位于下方。

4 应用，第 1 部分

现在我们来展示如何利用 Tyshkevich 的工作解决算法问题。众所周知，给定一个图 G，计算其最大团 $\omega(G)$ 的大小、最大稳定集 $\alpha(G)$ 的大小、最小顶点覆盖 $\beta(G)$ 的大小，以及其色数 $\chi(G)$，都是 NP困难问题。当 G 是可分解的时，分解能否简化这些参数的计算？我们先介绍 Whitman [18] 的一个优美结果。

引理 4.1。 [18] 设 $G = (G_k, A_k, B_k) \circ (G_{k-1}, A_{k-1}, B_{k-1}) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$。那么 $\omega(G) = |A_k| + |A_{k-1}| + \ldots + |A_1| + \omega(G_0)$ 且 $\chi(G) = |A_k| + |A_{k-1}| + \ldots + |A_1| + \chi(G_0)$。

对于 $\alpha(G)$和 $\beta(G)$，也有类似的结果。

引理4.2。设 $G = (G_k, A_k, B_k) \circ (G_{k-1}, A_{k-1}, B_{k-1}) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$。则 $\alpha(G) = |B_k| + |B_{k-1}| + \ldots + |B_1| + \alpha(G_0)$且 $\beta(G) = |A_k| + |A_{k-1}| + \ldots + |A_1| + \beta(G_0)$。

证明。首先，我们对 G的分量数使用归纳法，证明关于 $\alpha(G)$的公式。当 G 只有一个分量时，$G = G_0$ 因此 $\alpha(G) = \alpha(G_0)$。假设对于具有 k 个或更少分量的图，该结果成立。令 G 为一个具有 $k+1$ 个分量的图，因此 $G = (G_k, A_k, B_k) \circ (G_{k-1}, A_{k-1}, B_{k-1}) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$。令 $H = (G_{k-1}, A_{k-1}, B_{k-1}) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$，因此 $G = (G_k, A_k, B_k) \circ H$。记 B_H为 H中的一个最大稳定集。显然，$B_k \cup B_H$是 G的一个稳定集，因此 $\alpha(G) \geq |B_k| + |B_H|$。现在我们将证明 $\alpha(G) \leq |B_k| + |B_H|$。

设 D 是 G、$D_k = D \cap V(G_k)$ 和 $D_H = D \cap V(H)$ 的稳定集。注意，D_k 和 D_H 必须分别是 G_k 和 H 的稳定集。此外，D_k 在 A_k 中至多包含一个顶点，因为 A_k 中的顶点诱导出一个团。我们考虑两种情况：（i）$|D_k \cap A_k| = 0$；（ii）$|D_k \cap A_k| = 1$。对于情况（i），$|D| = |D_k| + |D_H| \leq |B_k| + |B_H|$。对于情况（ii），由于 $D_k \cap A_k \neq \emptyset$，$D_H$ 必须为空，因为 A_k 中的每个顶点都与 G 中的 H 的每个顶点相邻。因此，$|D| = |D_k| \leq |B_k| + 1 \leq |B_k| + |B_H|$。后一个不等式成立，是因为 H 是一个非空图，所以 $|B_H| \geq 1$。至此，我们已证明，在情况（i）和（ii）下，G 的每个稳定集的大小至多为 $|B_k| + |B_H|$。由此可得 $\alpha(G) = |B_k| + |B_H|$。但根据归纳假设，$|B_H| = |B_{k-1}| + \ldots + |B_1| + \alpha(G_0)$，因此 $\alpha(G) = |B_k| + |B_{k-1}| + \ldots + |B_1| + \alpha(G_0)$。

现在，$\alpha(G) + \beta(G) = |V(G)|$ 和 $|V(G)| = |V(G_k)| + |V(G_{k-1})| + \ldots + |V(G_0)|$。Thu s,

$$\begin{aligned}
\beta(G) &= |V(G)| - \alpha(G) \\
&= (|V(G_k)| + |V(G_{k-1})| + \ldots + |V(G_1)| + |V(G_0)|) - (|B_k| + |B_{k-1}| + \ldots + |B_1| + \alpha(G_0)) \\
&= (|V(G_k)| - |B_k|) + (|V(G_{k-1})| - |B_{k-1}|) + \ldots + (|V(G_1)| - |B_1|) + (|V(G_0)| - \alpha(G_0)) \\
&= |A_k| + |A_{k-1}| + \ldots + |A_1| + \beta(G_0)
\end{aligned}$$

其中最后一个等号用到了这一事实：$\alpha(G_0) + \beta(G_0) = |V(G_0)|$。 □

我们注意到，上述引理并不要求将 G 表示为其规范分解。G 的某个分解即可。我们还注意到，几乎总是有 $\omega(G) = \sum_{i=0}^{k} \omega(G_i)$ 或 $\alpha(G) = \sum_{i=0}^{k} \alpha(G_i)$，等等。当 G 的分裂图分量不是平衡的时，就会出现复杂情况。可能会有 $\omega(G_i) = |A_i| + 1$ 或 $\alpha(G_i) = |B_i| + 1$

因此，我们不能简单地在 $\omega(G)$ 和 $\alpha(G)$ 的公式中，分别用 $\omega(G_i)$ 替换 $|A_i|$ ，以及用 $\alpha(B_i)$ 替换 $|B_i|$ 。

* * * * *

然而，为了实现引理 4.1 和引理 4.2，我们将使用 Decompose(G) 将 G 分解为其不可分解分量。它将在 $i \geq 1$ 线性时间内输出每个 (G_i, A_i, B_i) 的配对度序列。因此，也可以在线性时间内得到 $|A_i|$ 和 $|B_i|$ （针对 $i \geq 1$ ）。这意味着，我们只需计算 $\omega(G)$ $\chi(G)$ $\alpha(G)$ 和 $\beta(G)$，分别使用 $\omega(G_0)$、 $\chi(G_0)$、 $\alpha(G_0)$ 和 $\beta(G_0)$ 即可。对于单图，Tyshkevich 已经完成了艰难的工作，找出了 G_0的所有可能候选项。

在证明中，我们将用到下面这个命题。

命题 4.3。当 G_i 是一个含有两个或更多顶点的不可分解分裂图时， G_i 是平衡的。也就是说， G_i 有唯一的 KS-划分 (A_i, B_i)，使得 $\omega(G_i) = |A_i|$ 且 $\alpha(G_i) = |B_i|$。

定理 4.4. 当 G 是单图时，参数 $\omega(G)$ $\chi(G)$、 $\alpha(G)$ 和 $\beta(G)$ 均可在线性时间内计算。

证明。设 G 的规范分解为 $(G_k, A_k, B_k) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$. 如上所述，Decompose($G$) 将输出每个 G_i, $i \geq 1$ 的配对度序列，因此 $|A_i|$ 和 $|B_i|$ 对于$i = 1, \ldots, k$ 均可在线性时间内求得。首先来确定 $\omega(G_0)$ 和 $\alpha(G_0)$。

