arXiv:2512.24061v1 [math.CO] 30 Dec 2025

关于 33 点 Erdős–Szekeres问题的札记

Dumitru Bogdan *

2025年12月

摘要

确定 ES(7) 是平面埃尔德什–塞凯赖什问题中第一个尚未解决的情形；该问题的一般猜想预言 ES(7) $= 33$。我们提出了一种针对 33 点情形的 SAT编码，基于三元组定向变量和四点集凸性判据来排除凸七边形，并结合凸层锚定约束。该框架为一系列锚定子族生成了不可满足证明证书。我们还报告了不同配置间显著的运行时间变异性，其中包括重尾行为；目前，这种行为主导着计算工作量，也促使我们进一步改进编码。

关键词： Erdős–Szekeres问题、SAT求解、离散几何、凸层、序型、自动推理。

1 引言

在他们发表于1935年的开创性论文中，埃尔德什和塞凯赖什研究了最小整数 ES(k)：在平面一般位置（任意三点不共线）的任意 ES(k) 个点集中，必定包含 k个处于凸位置的点 [1]。他们猜想其精确公式为

$$\mathrm{ES}(k) \;=\; 2^{k-2}+1 \qquad (k \geq 3).$$

该公式已在 $k \leq 6$ 的情形下得到验证。此外，经典的埃尔德什–塞凯赖什构造给出了相应的下界 ES(k) $\geq 2^{k-2}+1$ [1]。 $k = 7$ 的情形仍未解决：目前尚不清楚每个由 33 个一般位置的点组成的集合是否都包含一个凸七边形，也就是说，ES(7) 是否满足 $= 33$。

ES(k) 的上界已得到大幅改进；例如，Suk 证明了 ES(k) $\leq 2^{k+o(k)}$ [2]，Holmsen–Mojarrad–Pach–Tardos 则将其改进为 ES(k) $\leq 2^{k+O(\sqrt{k}\log k)}$[3]。计算方法也发挥了重要作用。Szekeres和Peters给出了计算机辅助证明，证明 ES(6) $= 17$ [4]；此外，在相关问题中也探索了基于SAT的方法（例如 [5, 6]）。近年来，针对埃尔德什–塞凯赖什类问题的SAT研究再度活跃：Scheucher为高维埃尔德什–塞凯赖什参数及相关问题开发了SAT模型，并通过证明证书验证了不可满足结果 [8]；Heule和Scheucher则利用紧凑的SAT编码、大规模搜索和证明生成，确定了精确的空六边形数 h(6) $= 30$ [9]。尽管空洞问题和高维变体不同于经典的（非空）平面 ES(7) 问题，这些研究展现了当前基于SAT的方法所能达到的水平，也促使人们进一步探索针对33点问题设计的编码。

在本研究中，我们通过将几何一致性及排除凸七边形编码为布尔可满足性问题，对 $n = 33$ 情形进行了计算研究。我们的编码以三元组上的定向变量为基础，并将凸位置精确归约为关于四点子集的约束。为减少对称性并支持系统化实验，

*计算机科学学院，布加勒斯特大学，bogdan.dumitru@fmi.unibuc.ro

我们还施加了凸层（凸包模板）锚定约束；对于一些难解的模板，则采用一种简单的子立方参数，进一步固定连续层之间的相对对齐方式。利用这一框架，我们为不断增多的锚定子族获得了不可满足性证书，并报告了各配置之间显著的运行时间差异：一些子问题在商用硬件上需要数周才能完成。

本文结构如下：第2节介绍三元组定向变量以及一族五点 CC 风格蕴含式（源自 Knuth）。第3节陈述凸位置的四点集判据，并说明如何由此得到紧凑的“无凸 7 点集”约束。第4节介绍凸层锚定模板以及一个简单的子立方参数。第5–6节概述实例规模和有代表性的计算结果。第7节讨论瓶颈并展望未来工作。

2 预备知识：三元组定向与 CC 风格约束

2.1 定向变量

设 $[n] = \{0, 1, \ldots, n-1\}$ 为点的标签。对于处于一般位置且可实现的点集，每个有序三元组（a、b、c）都有明确定义的定向（顺时针或逆时针）。我们用符号谓词表示这一点

$$\chi(a, b, c) \in \{+, -\},$$

交换参数时交替变号。

定义 1（3 杯 / 3 帽（约定））。固定一种约定：$\chi(a, b, c) = +$ 对应一个**三杯**，而 $\chi(a, b, c) = -$ 对应一个**三帽**（反之亦可）。约定本身是任意的，但必须始终如一地使用。

