$\gamma(Q_{35}) = 19$ and certified queen domination numbers up to $n = 46$

Jason Davies*

November 21, 2026

hexagon:2611.00830v1

Abstract

We show that no 18 queens dominate the 35×35 board, so that the domination number of the queen's graph Q_{35} is $\gamma(Q_{35}) = 19$ (OEIS A075458); since 2001 it had been known only that $\gamma(Q_{35}) \in \{18, 19\}$. An independent dominating set of 19 queens found by Östergård and Weakley then gives $i(Q_{35}) = 19$, and 35 is the smallest odd $n > 15$ with $\gamma(Q_n) = \lceil n/2 \rceil + 1$. The proof is computer-assisted. It splits a hypothetical dominating set by the box spanned by its outermost empty rows and columns, as in the *Hint and Conquer* method for even boards, and further by the deficits of occupied lines of the four kinds. Exact counting identities for the border and for the sides of the box make each of the resulting 128,941 cases a small SAT problem. Every case was refuted by CaDiCaL with a DRAT proof checked by `drat-trim`, in about 81 CPU hours. The reduction to the cases, the completeness of the case list and the upper bound are formalised in Lean 4 with Mathlib, modulo the CNF encoding and the SAT certificates. A second, separately written encoding, solved with Kissat and without any symmetry breaking, independently refutes the 585 hardest cases and a sample of 2200 others. The same method, with DRAT proofs, gives $\gamma(Q_{40}) = 21$, $\gamma(Q_{42}) = 22$, $\gamma(Q_{44}) = 23$ and $\gamma(Q_{46}) = 24$, the first certified values for these boards; they confirm claims made in 2009 on the basis of an uncertified integer-programming computation. With them, $\gamma(Q_n)$ is known for every $n \leq 46$. The method also confirms $\gamma(Q_{32}) = 17$, $\gamma(Q_{34}) = 18$ and $\gamma(Q_{36}) = 19$, and $\gamma(Q_{26}) = 14$ and $\gamma(Q_{28}) = 15$, which were first established in *Hint and Conquer* (2026). For $n = 26$ and $n = 28$ we also give a second, completely independent proof, by a relaxation of the grid of empty rows and columns to covers by diagonals, with DRAT certificates for both of its stages and a complete replay of every certificate formula.

AI transparency statement. The computer-assisted proofs, the code, the Lean formalisation and this manuscript were produced by artificial intelligence. The values for $n \leq 36$ other than $n = 35$ were established earlier by others, and those for $n = 40, \ldots, 46$ were claimed earlier without certificates (Section 1). The human author initiated the broader research and directed the AI systems, but did not write the text and has not checked the arguments. Details are given at the end of the paper.

1 Introduction

The queen's graph Q_n has the n^2 squares of the $n \times n$ board as vertices, two squares being adjacent when they share a row, a column or a diagonal. Its domination number $\gamma(Q_n)$ is the least number of queens that occupy or attack every square, and its independent domination number $i(Q_n)$ the least number of pairwise non-attacking queens that do so. The problem goes back to de Jaenisch

*jason@jasondavies.com

(1862) and is Problem C18 in Guy's collection; see the surveys [6, 24] and the introduction of [26]. The general lower bound

$$\gamma(Q_n) \;\geq\; \tfrac{n-1}{2}$$

was proved independently by Spencer (see [6]) and by Raghavan and Venkatesan [21], and equality holds only for $n \in \{3, 11\}$ [25, 8]. So $\gamma(Q_n) \geq \lceil n/2 \rceil$ for all other n. For every $n \leq 120$ other than 3 and 11 the value is known to be $\lceil n/2 \rceil$ or $\lceil n/2 \rceil + 1$ [20, 26]. For $n \leq 25$ the exact values come from constructions [20, 14], from exhaustive searches by Kearse and Gibbons ($n \leq 19$) [14] and by Bird ($n = 20, 22, 24$, with a general-purpose dominating-set solver on 64 processors) [4], and, for $n \leq 19$, from proof-producing SAT solving [22].