假设 G_0 是分裂图。当 G_0 是单顶点图时， $\omega(G_0) = \alpha(G_0) = 1$。当 G_0有两个或更多顶点时，根据命题 4.3， G_0 有唯一的 KS-划分 (A_0, B_0)，其中$\omega(G_0) = |A_0|$ 且 $\alpha(G_0) = |B_0|$。现在，Decompose(G) 也将输出 G_0的度序列。使用 DetermineSplit [8], ——一种基于 Hammer 和 Simeone 的定理、能够识别图是否为分裂图的算法—— [13]，(G_0, A_0, B_0) 的配对度序列，因此 $|A_0|$ 和$|B_0|$也可在线性时间内求得。

当 G_0 是非分裂图时， G_0 同构于表 1 所列四种图类型之一或其补图。表格第三列和第四列展示了 C_5、 mK_2、 U_2（m、 ℓ）和 U_3（m）的团数和独立数。很容易验证这些值是正确的。此外，由于 $\omega(\overline{G}) = \alpha(G)$，而 $\alpha(\overline{G}) = \omega(G)$，因此可以根据这些值求出其补图的团数和独立数。因此，要确定 $\omega(G_0)$ 和 $\alpha(G_0)$，请使用 IsUnigraph(G_0)识别 G_0 的类型，然后根据表 1 中的信息求出相应的团数和独立数。

我们已经证明， $\omega(G)$ 和 $\alpha(G)$ 可以在线性时间内计算。由于 $\beta(G) = |V(G)| - \alpha(G)$，因此 $\beta(G)$ 也可以在线性时间内计算。最后，Tyskevich 在 [17] 的推论 11 中证明，单图 G 是完美图，当且仅当 G_0 不是 5-圈。回顾一下，完美图的色数等于其团数。因此，当 G_0 不是 5-圈时，$\chi(G) = \omega(G)$。当 G_0 是 5- 圈时，根据引理 4.1， $\chi(G) = \omega(G) + 1$，因为 $\chi(C_5) = 3$，但 $\omega(C_5) = 2$。由于 $\omega(G)$ 可以在线性时间内计算，因此 $\chi(G)$ 也可以。 □

5 应用，第2部分

我们最初对 Tyshkevich 分解定理产生兴趣，源于我们关于图中对称性破缺的研究 [6, 2, 7, 8]。图 G 的对称性是对 G 的自同构的一种非正式说法，即一个双射 $\pi: V(G) \to V(G)$，它保持 G 的邻接关系。例如， C_5 有十个自同构：五种旋转和五种反射。若自同构是恒等映射，我们称其为平凡自同构；否则称为非平凡自同构。 G 的自同构集合 $Aut(G)$ 在函数复合运算下构成一个群，因此称为 G 的自同构群。

假设 G 的规范分解为 $(G_k, A_k, B_k) \circ \ldots \circ (G_1, A_1, B_1) \circ G_0$。我们想要了解 G 的自同构如何作用于这些 G_i。是否可能有 $Aut(G)$ 同构于 $Aut(G_k) \times Aut(G_{k-1}) \times \ldots \times Aut(G_1) \times Aut(G_0)$? 也就是说， G 的每个自同构π能否分解为函数 $(\pi_k, \pi_{k-1}, \ldots, \pi_1, \pi_0)$，使得每个 π_i 都是 G_i 的自同构；反过来， G_i、 $i = 0$、.....、 k 的自同构能否组合起来，构成 G 的一个自同构?

事实证明，上述猜想几乎成立。当 G 有两个或更多连续的不可分解分量，且它们都是类型相同的单顶点图时，猜想便不成立。来看图2中的阈值图，它的不可分解分量全都是单顶点图。设 G_i 表示包含 v_i 的分量。那么 G_1 和 G_2 的类型是 S_1 ，而 G_3 和 G_4 的类型是 K_1。事实上，函数 $\pi(v_0) = v_0$， $\pi(v_1) = v_2$， $\pi(v_2) = v_1$， $\pi(v_3) = v_4$， $\pi(v_4) = v_3$ 是该图的一个自同构。它“跨越”了不可分解分量，将 G_1 和 G_2 的顶点彼此映射，并将 G_3 和 G_4 的顶点彼此映射。

为了解决这一问题，我们在 [8] 中提出了 Tyshkevich 分解定理的紧凑版本。在 G 的分解中，当 G_i $i \geq 1$ 是单顶点图时，它要么具有 K_1 类型，要么具有 S_1 类型。此外，当 G_0 和 G_1 都是单顶点图时，我们称 G_0 的类型与 G_1 的类型相同。

G 的紧致规范分解是通过将 G 的规范分解中类型相同的连续单顶点分支尽可能合并为一个分支而形成的。也就是说，如果存在 m 个类型为 K_1 的连续分支，则将它们替换为具有 m 个顶点、其 KS-划分为 K-max 的完全图（即，度序列为 $((m-1)^m; \emptyset)$ ）。另一方面，如果它们的类型为 S_1，则将它们替换为具有 m 个孤立顶点且其 KS-划分为 S-max 的图（即，度序列为 $(\emptyset; 0^m)$ ）。

例如，如果我们用度序列描述 G 的不可分解分量，即 $G = (\emptyset; 0) \circ (0; \emptyset) \circ (0; \emptyset) \circ (0; \emptyset) \circ (\emptyset; 0) \circ (\emptyset; 0) \circ (\emptyset; 0) \circ (0)$，那么它的紧致规范分解为 $G = (\emptyset; 0) \circ (2^3; \emptyset) \circ (\emptyset; 0^4)$。另一方面，图2中阈值图的规范分解为 $(0; \emptyset) \circ (0; \emptyset) \circ (\emptyset; 0) \circ (\emptyset; 0) \circ (0)$，因此它的紧致规范分解为 $(1^2; \emptyset) \circ (\emptyset; 0^3)$。

定理 5.1。（紧致规范分解定理 [8]）图 G 的紧致规范分解，是由 Tyshkevich 对 G 的规范分解中，将类型相同的单顶点分量尽可能合并为一个分量而得到的。 G 的紧致分解在同构意义下唯一。每个分量都是至少含有两个顶点的不可分解图、完全图或孤立顶点图。

现在我们可以刻画 G 的自同构。

定理 5.2。设 $G = (G'_\ell, A'_\ell, B'_\ell) \circ (G'_{\ell-1}, A'_{\ell-1}, B'_{\ell-1}) \circ \ldots \circ (G'_1, A'_1, B'_1) \circ G'_0$ 是 G的紧致规范分解。那么 $Aut(G)$ 同构于

$$Aut(G'_\ell) \times Aut(G'_{\ell-1}) \times \ldots \times Aut(G'_1) \times Aut(G'_0).$$