我们为每个包含 $i < j < k$ 的无序三元组 $\{i, j, k\}$ 引入一个布尔变量：

$$x_{ijk} \in \{0, 1\}.$$

对于任意有序三元组 (a, b, c)，令 (i, j, k) 为其排序后的三元组，并定义一个带符号文字

$$\ell(a, b, c) \in \{\pm x_{ijk}\}$$

具体取决于 (a, b, c) 是 (i, j, k) 的偶排列还是奇排列。这样便以紧凑的方式在文字层面强制实现反对称性。

2.2 一个五点蕴含式（CCaxiom5）

为了将赋值限制在更接近可实现序型的范围内，可以施加有向拟阵／手性拓扑公理。我们采用一族遵循 Knuth 的 CC 公理思路的五点蕴含式 [7]。一种常见形式是，对于互不相同的点 p_1、.....、p_5：

$$\chi(p_1, p_2, p_3) = \chi(p_1, p_2, p_4) = \chi(p_1, p_2, p_5) = \chi(p_1, p_3, p_4) = \chi(p_1, p_4, p_5) = + \implies \chi(p_1, p_3, p_5) = +.$$

转换为合取范式后，这会变成如下形式的一个子句

$$\neg A_1 \vee \neg A_2 \vee \neg A_3 \vee \neg A_4 \vee \neg A_5 \vee C,$$

其中，A_i 和 C 是三元组定向变量的带符号文字。

约简生成。在本文所述的实现中，我们生成了这些五点子句的一个约简子集（这是出于对称性和冗余方面的考虑）。这是一种用于提升性能的松弛。

求解器结果的解读。省略子句只会扩大可接受布尔赋值的集合。因此，对于松弛实例得出的**不可满足**结果，仍可作为任何包含被省略子句的更强表述的有效证明证书。本文始终只报告不可满足结果。

3 凸位置的四点集判据

直接对“没有凸七边形”进行编码，往往需要遍历所有七点子集及其大量循环排列，从而导致在 $(n,k)=(33,7)$ 处出现巨大的子句爆炸。我们改用一种精确归约，将问题转化为四点子集。

命题 1 (凸位置的四点集判据)。 设 S 是平面上处于一般位置的有限点集。那么， S 处于凸位置，当且仅当 S 的每个四点子集都处于凸位置。

证明概要。 如果 S 处于凸位置，那么 S 的每个子集也都处于凸位置，因此每个四点子集都是凸的。

反过来，假设 S 不 处于凸位置。那么，存在某个点 $p\in S$ 不是 S 的凸包顶点，因此 p 严格位于凸多边形 $P=\mathrm{conv}(V)$ 内部，其中 $V\subseteq S$ 是凸包顶点的集合。固定一个凸包顶点 $v_0\in V$，并从 v_0 向所有不相邻的凸包顶点作对角线，对 P 进行三角剖分；这样便将 P 划分为若干个顶点属于 V 的三角形。由于 p 位于 P 内部，它必定位于其中一个三角形内，设该三角形为 $\triangle abc$，其顶点为 $a,b,c\in V\subseteq S$。那么，4 点集 $\{a,b,c,p\}$ 不处于凸位置（其中一个点位于另外三点构成的三角形内部），这与 S 的每个四点子集都处于凸位置的假设矛盾。□

4 编码 4 点类型并排除凸七点集

4.1 四个循环三元组定向

固定一个 4 点集 $\{a,b,c,d\}$ 及其排序（a， b， c， d）。考虑循环三元组

$$(a,b,c),\quad (b,c,d),\quad (c,d,a),\quad (d,a,b).$$

每个三元组 都有一个方向文字 , 因此，该 4 点集在 $\{+$ 中诱导出一个长度为 4 的符号模式 $,-\}^4$。并非所有 $2^4=16$ 模式都会出现在处于一般位置的4个点中；恰好有14种。在我们当前的实现中，每个4 点集引入14个选择变量，每个可实现模式对应一个变量，并约束这些变量两两互斥且至少有一个成立。