For odd n, Östergård and Weakley [20] list the values $\gamma(Q_n) = \lceil n/2 \rceil$ for many n, among them every $n \equiv 1 \pmod 4$ up to 129 and all odd n with $17 \leq n \leq 33$, and they leave $n = 35$ open: "$i(Q_{35}), \gamma(Q_{35}), i(Q_{36}), \gamma(Q_{36}) \in \{18, 19\}$". Their 19-queen set for Q_{35} is independent. The only odd n for which $\gamma(Q_n) = \lceil n/2 \rceil + 1$ was known is $n = 15$, where $\gamma(Q_{15}) = 9$ [14]. We settle $n = 35$.

Theorem 1.1. *No* 18 *queens dominate the* 35×35 *board. Hence* $\gamma(Q_{35}) = i(Q_{35}) = 19$.

Corollary 1.2. $n = 35$ *is the smallest odd* $n > 15$ *with* $\gamma(Q_n) = \lceil n/2 \rceil + 1$.

Proof. For odd n with $17 \leq n \leq 33$ the value $\gamma(Q_n) = \lceil n/2 \rceil$ is in the list of [20]. □

The upper bounds are explicit sets, the independent one from [20]. The lower bound is computer-assisted (Sections 2–3): we prove that a dominating set of 18 queens would satisfy the constraints of one of 128,941 explicit cases, and refute every case by SAT solving with a DRAT proof checked by `drat-trim` [27]. The reduction is formalised in Lean 4 (Section 7), and a second, separately written encoding solved with a different SAT solver cross-checks the cases that are hardest for the first one (Section 6).

The frame method and prior work. Our reduction splits the search space by the *box* spanned by the outermost empty rows and columns of a dominating set, and by the *deficits*, the numbers of queens in excess of the occupied lines of each of the four kinds. The border of the box and its sides obey exact counting identities, which make the case formulas small and strong. The box split, with the two side identities as "counting hints", is the method of the preprint *Hint and Conquer* [11] (October 2026), which used it to establish $\gamma(Q_n) = n/2{+}1$ for $n = 26, 28, 32, 34, 36, 38$. Exact border identities of this kind go back to the counting argument behind the bound $\gamma(Q_n) \geq (n-1)/2$ [21, 6]: Weakley [26, proof of Theorem 1] states an exact border identity for the box (the box and the border inequality appear in the work of Bozóki, Gál, Marosi and Weakley [5]), Neuhaus [17, Ch. 4] analysed boxes, their sides and "loss squares" of dominating sets of size $n/2$ by hand, and Karandikar and Dutta [13] used counts on the sides of the box, with a split into square and non-square boxes, for odd boards. What we add to the method of [11] is an explicit split by the deficit vector with exact cardinalities and at-most-one constraints, the whole-border identity, a residual symmetry reduction for the hardest cases, and its application to an odd board, where the identities have more slack (Section 2). [11] treats even n only.

Even boards. With the same method and DRAT proofs we determine four further even values (Section 4).

Theorem 1.3 (claimed in [17])**.** $\gamma(Q_{40}) = 21$, $\gamma(Q_{42}) = 22$, $\gamma(Q_{44}) = 23$ *and* $\gamma(Q_{46}) = 24$.

Before this work, the refereed literature gave only $\gamma(Q_n) \in \{n/2, n/2+1\}$ for these n [20, 26], and *Hint and Conquer* [11] covers even $n \le 38$. Neuhaus's 2009 dissertation [17] claims $\gamma(Q_n) = n/2+1$ for $n = 40, 42, 44, 46$ on the basis of an integer-programming computation (CPLEX) without certificates. Theorem 1.3 confirms these claims; as far as we know, it gives the first certified values. We also confirm the known even values.