* * * * *

我们在 [8] 中的结果推出了定理 5.2，但为完整起见，我们在此给出证明。当 G 是一个分裂图，其 KS-划分为 (A, B) 时，令 $Aut(G, A, B)$ 包含 G 中固定 A 且也固定 B 的自同构；也就是说，这些自同构将 A 中的顶点映射到 A 中的顶点，并将 B 中的顶点映射到 B 中的顶点。我们将用到以下结果。

命题 5.3. [8] 当 G 是一个平衡分裂图，其 KS-划分为 (A, B) 时，$Aut(G) = Aut(G, A, B)$。

命题 5.4. [8]设 G 是一个有两个或更多顶点的图。
(i) 若 G 有一个孤立顶点 u，则 $G = (\{u\}, \emptyset, \{u\}) \circ (G - u)$。
(ii) 若 G 有一个占优的[1] 顶点 v，则 $G = (\{v\}, \{v\}, \emptyset) \circ (G - v)$。

引理 5.5。 [8] 令 $G^* = (G, A, B) \circ H$。存在一个自同构 π，它将 G^* 中的 $u \in V(G)$ 映射到 $v \in V(H)$，当且仅当以下条件之一成立：
(i) $u \in A$ 和 G 的一个摆动顶点，而 v 是 H 的一个支配顶点；或
(ii) $u \in B$ 和 G 的一个摆动顶点，而 v 是 H的一个孤立顶点。

引理 5.5 很重要，因为它描述了自同构 G^*跨越过去、将 G 中的一个顶点映射到 H中的一个顶点时的两种条件。现在让我们证明定理 5.2。

证明。我们将对 ℓ进行归纳。当 $\ell = 0, G = G'_0$ 时，$Aut(G) = Aut(G'_0)$ 成立。假设对于规范分解中含有 ℓ 个或更少分量的图，该定理成立。设 G 是一个规范分解中含有 $\ell + 1$ 个分量的图。因此，我们可以将 G 表示为 $G = (G'_\ell, A'_\ell, B'_\ell) \circ H$，其中 $H = (G'_{\ell-1}, A'_{\ell-1}, B'_{\ell-1}) \circ \cdots \circ (G'_1, A'_1, B'_1) \circ G'_0$。我们将论证 $Aut(G) = Aut(G'_\ell) \times Aut(H)$。

根据定理 5.1，G'_ℓ 有三种可能。对于每种情况，我们将论证引理 5.5 的两个条件均不成立。由此可知，G的每个自同构 π 都会将 $V(G'_\ell)$ 映射到 $V(G'_\ell)$，并将 $V(H)$ 映射到 $V(H)$。也就是说，$Aut(G) \subseteq Aut(G'_\ell) \times Aut(H)$。

当 G'_ℓ 是一个至少有两个顶点的不可分解分裂图时，根据命题 4.3，G'_ℓ 是一个平衡分裂图，因此没有摆动顶点。引理 5.5 中的条件 (i) 和 (ii) 均不成立。

[1]一个顶点是 $dominant$ 如果它与图中所有其他顶点都相邻 图。

反之，假设 G'_ℓ 是一个完全图，其 KS-划分为 $((m-1)^m;\emptyset)$。注意，引理 5.5 的条件 (ii) 不成立，因为 B 是空集，因此我们只需检查条件 (i)。 G'_ℓ 是一个类型为 K_1 的单顶点图，或者是通过将连续的类型为 K_1 的单顶点图尽可能合并而形成的。如果 $|H|=1$，那么 $H=G'_0$ 是一个单顶点图，本应与 G'_ℓ 一同合并到 G 的紧致规范分解中。因此， $|H|\geq 2$。如果 H 有一个支配顶点，那么根据命题 5.4， H 可分解，其最左侧分支的类型为 K_1。同样，这个类型为 K_1 的分支本应与 G'_ℓ 合并。因此， H 没有支配顶点。

最后，假设 G'_ℓ 是一个孤立顶点图，其 KS-划分为（$\emptyset$; $(m-1)^m$）。由于 A 是空集，引理 5.5 的条件 (i) 不成立。当 $|H|=1$、 $H=G'_0$ 和原本应该与 G'_ℓ合并。因此， $|H|\geq 2$。如果 H 有一个孤立顶点，那么根据命题 5.4， H 可分解，且最左侧分支的类型为 S_1。同样，最左侧分支原本应该与 G'_ℓ合并。因此， H 没有孤立顶点，所以引理 5.5 的条件 (ii) 不成立。

我们已经证明了 $Aut(G)\subseteq Aut(G'_\ell)\times Aut(H)$。下面证明其逆命题。设 $\pi_1\in Aut(G'_\ell)$ 且 $\pi_2\in Aut(H)$。在 $V(G)$ 上定义 π，使得当 $v\in V(G'_\ell)$ 时， $\pi(v)=\pi_1(v)$；当 $v\in V(H)$ 时， $\pi(v)=\pi_2(v)$。现在需要说明 π 保持了 G的邻接关系。

令 $x,y\in V(G'_\ell)$ 或 $x,y\in V(H)$。那么， xy是 G 的一条边，当且仅当 $\pi(x)\pi(y)$ 是 G 的一条边，因为 π 在 G'_ℓ 中的作用类似于 π_1，而在 H 中的作用类似于 π_2。因此，假设 $x\in V(G'_\ell)$ 且 $y\in V(H)$；当 $x\in V(H)$ 且 $y\in V(G'_\ell)$ 时的情形也可类似处理。设 xy 是 G 的一条边。由于 B'_ℓ 中没有顶点与 H 中的顶点相邻， $x\in A'_\ell$。如果我们能证明 $\pi(x)\in A'_\ell$，那么 $\pi(x)\pi(y)$ 也是 G 的一条边，因为 A'_ℓ 中的每个顶点都与 H 中的每个顶点相邻。

因为 $A'_\ell\neq\emptyset$, G'_ℓ 不可能是孤立顶点图。当 G'_ℓ 是一个至少有两个顶点的不可分解图时，(A'_ℓ, B'_ℓ) 是其唯一的 KS-划分。 G'_ℓ 的每个自同构都必须固定 A'_ℓ 和 B'_ℓ，因此 $\pi_1(x)\in A'_\ell$，因为 $x\in A'_\ell$。另一方面，当 G'_ℓ 是一个具有 KS-划分 $((m-1)^m;\emptyset)$ 的完全图时，其所有顶点都属于 A'_ℓ，所以 $\pi_1(x)\in A'_\ell$ 也成立。我们已经证明，当 $x\in A'_\ell$, $\pi(x)=\pi_1(x)\in A'_\ell$。