4.2 选择变量与合取范式

形式化地，对于每个4 点集 Q和每个可实现模式索引 $p\in\{1,\dots,14\}$，我们引入一个变量 $t_{Q,\,p}$；讨论固定的4 点集时，我们省略 Q 下标。

对于每个4 点集，我们引入14个选择变量 $t_1,\dots,t_{14}$。每个选择变量都被约束为等价于四个文字（即循环三元组定向）的合取，使用标准的实化模板

$$t\leftrightarrow(L_1\wedge L_2\wedge L_3\wedge L_4),$$

编码为5个子句。我们还要求至少有一个选择变量成立。因此，每个4 点集贡献$14\times 5+1=71$ 个子句。

注1 （凸与非凸模式）。 在此使用的生成器版本中，6种可实现模式被视为凸模式：

$$++++,\quad ----,\quad ++--,\quad --++,\quad -++-,\quad +--+,$$

其余 8 种可实现模式则被视为非凸。

4.3 不存在凸 7 点集子句

对于每个 7 点集 K，命题 1 意味着，K 处于凸位置，当且仅当它的所有 4 元子集都是凸的。因此，要排除凸七点集，只需确保每个 7 点集都至少包含一个非凸 4 元子集。在我们的编码中，对于每个 7 点集 K，我们添加一个子句

$$\bigvee_{Q\in\binom{K}{4}} \bigvee_{p\in\mathcal{N}} t_{Q,p},$$

其中，$\mathcal{N}$ 索引 4 点集 Q 的非凸模式。该子句的长度为 $35\times 8=280$。

5 凸层（凸包模板）与简单子立方化参数

为减少对称性并探索结构化子族，我们加入凸层约束（也称为凸层分解），以强制形成大小预先指定的嵌套凸层序列。

5.1 凸层（凸包模板）锚定

固定 n 和一个层大小向量

$$\mathbf{h}=(h_0,h_1,\dots,h_{r-1}),\qquad h_i\ge 3,\qquad H\coloneqq\sum_{i=0}^{r-1}h_i\le n.$$

我们将点 0、……、h_0-1 视为按循环顺序排列的外层凸 h_0 边形的顶点，将点 h_0、……、h_0+h_1-1 视为下一层的点，依此类推。其余 $n-H$ 个点视为不受约束的内部点。

然后，我们添加强制以下条件成立的单子句：

- **层内凸性：** 层内按选定循环顺序排列的每组三个顶点都具有固定定向。

- **嵌套：** 每个更深层的点都位于外层某条有向边的内侧（通过对适当三元组的固定定向进行编码）。

5.2 一个简单的子立方参数（锚定连续层)

对于某些困难的凸层模板，我们进一步添加由一个向量控制的一小组单子句

$$\mathbf{w}=(w_0,w_1,\dots,w_{r-1}),$$

其中，每个 w_i 都被解释为层 i 内的偏移量（按照约定，$w_0=0$ ）。该参数用作一种手动子立方划分机制：对$\mathbf{w}$ 的不同选择会产生相互独立的子实例。

几何意义。令 L_i 表示层 i，并令 $s_i=\sum_{j<i}h_j$ 为其起始索引，因此 $L_i=\{s_i,\dots,s_i+h_i-1\}$。对于每个 $i\ge 1$，我们在前一层L_{i-1}中固定一个锚点顶点 $a=s_{i-1}$，并从当前层中选取两个顶点：

$$b=s_i,\qquad c=s_i+w_i.$$

接着，我们对方向文字添加单元约束，强制所有编号为 $>s_i$ 的点（即 L_i 的所有其余顶点，以及更深层中的所有点／剩余点）都位于 a处的楔形内

该楔形由射线 ab 和 ac围成。等价地，从 a的视角来看，点对（b，c）被强制表现为内层的两个极端（支撑）顶点，从而固定 L_{i-1} 与 L_i之间部分的相对"旋转"。当 $w_i = 0$ （因此 $c = b$）时，会省略在第 i 层进行的这项额外锚定。

在计算中的作用。 {w} 约束并非旨在提供一种平衡或穷尽式的拆分策略；相反，它们提供了一种规模较小、基于几何信息的方法，将某些高度对称的模板划分为子实例，而这些子实例的运行时间可能大不相同。