Theorem 1.4 ([11]; claimed in [17])**.** *$\gamma(Q_{26}) = 14$, $\gamma(Q_{28}) = 15$, $\gamma(Q_{32}) = 17$, $\gamma(Q_{34}) = 18$ and $\gamma(Q_{36}) = 19$.*

These values were first established in *Hint and Conquer* [11], together with $\gamma(Q_{38}) = 20$, by SAT solving with DRAT proofs. They had been claimed earlier in Neuhaus's 2009 dissertation [17], which states $\gamma(Q_n) = i(Q_n) = n/2 + 1$ for $n = 20, 22, \ldots, 28$ and $n = 32, \ldots, 46$ on the basis of an integer-programming computation (CPLEX) for which no run times, logs or certificates are given for $n \ge 20$. For $n = 26$, the repository `jkolantree/BSC` on GitHub [12] contains a Lean 4 development whose final theorem states $\gamma(Q_{26}) = 14$; its proof, committed in August 2026, uses parity "shadows" of a 13-queen set, polynomial (Nullstellensatz and moment) identities and a case analysis of monochromatic sets, and the repository reports an external re-check of the Lean statements by a second kernel. We have not built or checked it. An earlier version of the present paper overlooked [11] and [17].

For $n = 26$ and $n = 28$ we give, in addition, a second proof that is independent of the box method in reduction and in code (Section 8). It relaxes the grid of empty rows and empty columns to covers by diagonals of each colour, enumerates the feasible grids, and then places the queens; both stages carry DRAT certificates, and every certificate formula has been rebuilt from the survivor lists, solved and checked again.

For reference we collect the neighbouring known values.

Proposition 1.5 (known values)**.** *The terms $a(n) = \gamma(Q_n)$ of OEIS A075458 for $26 \le n \le 46$ are*

$$a(26) = a(27) = 14, \quad a(28) = a(29) = a(30) = 15, \quad a(31) = 16, \quad a(32) = a(33) = 17,$$
$$a(34) = 18, \quad a(35) = a(36) = a(37) = 19, \quad a(38) = a(39) = 20, \quad a(40) = a(41) = 21,$$
$$a(42) = a(43) = 22, \quad a(44) = a(45) = 23, \quad a(46) = 24.$$

Here $a(26)$, $a(28)$, $a(32)$, $a(34)$, $a(36)$ and $a(38)$ are due to [11] (Theorem 1.4), $a(35)$ is Theorem 1.1 and $a(40)$, $a(42)$, $a(44)$, $a(46)$ are Theorem 1.3. The values $a(27)$, $a(29)$, $a(30)$, $a(31)$, $a(33)$, $a(37)$, $a(39)$, $a(41)$ and $a(45)$ equal $\lceil n/2 \rceil$ and are listed by Östergård and Weakley [20]. For $n = 43$, [20] gives $\{22, 23\}$; the 22-queen dominating set of Q_{43} in [17] and the lower bound $\lceil n/2 \rceil$ [20, 8] give $a(43) = 22$. So $\gamma(Q_n)$ is known for every $n \le 46$.

Checks. By Corollary 2 of Östergård and Weakley [20], $\gamma(Q_n) \ge n/2$ for $n < 143$, $n \ne 3, 11$; this gives the lower bounds 14, 15, 15, 16 for $n = 27, 29, 30, 31$. For the upper bounds we check explicit sets directly in our verification script. For $n = 27$ we use the 14-set of Östergård and Weakley [20, §5]: in centred coordinates $-13, \ldots, 13$ it consists of the 13 squares (x, y) with $x = -12, -10, \ldots, 12$ and $y = 6, -12, -6, 8, 0, -8, 10, -2, 4, 12, -10, -4, 2$ respectively, together with $(-9, -9)$. For $n = 30$ we use the monochromatic 15-set given by Weakley [26, proof of Proposition 21]: in coordinates $-14, \ldots, 15$, the squares (x, y) with $x = -13, -11, \ldots, 15$ and $y = 1, -9, 9, 15, -11, -5, 3, -3, 13, 11, 7, -13, -7, -1, 5$ respectively. None of its squares lies in the row or the column with coordinate -14; deleting them leaves a 29×29 board on which the same 15 squares still dominate, so $\gamma(Q_{29}) \le 15$. Adding a 31st row and column and a queen at their intersection gives a 16-set dominating Q_{31}. The sets for $n = 32, \ldots, 36$ are given in Sections 3 and 4, and the verification script also checks the 22-set of Q_{43} of [17]. We did not check the values for $n = 33, 37, 38, 39, 41, 45$. □