另一方面，假设 xy 不是 G的一条边。那么 $x\in B'_\ell$。如果我们能证明 $\pi(x)\in B'_\ell$也成立，那么 $\pi(x)\pi(y)$也不是 G的一条边。现在， G'_ℓ 不可能是完全图，因为 $B'_\ell\neq\emptyset$。因此， G'_ℓ 必定是一个至少有两个顶点的不可分解图，其唯一的KS-划分为(A'_ℓ, B'_ℓ)；或者 G'_ℓ 是一个孤立顶点图，其 KS-划分为$(\emptyset; (m-1)^m)$。在前一种情况下， π_1 固定 B'_ℓ，所以 $\pi_1(x)\in B'_\ell$。在后一种情况下， G'_ℓ 的所有顶点都属于 B'_ℓ，所以显然有 $\pi_1(x)\in B'_\ell$。因此， $\pi(x)=\pi_1(x)\in B'_\ell$。

我们已经证明， $Aut(G'_\ell)\times Aut(H)$中的每个自同构也是$Aut(G)$的一个自同构，因此 $Aut(G)=Aut(G'_\ell)\times Aut(H)$。根据归纳假设， $Aut(H)=Aut(G'_{\ell-1})\times\cdots\times Aut(G'_1)\times Aut(G'_0)$，因此 $Aut(G)=Aut(G'_\ell)\times Aut(G'_{\ell-1})\times\cdots\times Aut(G'_1)\times Aut(G'_0)$。 □

* * * * *

在图论中，对称性破缺的目标是区分图的各个顶点。如果存在一个将 u 映射到v的自同构，那么顶点 u 和 v 就可能被混淆。消除这种混淆的一种方法是给顶点着色，就像我们在图5中为 C_5所做的那样。

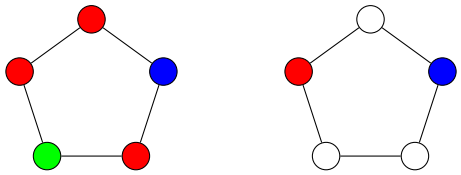


图 4：左侧是 C_5 的区分标记，右侧是 C_5 的固定集。属于固定集的顶点被赋予不同的颜色，而不属于该集合的顶点则被赋予空颜色。

可能彼此混淆的顶点是那三个红色顶点。但其中一个红色顶点的邻居分别是蓝色和绿色，另一个的邻居分别是红色和绿色，还有一个的邻居分别是红色和蓝色。因此，根据顶点的颜色及其邻居的颜色，就能识别出所有顶点。

在区分标记中，为图的顶点赋予颜色，使标记后的图只有一个自同构，即恒等映射。形式上，ϕ 个颜色的区分标记 G [1] 为 G 的顶点着色，使得对于 G 的每个非平凡自同构 π，存在某个顶点 v，使得 $\phi(v) \neq \phi(\pi(v))$。换句话说，$\phi$ 消除了 G 的所有非平凡自同构，使得标记后的图 (G, ϕ) 唯一的自同构是恒等映射。G 的区分数 $D(G)$ 是创建 G 的区分标记所需的最少颜色数。例如，$D(K_n) = n$，$D(P_n) = 2$ 对于 $n \geq 2$，而 $D(K_{n,\ n}) = n + 1$。

与区分标记相关的一个概念是固定集 [10, 4, 11]。$V(G)$的子集 S 是 G 的固定集，如果 G 中唯一固定 S 中每个元素的自同构是恒等映射。也就是说，每当 G 的一个自同构将每个 $s \in S$ 映射到自身时，它也将每个 $v \notin S$ 映射到自身。图5展示了 C_5的一个固定集。G的固定数 $Fix(G)$是最小固定集的大小。例如，对于 $n \geq 2$，$Fix(K_n) = n - 1$，而 $Fix(P_n) = 1$；$Fix(K_{n,n}) = 2(n - 1)$。我们注意到，$Fix(G) = 0$ 当且仅当 G 是刚性图（即 G 没有非平凡自同构）。

G 的每个固定集 S 都可以通过将 S中的每个顶点个体化，转化为一个区分标记 ϕ_S。S 中的每个顶点被赋予一种不同的颜色，而不在 S 中的顶点则被赋予另一种颜色（例如“空”颜色）。因此，$D(G) \leq Fix(G) + 1$。有时，$D(G) = Fix(G) + 1$，但 $D(G)$可以显著小于 $Fix(G)$，因为颜色可以在区分标记中重复使用，但在固定集中不能重复使用，空颜色除外。

我们提出的一个研究问题是，当 G 是单图时，能否高效地计算 $D(G)$。这一探究方向很有意思，因为单图有一种非常简单的同构算法：当 G 是单图而 H 是另一个图时，G 与 H 同构，当且仅当 G 与 H 的度序列完全相同。在关于区分标记 [6, 2, 7], 的研究中，我们发现，只要 G 属于一个具有高效同构算法的图族，就能在多项式时间内计算 $D(G)$。Tyshkevich 的结果对于解决这一问题至关重要。

定理 5.2 使对称性破缺变得简单明了。通俗地说，它意味着，如果我们想要破除 G 的非平凡自同构，只需破除 G 的紧致规范分解中各个组成部分的非平凡自同构。

定理 5.6。 [8] 令 $G = (G'_\ell, A'_\ell, B'_\ell) \circ (G'_{\ell-1}, A'_{\ell-1}, B'_{\ell-1}) \circ \ldots \circ (G'_1, A'_1, B'_1) \circ G'_0$ 为紧 t

G的规范分解。令 ϕ 为 G 的一个标记，并将 ϕ_i 记作当 ϕ限制于 $V(G_i')$时， G_i' 上的标记。那么，对于 $i=0$, 1, . . . , ℓ， ϕ 是 G 的一个区分标记，当且仅当 ϕ_i 是 G_i' 的一个区分标记。因此， $D(G) = \max\{D(G_i')$ $i=0,\ldots,\ell\}$。

为理解该定理为何成立，假设 ϕ_i 对于某个 G_i'不是区分标记。则 (G_i', ϕ_i) 具有一个非平凡自同构 π_i。当该自同构与 G_j'、 $j \neq i$的恒等映射结合时，所得结果便是 (G, ϕ) 的一个非平凡自同构。因此， ϕ对于 G不是区分标记。另一方面，假设 ϕ 不是 G 的区分标记，因此 (G, ϕ) 具有一个非平凡自同构 π。于是， π 可以表示为 $(\pi_\ell, \pi_{\ell-1}, \ldots, \pi_1, \pi_0)$，其中每个 π_i 都是 (G_i', ϕ_i) 的一个自同构。其中某一个，设为 π_j，是 (G_j', ϕ_j) 的非平凡自同构，因为 π是非平凡的。因此， ϕ_j 不是 G_j' 的区分标记。最后，用于为 G_i' 创建区分标记的颜色可以重新用于为 G_j'、 $j \neq i$创建区分标记。因此， $D(G) = \max\{D(G_i')$ $i=0$， . . . , $\ell\}$。

对固定集应用同样的推理，我们得到下一个结果。

定理 5.7. 设 $G = (G_\ell', A_\ell', B_\ell') \circ (G_{\ell-1}', A_{\ell-1}', B_{\ell-1}') \circ \ldots \circ (G_1', A_1', B_1') \circ G_0'$是 G的紧致规范分解。设 $S \subseteq V(G)$，且设 $S_i = S \cap V(G_i')$。那么， S是 G的固定集，当且仅当对于 $i=0, 1, \ldots, \ell$， S_i是 G_i' 的固定集。因此， $Fix(G) = \sum_{i=0}^{\ell} Fix(G_i')$。