6 $n(n,k) = (33,7)$的实例规模

6.1 变量计数

三元组变量。 共有 $\binom{33}{3} = 5456$ 个无序三元组，因此有 5456 个基础定向变量。

四元集选择器。 共有 $\binom{33}{4} = 40920$ 4 个 4 点集，每个 4 点集有 14 个选择器，共计 $40920 \times 14 = 572$，880 个变量。

总计。 因此，基础实例包含

$$5456 + 572{,}880 = 578{,}336$$

布尔变量。

6.2 子句计数

占主导地位的子句块有：

- **约化五点约束：** 对于 $n = 33$，此块在记录的生成器版本中贡献了 9，493，440 个子句。
- **4-集一致性：** $71\binom{33}{4} = 2$，905，320 个子句。
- **无凸七点集约束（通过 4 点集）：** $\binom{33}{7} = 4$，272，048 个子句，每个子句长度为 280。

因此，基础合取范式（不含凸包约束）有

$$9{,}493{,}440 \;+\; 2{,}905{,}320 \;+\; 4{,}272{,}048 \;=\; 16{,}670{,}808$$

个子句。凸包约束只会增加单子句（数百到几千个），因此不会显著改变合取范式的规模，但可能会显著影响运行时间。

7 结果与观察

本节总结了在凸层模板下，对 $(n,\ k) = (33,\ 7)$ 观察到的一组不可满足运行时间，以及（可选的）w 子立方参数。所列时间为单线程实际运行时间，应作定性解读。

所有合取范式实例均由 Python 脚本生成（PySAT [10]），并使用 Kissat [11] 求解（单线程运行）。

备注 2 （运行时间变异性）。 即使是在密切相关的凸层系列中，运行时间也会有很大差异。例如，在 5^6 模板中，我们观察到子立方体的运行时间从 2.50×10^3 秒到 2.28×10^6 秒不等；而在 4-支配模板中，则从 8.76×10^4 秒到 2.10×10^6 秒不等。这促使我们改进编码（以减少子句/文字规模并增强传播），并为最难处理的对称构型采用更强的几何锚定。

层大小 $\mathbf{h}$（总和 H）	子立方体参数 $\mathbf{w}$	观测时间（秒）
$6, 6, 6, 3, 6, 6\,(H = 33)$	（未记录）	$\approx 1.73 \times 10^5$
$3^{11}\ (H = 33)$	（未记录）	$\approx 6.05 \times 10^5$
$3,\ 3,\ 4,\ 3,\ 3,\ 6,\ 6,\ 5\ \ (H = 33)$	$[0, 1, 1, 1, 1, 1, 4, 4]$	2.26×10^5
$5^6\ (H = 30)$	（基准运行）	2.59×10^5
$5^6\ (H = 30)$	$[0, 4, 4, 4, 4, 4]$	2.50×10^3
$5^6\ (H = 30)$	$[0, 1, 1, 1, 1, 1]$	1.60×10^4
$5^6\ (H = 30)$	$[0, 1, 1, 1, 0, 0]$	2.28×10^6
$4^8\ (H = 32)$	$[0, 3, 3, 3, 3, 3, 3, 3]$	8.76×10^4
$4^8\ (H = 32)$	$[0, 1, 2, 3, 1, 2, 3, 1]$	3.19×10^5
$4, 3, 4, 4, 4, 4, 4, 4\,(H = 31)$	$[0, 1, 3, 2, 1, 2, 1, 1]$	2.10×10^6
$4, 3, 3, 5, 4, 6, 3, 4\,(H = 32)$	$[0, 2, 2, 4, 1, 1, 1, 1]$	3.15×10^3
$3, 5, 3, 5, 3, 3, 5, 5\,(H = 32)$	$[0, 1, 2, 1, 2, 1, 1, 4]$	3.59×10^4
$3,\ 4,\ 3,\ 4,\ 4,\ 3,\ 4,\ 4,\ 3\ \ (H = 32)$	$[0, 1, 2, 1, 2, 1, 1, 3, 2]$	9.06×10^5