Section 2 sets up the frame method with complete proofs, and Section 3 reports the computation for $n = 35$. Section 4 gives the even boards, Section 5 the validation of the method, Section 6 the cross-check for $n = 35$ and Section 7 the Lean formalisation. Section 8 is the second proof for $n = 26, 28$. The code, the records of all computations, the Lean sources and a verification script are available online.[1]

2 The frame method

2.1 Boxes, needy squares and deficits

In this part of the paper a square is a pair (x, y) with $0 \le x, y < n$, where x is the column and y the row. The *difference diagonal* of (x, y) is $t = y - x$ and its *sum diagonal* is $s = x + y$; a *line* is a row, a column or a diagonal of either kind. Let D be a set of exactly k squares (queens), and let R, C, T, S be the sets of rows, columns, difference diagonals and sum diagonals that contain a queen. The *deficits*

$$d_R = k - |R|, \qquad d_C = k - |C|, \qquad d_T = k - |T|, \qquad d_S = k - |S|$$

are non-negative; for instance $d_T = 0$ means that no two queens share a difference diagonal. A row or column is *empty* if it contains no queen, and a square is *needy* if its row and its column are both empty. A needy square is not in D and is not attacked along its row or column, so:

(N) if D dominates the board, every needy square lies on an occupied diagonal.

Assume that at least two rows and two columns are empty; this holds if $k \le n-2$. Let $a < b$ be the least and the greatest empty column and $c < d$ the least and the greatest empty row, and put $w = b-a \ge 1$, $h = d-c \ge 1$. The *box* is $[a, b]\times[c, d]$. Its *border* is the set of its squares with $x \in \{a, b\}$ or $y \in \{c, d\}$, each counted once; the *vertical sides* are the $2(h + 1)$ border squares with $x \in \{a, b\}$, and the *horizontal sides* the $2(w + 1)$ border squares with $y \in \{c, d\}$, so the four corners belong to both. The *frame* of D is the quadruple of margins $(m_l, m_r, m_b, m_t) = (a,\ n - 1 - b,\ c,\ n - 1 - d)$.

Lemma 2.1. *Let D be as above.*

(E1) Every empty column lies in $[a, b]$ and every empty row in $[c, d]$. Hence $m_l + m_r \le k - d_C$ and $m_b + m_t \le k - d_R$.

(E2) No border square is occupied.

(E3) A square of a horizontal side is needy if and only if its column is empty, and a square of a vertical side if and only if its row is empty. Each horizontal side contains exactly $n - |C|$ needy squares, each vertical side exactly $n-|R|$, the corners are needy, and the border contains exactly $N = 2(n - |R|) + 2(n - |C|) - 4$ needy squares.

(E4) A diagonal meets the border in at most two squares.

(E5) Suppose $w \ge h$. A diagonal contains at most one square of the vertical sides, except when $w = h$: then the difference diagonal $u_0 = \{t = c - a\}$ contains the corners (a, c), (b, d) and the sum diagonal $v_0 = \{s = a + d\}$ contains the corners (a, d), (b, c). Symmetrically, if $h \ge w$ a diagonal contains at most one square of the horizontal sides, with the same two exceptions when $w = h$.