* * * * *

为了计算单图的区分数和固定数，我们再次利用 Tyshkevich 对不可分解单图的刻画。关于单图的区分数可在线性时间内计算，读者可参阅 [8] 。

Th定理5.8. [8] 当 G 是单图时，可以在线性时间内计算 $D(G)$。

现在我们来证明一个关于单图固定数的类似结果。我们先给出一些基本结果，再利用它们确定表1和表2中各图的固定数。

命题5.9. 以下结论成立:

(i) $Fix(C_n) = 2$当 $n \geq 3$时， $Fix(P_n) = 1$当 $n \geq 2$时， $Fix(mK_2) = m$， $Fix(K_{1,\ell}) = \ell - 1$当 $\ell \geq 2$时，$Fix(K_n) = n - 1$。

(ii) 当 G 是刚性图时， $Fix(mG) = m - 1$。

(iii) 当 G 和 H 都是非刚性图时， $Fix(G \cup H) = Fix(G) + Fix(H)$。

引理5.10。表3中各图的固定数是正确的。

证明。根据命题5.9， C_5、 mK_2和 U_2（m、 ℓ）$= mK_2 \cup K_{1,\ell}$的固定数可直接得到。对于 U_3（m），中心顶点的度至少为4，而所有其他顶点的度均为2，因此 U_3（m）的每个自同构都必须将中心顶点映射到其自身。因此， U_3（m）的一个固定集实际上就是固定 $mK_2 \cup P_3$。根据命题5.9， Fix（$mK_2 \cup P_3$）$= Fix$（mK_2）$+ Fix$（P_3）$= m+1$，所以 Fix（U_3（m））$= m+1$。 □

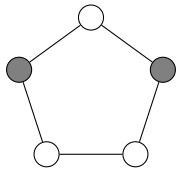
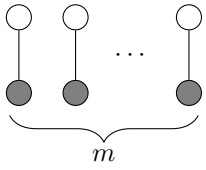
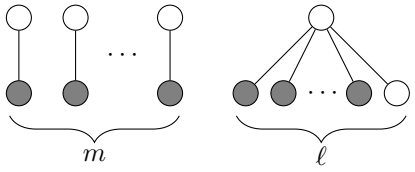
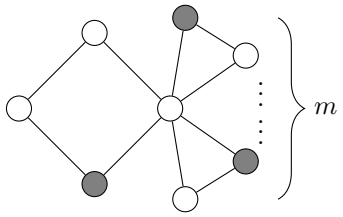
不可分解 非分裂单图 G	最小固定集	$Fix(G)$
C_5		2
mK_2 与 $m \geq 2$		m
$U_2(m,\ell) = mK_2 \cup K_{1,\ \ell}$ 与 $m \geq 1$、$\ell \geq 2$		$m+\ell-1$
U_3（m）、$m \geq 1$		$m+1$

表3：不可分解非分裂单图的四种类型、以灰色标出的最小固定集及其固定数。

引理5.11。表4中的图的固定数是正确的。

证明。令$K_{1,\ p}^*$为$K_{1,\ p}$，以其中心为根。[2] 每个 S（p、q）的自同构都会将$K_{1,\ p}$的一个中心映射到其自身或另一个中心。因此，我们可以简单地将 S（p、q）视为$qK_{1,\ p}^*$。当 $p=1$、$K_{1,\ p}^*$是刚性图时，根据命题5.9，Fix（$qK_{1,\ p}^*$）$=q-1$。但当 $p>1$、$K_{1,\ p}^*$仅仅是$K_{1,\ p}$时，它不再是刚性图。因此，根据命题5.9，Fix（$qK_{1,\ p}^*$）$=qFix$（$K_{1,\ p}$）$=q$（$p-1$）。

对于 $S_2(p_1,q_1,p_2,q_2,\cdots,p_m,q_m)$，只有当两个星图的叶节点数相同时，图的自同构才会将一个星图的中心映射到另一个星图的中心。因此，我们只需关注固定每个 $S(p_i,q_i)$，其中 $i=1$，.....，m。对于 $S_3(p,q_1,q_2)$，图的每个自同构都会将 e映射到自身，因此我们只需固定子图 $S(p,q_1)\cup S(p+1,q_2)$。最后，对于 $S_4(p,q)$，每个图的自同构都会固定 e 和 f，因此我们只需固定 $S(p,2)$ 和 $S(p+1,q)$。它们的固定数可由前文讨论得出。 □

[2]注意，当 $p\geq 2$时，Aut（$K_{1,\ p}^*$）$=Aut$（$K_{1,\ p}$）。因此，我们实际上只需要$K_{1,\ p}^*$当 $p=1$时。

现在我们已知表1和表2中不可分解单图的固定数，还需要考虑它们的补图、逆图以及其逆图的补图的固定数。

命题5.12. 设 G 和 H 是顶点集相同的两个图；即 $V(G) = V(H)$。此外， $Aut(G) = Aut(H)$。那么， G 的每个固定集都是 H 的固定集，反之亦然，因此$Fix(G) = Fix(H)$。

证明。假设 S 是 G 的固定集，但不是 H的固定集。那么存在某个 $\pi \in Aut(H)$，使得π 将 S 中的每个元素映射到其自身，但对于某个 $v \in \overline{S}$，有 $\pi(v) \neq v$。由于 $Aut(H) = Aut(G)$，可知 $\pi \in Aut(G)$，且在 G中的行为相同，这与 S 是 G的固定集这一事实矛盾。因此， S 是 H的固定集。同样的论证表明， H 的每个固定集也是 G的固定集。因此， $Fix(G) = Fix(H)$。 □

命题5.13. 对任意图 G， $Fix(G) = Fix(\overline{G})$。此外，当 G 是不可分解的分裂图时， $Fix(G) = Fix(G^I)$。

证明。对于任意图 G, $Aut(G) = Aut(\overline{G})$。在 [8], 的命题2.19中，我们还证明了，当 G是不可分解分裂图时， $Aut(G) = Aut(G^I)$。因此，根据命题5.12，对于任意图 G， $Fix(G) = Fix(\overline{G})$；对于任意不可分解分裂图 G， $Fix(G) = Fix(G^I)$。 □

下面给出关于单图固定数的定理。

定理5.14。当 G 是单图时， $Fix(G)$ 可以在线性时间内计算。

证明。定理 5.7给出了一种计算 $Fix(G)$ 的方法。首先，求出 $G = (G'_\ell, A_\ell, B_\ell) \circ \cdots \circ (G'_1, A'_1, B'_1) \circ G'_0$ 的紧致规范分解。接下来，对每个 G'_i，确定 $Fix(G'_i)$。最后，返回总和 $Fix(G'_\ell) + \ldots + Fix(G'_1) + Fix(G'_0)$。