表 1：在凸层模板和$\mathbf{w}$参数下，$(n,\ k) = (33,\ 7)$ 的部分不可满足运行时间（单线程墙钟秒数）。

可复现性。 本文所报告结果所用的 TheSAT 实例生成器及脚本均已公开。[1] 该代码库包含与此处所述编码相对应的实现，以及用于所报告凸层模板的配置。

8 讨论与未来工作

本文所 प्रस्तुत的编码使得生成并求解源自 $(33,\ 7)$ 情形的大型可满足实例成为可行，至少对于通过凸层锚定获得的许多结构化子族而言如此。主要的实际挑战在于，求解器时间在相邻的锚定族之间变化很大。

未来工作

我们列出两个看起来很有前景的编码方向，并将在未来的修订版中进行实验性测试。

1. **降低 4 点集编码的粒度**。当前实现为每个 4 点集使用多个选择变量，以区分所有可实现的 4 点类型。然而，7 点集排除约束最终只取决于每个 4 点集是“凸”还是“非凸”这一粗略区分。因此，可以考虑用一种更粗略的表示来取代详细的 4 点类型记录，只保留 7 集约束所需的信息，从而减少辅助变量和 4 点集子句。

2. **利用更粗粒度的 4 集信息简化 7 集约束**。目前，每个 7 集子句都是一个很长的析取式，由许多特定模式文字展开而成。如果能为每个 4 点集提供一个更粗粒度的凸／非凸指示变量，相应的 7 集约束就能用短得多的子句表示。除了降低内存压力外，较短的子句通常还能加强传播，因为当一个 7 集中的许多 4 元子集都被迫为凸时，该 7 集便会受到约束。

[1] https://github.com/bogdan27182/esc-paper

第二个（正交的）方向是加强对外层凸层的几何锚定，在保持预期情形划分完备性的同时减少对称性；这一方向将通过实验进行探索。

9 结论

我们针对 $(n, k) = (33, 7)$ 平面埃尔德什–塞凯赖什问题提出了一种SAT编码，该编码使用三元组定向变量、精简的CC风格五点约束族，以及精确的四点集判据（命题 1）来排除凸 7 点集。该编码支持额外的凸层锚定约束，并为若干锚定族生成了不可满足性证书。要取得进一步进展，似乎既需要改进编码，也需要针对最难处理的对称构型加强几何锚定。

参考文献

[1] P. 埃尔德什和 G. 塞凯赖什。几何中的一个组合问题。*Compositio Mathematica*，2：463–470，1935年。[2] A. Suk。关于埃尔德什–塞凯赖什凸多边形问题。《美国数学会杂志》，30(4)：1047–1053，2017。（预印本：arXiv:1604.08657。）[3] A. Holmsen、H. N. Mojarrad、J. Pach 和 G. Tardos。埃尔德什–塞凯赖什问题的两种扩展。《欧洲数学会杂志》，22(12)：3997–4010，2020年。[4] G. 塞凯赖什和 L. Peters。17 点埃尔德什–塞凯赖什问题的计算机解。《*ANZIAM*杂志》，48(2)：151–164，2006年。[5] M. Balko 和 P. Valtr。对埃尔德什–塞凯赖什猜想的SAT攻击。载于第*33*届国际计算几何研讨会论文集（*SoCG 2017*）。Schloss Dagstuhl–Leibniz 信息学中心，2017年。[6] F. Marić。关于至多有 6 个顶点的多边形的埃尔德什–塞凯赖什定理的快速形式化证明。《自动推理杂志》，58(4)：539–550，2017年。[7] D. E. Knuth。《公理与凸包》。施普林格，1992年。[8]M. Scheucher。对 $\mathbb{R}^d$ 中埃尔德什–塞凯赖什数及空六边形定理的SAT攻击。《几何与拓扑计算》，2(1)，2：1–2:13，2023年。doi:10.57717/cgt.v2i1.12。[9] M. J. H. Heule 和 M. Scheucher。快乐结局：每个由 30 个点组成的集合中都有一个空六边形。*arXiv* 预印本 arXiv:2403.00737，2024年。[10] A. Ignatiev、A. Morgado 和 J. Marques-Silva。PySAT：用于使用SAT预言机进行原型设计的Python工具包。载于*SAT 2018*论文集，《计算机科学讲义》10929，428–437页。施普林格，2018年。[11] A. Biere、M. Fleury、L. Heisinger 和 C. Sinz。CaDiCaL、Kissat、Paracooba、Plingeling 和 Treengeling 参加SAT竞赛2020。*SAT*竞赛*2020*论文集，2020年。