[1] https://github.com/jasondavies/queen-domination

Proof. (E1) holds because a, b, c, d are extreme; the $m_l + m_r$ columns outside $[a, b]$ are occupied, so their number is at most $|C| = k - d_C$, and likewise for rows. (E2) A border square lies in column a or b or in row c or d, all empty. (E3) The row of a square of a horizontal side is c or d, which is empty, so the square is needy if and only if its column is. By (E1) all $n - |C|$ empty columns lie in $[a, b]$, so each horizontal side has exactly $n - |C|$ needy squares; the vertical sides are analogous. Adding the four sides counts the four (needy) corners twice. (E4) Parametrise a difference diagonal as $(x_0 + j, y_0 + j)$. Its squares in the box are those with j in an interval $[j_1, j_2]$, and a square is on the border only if one of its coordinates attains a bound of the box; the lower bounds can only be attained at $j = j_1$ and the upper bounds only at $j = j_2$, since both coordinates increase with j. A sum diagonal $(x_0 + j, y_0 - j)$ is handled in the same way. (E5) A diagonal has one square in each column it meets. If it contains (a, y) and (b, y') with $c \le y, y' \le d$, then $|y - y'| = w \le h$, so $w = h$ and $\{y, y'\} = \{c, d\}$, which gives the four corners as stated. □

2.2 Three identities

For a set X of border squares, a diagonal L and a square q, let $\beta_X(L)$ be the number of squares of X on L, let $m(q) \in \{0, 1, 2\}$ be the number of occupied diagonals through q, and let U_X be the number of needy squares $q \in X$ with $m(q) = 0$; by (N), $U_X = 0$ if D dominates.

Theorem 2.2. *Let D be a set of k squares with at least two empty rows and two empty columns.*

(F) With X the border,

$$\sum_L \bigl(2 - \beta_X(L)\bigr) + \#\{(L, q) : q \in X \text{ not needy},\ q \in L\} + \#\{q \in X \text{ needy} : m(q) = 2\} \\ = 8k - 4n + 4 - 2(d_R + d_C + d_T + d_S) + U_X,$$

where L runs over the occupied diagonals.

(V) If $w \ge h$, then with X the vertical sides,

$$\#\{L : \beta_X(L) = 0\} + \#\{(L, q) : q \in X \text{ not needy},\ q \in L\} + \#\{q \in X \text{ needy} : m(q) = 2\} \\ = 4k - 2n - d_T - d_S - 2d_R + [w = h]\,\sigma + U_X,$$

where L runs over the occupied diagonals and $\sigma \in \{0, 1, 2\}$ is the number of occupied diagonals among u_0, v_0.

(H) If $h \ge w$, the same holds for the horizontal sides with d_C in place of d_R.

All terms on the left-hand sides are non-negative.

Proof. (F) Count the incidences I between occupied diagonals and border squares. By diagonals, $I = \sum_L \beta_X(L) = 2(|T| + |S|) - \sum_L (2 - \beta_X(L))$, and each term $2 - \beta_X(L)$ is non-negative by (E4). By squares, $I = \sum_{q \in X} m(q)$. A needy square contributes $m(q) = 1 + [m(q) = 2] - [m(q) = 0]$, and the non-needy squares contribute the number of pairs (L, q) with q not needy. With (E3), $I = N + \#\{q \text{ needy} : m(q) = 2\} - U_X + \#\{(L, q) : q \text{ not needy}\}$. Equating the two expressions and substituting $|T| = k - d_T$, $|S| = k - d_S$, $n - |R| = n - k + d_R$ and $n - |C| = n - k + d_C$ gives (F). (V) By (E5), $\beta_X(L) \in \{0, 1\}$ for every diagonal except $\beta_X(u_0) = \beta_X(v_0) = 2$ when $w = h$. So counting by diagonals gives $I = |T| + |S| - \#\{L : \beta_X(L) = 0\} + [w = h]\sigma$, and counting by squares, with the $2(n - |R|)$ needy squares of the vertical sides from (E3), gives $I = 2(n - |R|) + \#\{q \text{ needy} : m(q) = 2\} - U_X + \#\{(L, q) : q \text{ not needy}\}$. (H) is (V) for the transposed set. □