可以轻易修改 Decompose(G) 和 IsUnigraph(G)，使其在线性时间内返回紧致规范分解中每个分支 G'_i 的（配对）度序列和类型 G。当 G'_0 是不可分解的非分裂图时，表3给出了 $Fix(G_0)$ 的答案，因为 G_0 同构于表中的某个图或其补图，而根据命题5.13，这些图具有相同的固定数。当 G'_i, $i \geq 0$ 是含有两个或更多顶点的不可分解分裂图时，出于同样的原因，可以在表4中找到 $Fix(G'_i)$。最后，当 G'_i 是完全图或孤立顶点图时， $Fix(G'_i) = |V(G'_i)| - 1$。显然，计算 $i = 0, \ldots, \ell$ 的 $Fix(G'_i)$，所需时间与 G的规模成线性关系。定理得证。 □

6 工具包

为了让我们的讨论更容易理解，我们创建了一个工具包 [3]，用于实现本文中描述的算法。这些工具可以完成以下工作：

[3] https://chelseal11.github.io/tyshkevich_分解_工具包/

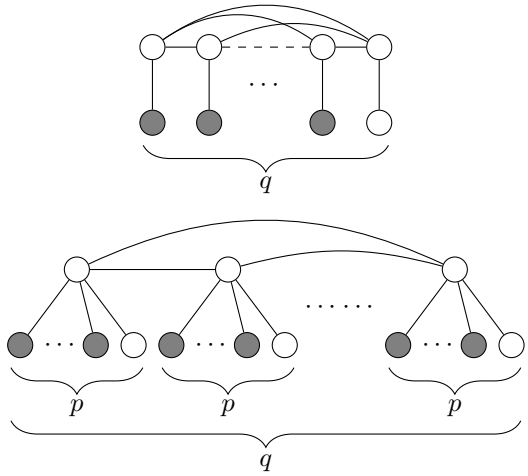
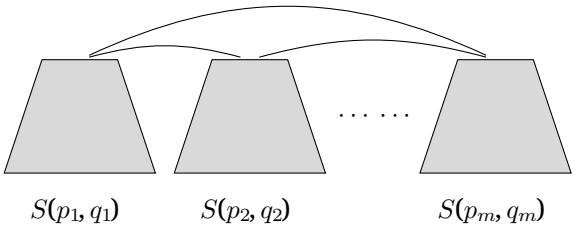
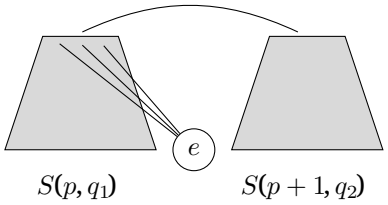
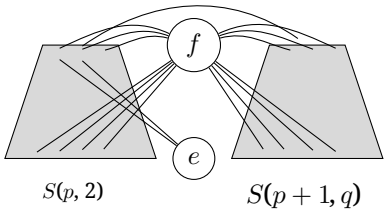
不可分解分裂 单图 G 与 $\lvert V(G)\rvert \geq 2$	最小固定集	$Fix(G)$
$S(p,q)$, $p \geq 1, q \geq 2$		$q-1$ 当 $p=1$; $q(p-1)$ 否则
$S_2(p_1,q_1,\ldots,p_m,q_m)$, $m \geq 2$ 和 $p_1 > p_2 > \ldots > p_m \geq 1$		$\sum_{i=1}^{m} Fix(S(p_i,q_i))$
$S_3(p,q_1,q_2)$, 其中 $p \geq 1, q_1 \geq 2, q_2 \geq 1$		$Fix(S(p,q_1))+$ $Fix(S(p+1,q_2))$
$S_4(p,q)$, 其中 $p \geq 1$、 $q \geq 1$		$Fix(S(p,2))+$ $Fix(S(p+1,q))$

表4：具有两个或更多顶点的不可分解的非分裂图的四种类型及其固定数。第二列中的阴影顶点也标示出应选入固定集的顶点。

- 给定一列图，它会输出这些图的复合。输入图和输出图均以其（配对）度序列描述。

- 给定一列不可分解单图，它会输出这些单图的复合。用户从下拉菜单中选择每个不可分解单图的类型，并指定相关参数。

- 给定正整数 n 和 k，它会生成具有 n 个顶点且包含 k 个不可分解分量的单图。

- 给定一个图的度序列，它会输出该图的规范分解。每个不可分解分量均以其（配对）度序列描述。它会判断输入图是否为单图；如果是，还会指出每个不可分解分量的类型。它还可以计算原始图的团数、色数、独立数和顶点覆盖数。

- 给定一个图的度序列，它会输出该图的紧致规范分解。同样，每个分量均以其度序列描述。如果输入图是单图，它还会输出每个分量的类型。此外，它还可以计算原始图的区分数和固定数。

7 结论

我们介绍了 Tyshkevich 的主要成果，并阐述了她的工作所蕴含的算法含义，展示了当输入图为单图时，如何在线性时间内计算各种图参数。很可能还可以为单图设计更多多项式时间算法。本文的成果也引出了一个问题：类似的方法能否适用于其他图类？也就是说，能否对它们的不可分解分量进行完整分类，并将其用于算法研究？我们希望，本文的阐述以及所开发的工具包，能够鼓励读者深入探讨这一主题，并推动更多研究。

参考文献

[1] M. O. Albertson 和 K. L. Collins。《图中的对称性破缺》。Electronic Journal of Combinatorics，3:R18，1996。[2] V. Arvind、C. T. Cheng 和 N. R. Devanur。关于平面图及更广泛图类的区分数计算：一种计数方法。SIAM Journal on Discrete Mathematics，22(4):1297–1324，2008。[3] A. Borri、T. Calamoneri 和 R. Petreschi。通过图的叠加识别单图。J. Graph Algorithms Appl.，15(3):323–343，2011。[4] D. L. Boutin。利用确定集识别图自同构。Electron. J. Comb.，13(1)，2006。