For a dominating set the right-hand sides are therefore non-negative, which gives

$$d_R + d_C + d_T + d_S \le 4k - 2n + 2,$$
$$d_T + d_S + 2d_R \le 4k - 2n + 2[w = h] \text{ if } w \ge h, \quad d_T + d_S + 2d_C \le 4k - 2n + 2[w = h] \text{ if } h \ge w.$$

For even n and $k = n/2$ the slack $4k - 2n$ is 0: the deficits sum to at most 2, and if $w > h$ then $d_R = d_T = d_S = 0$ and all terms of (V) vanish, so every occupied diagonal passes through exactly one needy square of the vertical sides, and no needy square of the vertical sides lies on two occupied diagonals. For odd n and $k = (n + 1)/2$, the case of Theorem 1.1, the slack is 2: the deficits sum to at most 4, and for $w > h$ at most $2 - d_T - d_S - 2d_R$ units of waste are allowed on the vertical sides. This weaker rigidity is what makes odd boards harder. For odd n and $k = (n - 1)/2$ the right-hand side of (V) is negative unless $w = h$ and both box diagonals are occupied, and (F) forces all deficits and all waste to vanish; this is the configuration of the classical analysis of the cases $n = 3, 11$ (compare [13, Theorem 5.5]).

2.3 Cases and their completeness

A *case* for (n, k) is a frame together with a deficit vector, $c = (m_l, m_r, m_b, m_t; d_R, d_C, d_T, d_S)$. Every set of k squares with at least two empty rows and columns has exactly one case. The eight symmetries of the board act on cases: the reflection $x \mapsto n - 1 - x$ exchanges m_l with m_r and d_T with d_S; the reflection $y \mapsto n - 1 - y$ exchanges m_b with m_t and d_T with d_S; the transposition exchanges (m_l, m_r) with (m_b, m_t) and d_R with d_C. A frame is *canonical* if it is lexicographically least among its images. A case is *admissible* if each deficit is at most $k - 1$, $m_l + m_r \le k - d_C$, $m_b + m_t \le k - d_R$, $w, h \ge 1$, and the three inequalities displayed after Theorem 2.2 hold. Let $\mathcal{L}(n, k)$ be the list of all admissible cases with a canonical frame. For $(n, k) = (35, 18)$ it has 5050 frames and 128,941 cases.

Proposition 2.3. *Let $1 \le k \le n - 2$, and suppose that for every case $c \in \mathcal{L}(n, k)$ no dominating set of k queens has case c. Then $\gamma(Q_n) > k$.*

Proof. Let D dominate with $|D| \le k$; adding squares, we may assume $|D| = k$. Then at least $n - k \ge 2$ rows and columns are empty. Choose a symmetry g for which the frame of $g(D)$ is canonical. The set $g(D)$ dominates, it has an admissible case by Lemma 2.1(E1) and Theorem 2.2 (each deficit is at most $k - 1$ because some line of each kind is occupied), and that case lies in $\mathcal{L}(n, k)$. □

The list contains every admissible deficit vector for every canonical frame; we do not reduce deficit vectors by the stabiliser of the frame, so no symmetry argument beyond frames is needed. For odd n and $k = (n + 1)/2$, 58 different deficit vectors occur (for even n and $k = n/2$, at most nine per frame), which is one reason why $n = 35$ has more than seven times as many cases as $n = 36$.

2.4 The formulas

For a case c, the formula $\Phi(c)$ has a variable $q_{x,y}$ for each square and occupancy variables for all lines, and consists of the following clauses.

DEF Each occupancy variable is equivalent to the disjunction of the queen variables on its line.

DOM For each square, its row, its column or one of its diagonals is occupied.

CARD Exactly $k-d_R$ rows, $k-d_C$ columns, $k-d_T$ difference diagonals and $k-d_S$ sum diagonals are occupied, and exactly k queens are placed (this last constraint is omitted when some deficit is 0, where it is implied).

AMO For each kind of line with deficit 0, no two queens on a line of that kind.

FRAME The columns a, b and the rows c, d are empty, and all columns outside $[a, b]$ and all rows outside $[c, d]$ are occupied.

(F), (V), (H) The left-hand sides of the identities of Theorem 2.2, written as sums of "waste literals", are at most the right-hand sides with $U_X = 0$; (V) only if $w \geq h$ and (H) only if $h \geq w$. A diagonal L contributes $2 - \beta_X(L)$ copies of its occupancy literal; a non-needy pair (L, q) is an auxiliary variable implied by the occupancy of L and the non-neediness of q, which on a side of the box is a single line literal by (E3); a needy square on two occupied diagonals is an auxiliary variable implied by the two occupancies. For $w = h$ the side constraint reads $(\text{waste}) + [\neg u_0] + [\neg v_0] \leq 4k - 2n - d_T - d_S - 2d_R + 2$.

Cardinality constraints use the sequential counter [23], implemented in about twenty lines.

Lemma 2.4. *If a dominating set D of exactly k queens, $k \leq n-2$, has case c, then $\Phi(c)$ is satisfiable.*

Proof. Give every variable its intended meaning: q the indicator of D, occupancy variables the occupancy of lines, auxiliary variables the conjunctions they stand for, and counter variables $s_{i,j}$ ="at least $j+1$ of the first $i+1$ literals are true". DEF holds by definition and DOM because D dominates; CARD and AMO hold by the definition of the deficits, FRAME by (E1), and the waste constraints by Theorem 2.2 with $U_X = 0$ (on the border, non-neediness is expressed correctly by (E2), (E3)). □

By Proposition 2.3 and Lemma 2.4, if $\Phi(c)$ is unsatisfiable for every $c \in \mathcal{L}(n, k)$, then $\gamma(Q_n) > k$.

2.5 Residual symmetry and cubes

Some cases are fixed by non-trivial symmetries of the board. For a case c let G_c be the set of symmetries g with $g(c) = c$, and let $v(D)$ be the vector of row and column occupancies $(o_R[0..n-1], o_C[0..n-1])$. For each symmetry g there is a fixed permutation π_g of coordinates with $v(g(D)) = \pi_g(v(D))$. Let $\mathrm{SBP}(c)$ be the lex-leader constraints $v \leq_{\mathrm{lex}} \pi_g(v)$ for $g \in G_c$.

Lemma 2.5. *If some dominating set of k queens has case c, then $\Phi(c) \wedge \mathrm{SBP}(c)$ is satisfiable.*

Proof. If D has case c and $g \in G_c$, then $g(D)$ is a dominating k-set with case $g(c) = c$. Choose $g \in G_c$ with $v(g(D))$ lexicographically least and put $D' = g(D)$. For $h \in G_c$, $\pi_h(v(D')) = v(hg(D)) \geq v(D')$ because $hg \in G_c$. By Lemma 2.4 applied to D', with the chain variables of the lex-leader constraints set accordingly, the formula is satisfied. □

So Proposition 2.3 remains valid if $\Phi(c) \wedge \mathrm{SBP}(c)$ is used in place of $\Phi(c)$ for any subset of the cases. For a case with box $[a, b] \times [c, d]$, *layer* j consists of the four occupancy variables of the rows $c+j$, $d-j$ and the columns $a+j$, $b-j$. The 16 assignments of a layer are exhaustive, so a formula is unsatisfiable if it is unsatisfiable under each of them. Our driver solves a hard case with SBP as a whole and replaces a *cube* (a partial assignment of layers $1, \ldots, j$) that exceeds the time limit by its 16 extensions to layer $j+1$; a separate check verifies from the records alone that the refuted cubes of each case form such a cover.