[5] T. Calamoneri 和 R. Petreschi。单图的 $L(2,1)$-标记。《离散应用数学》，159:1196–1206，2011。[6] C. T. Cheng。关于树和森林的区分数计算。Electronic Journal of Combinatorics，13:R11，2006。[7] C. T. Cheng。关于区间图的区分数和区分色数计算及其他结果。《离散数学》，309(16):5169–5182，2009。[8] C. T. Cheng。Tyshkevich 分解定理与单图的区分数。《离散数学》，348(8)，2025。[9] C. T. Cheng、K. L. Collins 和 A. Trenk。分裂图与 Nordhaus-Gaddum 图。《离散数学》，339:2345–2356，2016。[10]D. Erwin 和 F. Harary。通过固定节点破坏自同构。《离散数学》，306(24):3244–3252，2006。[11] G. Fijavz 和 B. Mohar。曲面上图的刚性与分离指标。Graphs Comb.，26(4):491–498，2010。[12] S. L. Hakimi。关于一组整数能否实现为线性图顶点的度。i.《工业与应用数学学会杂志》，10(3):496–506，1962。[13] P. Hammer 和 B. Simeone。图的分裂度。《组合数学》，1:275–284，1981。[14] V. Havel。关于有限图存在性的一条注记。ˇCasopis pro pˇestov ´an ´ı matematiky，80(4):477–480，1955。[15] D. J. Kleitman 和 S.-Y. Li。关于唯一图序列的一则注记。《应用数学研究》，54:283–287，1975。[16] Y. Nakahata。关于单图的团宽度。arXiv 1905.12461，2022。[17] R. Tyshkevich。《图序列和单图的分解》。《离散数学》，220:201–238，2000。[18] R. Whitman。由度序列和色数确定的图。arXiv 2406.03667，2024。

附录

下面给出算法 IsUnigraph(G)。给定图 G，该算法将判断 G 是否为单图。如果 G 是单图，它还会返回一个列表，其中包含 G 各个不可分解分量的类型。该算法依赖于 Decompose(G) 和 DetermineSplit(DegSeq) 这两个算法，它们可在 [8] 中找到。

Decompose(G) 返回一个栈，其中包含 G 规范分解中各个不可分解分量的（配对）度序列。栈顶是 G 第一个不可分解分量的度序列，该分量可以是非分裂图，也可以是分裂图。其后是第二个不可分解分量的度序列，依此类推。DetermineSplit(DegSeq) 用于判断度序列为 DegSeq 的图是否为分裂图。如果该图是分裂图，该算法将返回对应于一个 KS-划分的配对度序列；否则，返回原始 DegSeq。

DetermineNonSplitType(DegSeq) 假定 DegSeq 是某个非分裂图的度序列。如果该图是不可分解非分裂单图，该算法将返回该图的类型；否则，返回 $null$。对于分裂图，DetermineSplitType(DegSeq) 的行为相同。

算法1 IsUnigraph(G)

```
T ←Decompose(G)
componentList ← []
DegSeq ←DetermineSplit(T.top)
若 DegSeq 不是配对度序列，则
    ans ←DetermineNonSplitType(DegSeq)
    如果 ans = null 则
        返回（“不是单图”）
    else
        将 ans 追加到分量列表
        T.pop
当 T.size ≥ 1 时
    DegSeq ← T.pop
    ans ←DetermineSplit类型(DegSeq)
    如果 ans = null 则
        返回（“不是单图”）
    else
        将 ans 追加到分量列表
返回（“单图”，分量列表）
```

设 H 是某个非分裂图，其度序列为 DegSeq。DetermineNonSplitType(DegSeq) 会考察 H 和 $\overline{H}$，并检查其中是否有一个与 C_5、mK_2、U_2（m、ℓ）或 U_3（m）同构；这四个图是不可分解非分裂单图。如果找到匹配项，算法会返回匹配信息—— H （原始图或补图）的变体，以及与之匹配的不可分解非分裂单图。否则，算法返回 $null$，表示 H 不是不可

分解非分裂单图。提醒读者，如果 $\overline{H}$ 与例如 mK_2 同构，那么 H 也与 $\overline{mK_2}$ 同构。因此，算法返回的匹配信息就是 H 的类型。

设 S 为 H 或 $\overline{H}$ 的度序列。为检查是否匹配，算法会将 S 与 C_5、mK_2、$U_2(m、\ell)$ 和 $U_3(m)$ 的度序列进行比较。我们以 $U_2(m、\ell)$ 为例说明。由表 1 可知，$U_2(m、\ell)$ 的度序列为 $(\ell、1^{2m+\ell})$，其中 $m \geq 1$、$\ell \geq 2$。要确定 S 是否是某个 m 和 ℓ 对应的 $U_2(m、\ell)$ 的度序列，算法首先检查 S 的“长度”是否为 2；也就是说，S 有两个不同的度，因此 $S = (d_1^{r_1}、d_2^{r_2})$，其中 $d_1 > d_2$。接着，它验证 $r_1 = 1$、$d_2 = 1$ 和 $r_2 - d_1$ 是否为偶数。若是，则将参数 m 设为 $(r_2 - d_1)/2$，并将 ℓ 设为 d_1。最后，它检查 $m \geq 1$、$\ell \geq 2$ 是否成立。若这些条件全部满足，则 $d_1 = \ell$、$r_1 = 1$、$d_2 = 1$、$r_2 = 2m + \ell$，其中 $m \geq 1$、$\ell \geq 2$。也就是说，$S = (\ell、1^{2m+\ell})$，即 $U_2(m、\ell)$ 的度序列。

$U_3(m)$ 的过程略有不同。根据表 1，$U_3(m)$ 的度序列为 $(2m + 2, 2^{2m+3})$，其中包含 $m \geq 1$。在对 S 的长度、d_1 的奇偶性以及 r_1 和 d_2 的值进行初始检查后，算法根据 d_1 推导出参数 m。仅检查 $m \geq 1$ 是不够的，因为 m 也与 r_2 有关。它还会验证 $r_2 = 2m + 3$。这样，对于某个 $m \geq 1$，有 $(d_1^{r_1}, d_2^{r_2}) = (2m + 2, 2^{2m+3})$。

算法2 DetermineNonSplitType(DegSeq)

```
变体 ← ["原始"、"补图"]
测试 ← ["C5"、"MK2"、"U2"、"U3"]
对于 variant 每个变体，执行
    S ← applyVariant(variant, DegSeq)
    对测试中的 test 执行
        如果 test = "C5" 且 S = (2^5)，则
            返回 (variant, "C5")
        如果 test = "MK2" 且 length(S) = 1 则
            (d_1^{r_1}) ← S
            如果 d_1 = 1 且 r_1 为偶数，则
                m ← r_1/2
                如果 m ≥ 2 那么
                    返回 (variant, "MK2"，m)
        如果 test = "U2" 且 length(S) = 2 那么
            (d_1^{r_1}, d_2^{r_2}) ← S，其中 d_1 > d_2
            如果 r_1 = 1 且 d_2 = 1 且 (r_2 − d_1) 是偶数，则
                m ← (r_2 − d_1)/2，  ℓ ← d_1
                如果 m ≥ 1 且 ℓ ≥ 2，则
                    返回 (variant，  "U2"，  m，  ℓ)
        如果 test = "U3" 且 length(S) = 2，则
            (d_1^{r_1}，d_2^{r_2}) ← S，其中 d_1 > d_2
            如果 d_1 是偶数，且 r_1 = 1 和 d_2 = 2，则
                m ← (d_1 − 2)/2
                如果 m ≥ 1 且 r_2 = 2m + 3，则
                    返回 (variant, "U3"，m)
返回(null)
```

算法 3 DetermineSplitType(DegSeq)

变体 ← [“原始”、“逆”、“补图”、“逆补”]
测试 ← [“SVG”、“SPQ”、“S2”、“S3”、“S4”]
对 $variant$ 各变体执行
  $S = (D_A;\ D_B) \leftarrow applyVariant(variant,$ DegSeq)
  对 $test$ 各测试执行
    如果 $test =$ “SVG” 则
      如果 $length(D_A) = 1$ 且 $length(D_B) = 0$ 则
        $(d_1^{r_1}) \leftarrow D_A$
        如果 $d_1 = 0$ 且 $r_1 = 1$ 则
          返回($variant$, “SVG-K1”)
      如果 $length(D_A) = 0$ 且 $length(D_B) = 1$ 则
        $(d_2^{r_2}) \leftarrow D_B$
        如果 $d_2 = 0$ 和 $r_2 = 1$ 则
          返回($variant$, “SVG-S1”)
    如果 $test =$ “SPQ”且 $length(D_A) = 1$ 和 $length(D_B) = 1$ 则
      $(d_1^{r_1}) \leftarrow D_A$, $(d_2^{r_2}) \leftarrow D_B$
      如果 r_2 模 $r_1 = 0$ 且 $d_2 = 1$ 则
        $p \leftarrow r_2/r_1$, $q \leftarrow r_1$
        如果 $p \geq 1$ 且 $q \geq 2$ 且 $d_1 = p + q - 1$ 则
          返回($variant$, “SPQ”, p, q)
    如果 $test =$ “S2” 且 $length(D_A) \geq 2$ 且 $length(D_B) = 1$ 则
      $(d_1^{r_1}, d_2^{r_2}, \ldots, d_m^{r_m}) \leftarrow D_A$ 与 $d_1 > d_2 > \cdots > d_m$ 且 $(d_B^{r_B}) \leftarrow D_B$
      $n \leftarrow r_1 + r_2 + ... + r_m$
      对于 $i \leftarrow 1$ 到 m 执行
        $p_i \leftarrow d_i - n + 1$, $q_i \leftarrow r_i$
      如果 $p_m \geq 1$, 则
        如果 $d_B = 1$ 且 $r_B = \sum_{i=1}^{m} p_i q_i$, 则
          返回($variant$, “S2”, (p_1, q_1, p_2, q_2, ..., p_m, q_m))
    如果 $test =$ “S3”且 $length(D_A) = 1$ 且 $length(D_B) = 2$, 则
      $(d_1^{r_1}) \leftarrow D_A$, $(d_2^{r_2}, d_3^{r_3}) \leftarrow D_B$ 与 $d_2 > d_3$
      如果 $r_2 = 1$ 和 $d_3 = 1$ 则
        $p \leftarrow d_1 - r_1$, $q_1 \leftarrow d_2$, $q_2 \leftarrow r_1 - d_2$
        如果 $p \geq 1$ 和 $q_1 \geq 2$ 和 $q_2 \geq 1$ 则
          如果 $r_3 = pq_1 + (p + 1)q_2$ 则
            返回($variant$, “S3”, p, q_1, q_2)
    如果 $test =$ “S4”且 $length(D_A) = 2$ 和 $length(D_B) = 1$ 那么
      $(d_1^{r_1}, d_2^{r_2}) \leftarrow D_A$ 与 $d_1 > d_2$, $(d_3^{r_3}) \leftarrow D_B$
      如果 $r_1 = 1$ 且 $d_3 = 2$ 那么
        $q \leftarrow r_2 - 2$, $p \leftarrow d_2 - 3 - q$
        如果 $p \geq 1$ 且 $q \geq 1$ 那么
          如果 $d_1 = 2(p + q + 1) + qp$ 且 $r_3 = qp + 2p + q + 1$ 则
            返回($variant$, “S4”, p, q)
返回($null$)

这一次，设 H 是某个分裂图，DegSeq 是它的度序列。算法 DetermineSplitType(DegSeq) 的工作方式与 DetermineNonSplitType 相同：将度序列分别与单顶点图、 $S(p,q)$、 $S_2(p_1,q_1,\ldots,p_k,q_m)$、$S_3(p,q_1,q_2)$ 和 $S_4(p,q)$ 的度序列进行比较。算法首先检查度序列是否具有正确的基本格式，然后推导参数并检查参数是否有效。最后，如果度序列中某些部分虽由这些参数决定、却未用于推导参数，算法就会检查这些部分的取值是否正确。

设 S 是 H、 $\overline{H}$、 H^I 或 $\overline{H}^I$的度序列。下面演示算法检查 S是否与某个 p_i、 q_i、 $i=1$、$\ldots$、m对应的 $S_2(p_1,q_1,\ldots,p_m,q_m)$同构时，上述步骤如何进行。根据表2，后者的度序列为$((p_1+n-1)^{q_1},\cdots,(p_m+n-1)^{q_m};\ 1^{p_1q_1+\cdots+p_mq_m})$，其中 $n=\sum_{i=1}^{m}q_i$，且对于 $i=1,\ldots,m$， $m\geq 2$、 $p_1>p_2>\ldots>p_m\geq 1$ 和 $q_i\geq 1$。

首先，算法检查配对度序列（D_A， D_B）中， D_A的长度是否至少为 2，且 D_B的长度是否为 1。若是，则其格式为（$d_1^{r_1}$， $d_2^{r_2}$，$\ldots$，$d_m^{r_m}$；$d_B^{r_B}$），其中 $m\geq 2$和 $d_1>d_2>\ldots>d_m$。然后，它为 $i=1$，$\ldots$，m设置参数 $n=\sum_{i=1}^{m}r_i$、 $p_i=d_i-n+1$和 $q_i=r_i$。接下来，它验证 $p_m\geq 1$ 和 $d_B=1$。由于 $r_i\geq 1$，因此无需检查 $q_i\geq 1$ 。最后，它检查 $r_B=\sum_{i=1}^{m}p_iq_i$。如果所有这些条件都满足，则 $d_i=p_i+n-1$和 $r_i=q_i$for$i=1$，$\ldots$， m，其中 $n=\sum_{i=1}^{m}q_i$。由于 d_i's 按递减顺序取值，因此 p_i's 也按递减顺序，且 $p_m\geq 1$。同样， $d_B^{r_B}=(1^{p_1q_1+\cdots+p_mq_m})$。

在 DetermineNonSplitType 和 DetermineSplitType 中， H 的变体数量以及与 H 的一个变体进行比较的图的数量都是常数。求出 H 每个变体的度序列需要 $O(|V(H)|)$ 时间。检查 H 的一个变体是否同构于某个不可分解单图所需的时间为 $O(|V(H)|)$。因此，这两种算法的运行时间均为 $O(|V(H)|)$。这意味着在 IsUnigraph 中，第 1 至 18 行需要 $O(|V(G)|)$ 时间。Decompose 和 DetermineSplit 的运行时间也都是线性的。因此，IsUnigraph 的运行时间为线性时间。