3 The computation for $n = 35$

Upper bound. The 18-queen dominating set of Q_{34} of Section 4 together with the corner $(34, 34)$ is a 19-queen dominating set of Q_{35}: $(1, 19)$, $(3, 7)$, $(5, 27)$, $(7, 9)$, $(9, 17)$, $(11, 1)$, $(13, 23)$, $(15, 15)$, $(17, 31)$, $(19, 5)$, $(21, 13)$, $(23, 21)$, $(25, 3)$, $(27, 33)$, $(29, 11)$, $(31, 25)$, $(32, 32)$, $(33, 29)$, $(34, 34)$, as (column, row). It is not independent. The 19-set of Östergård and Weakley [20, §5] is: in centred coordinates $-17, \ldots, 17$, the squares (x, y) with $x = -17, -15, \ldots, 17$ and $y = 3, -17, -9, 13, 7, -11, 1, 9, -7, 11, -13, -5, 15, -15, -1, 5, 17, -3$, together with $(2, 2)$. Our verification script checks that both dominate Q_{35} and that the second is independent, so $i(Q_{35}) \leq 19$.

Lower bound. The list $\mathcal{L}(35, 18)$ has 5050 canonical frames and 128,941 cases. Every formula $\Phi(c)$ was written as a DIMACS file, its SHA-256 hash recorded, and CaDiCaL 3.1.0 [2] (default options) run on it with a DRAT proof, which `drat-trim` checked in backward mode; the proofs were deleted after checking. The run had three phases (Table 1). The non-square boxes were run first, then the square boxes with a positive deficit, each case with a time limit of 900 s. The remaining 585 cases, all with deficit vector 0 (so that the 18 queens would be pairwise non-attacking), are the square boxes and those with $|w - h| = 1$ and small margins; they were run with residual symmetry breaking (Lemma 2.5) and adaptive cube splitting with a limit of 1800 s per cube. Of these, 508 were refuted as a whole and 77 by covers of cubes, 76 of them by the 16 cubes of layer 1 and 1 with a second layer, using 1755 refuted cubes in all. The records also contain 8 time-outs: seven cases that timed out in an earlier pass with a 300 s limit and were then run in phase 3, and one cube that was replaced by its 16 children.

phase	cases	CaDiCaL (CPU h)	`drat-trim` (CPU h)
1: non-square boxes	115,601	16.4	9.3
2: square boxes, positive deficit	12,755	15.2	7.6
3: deficit 0, $\lvert w - h\rvert \leq 1$ (SBP, cubes)	585	21.5	10.7
total	128,941	53.1	27.6

Table 1: The computation for $n = 35$, $k = 18$: every case refuted, every DRAT proof verified. Single-core CPU hours on a shared machine, up to eight processes.

Every case of $\mathcal{L}(35, 18)$ is refuted, by a plain record or by a complete cover of cubes, all UNSAT with verified proofs; there is no satisfiable case. The proofs totalled about 232 GB, the largest 1.25 GB. The slowest plain case took 368 s to solve, and the slowest case in all, solving and checking all its cubes, 3804 s. At the end of the run every one of the 130,111 refuting formulas was rebuilt from the case by the encoder and its hash compared with the recorded one; the verification script repeats this for a seeded sample of 2000 formulas by default and for all of them on request, re-solves a sample of cases with proofs, and checks that the set of cases in the records equals $\mathcal{L}(35, 18)$, rebuilt from first principles.

What the proof rests on. Theorem 1.1 follows from Proposition 2.3, Lemmas 2.4 and 2.5, the cover argument for cubes, and the explicit sets above. Its correctness depends on the proofs of Section 2; on the enumeration of $\mathcal{L}(n, k)$ (about 60 lines of Python, `code/frame/geom.py`); on the encoder (`code/frame/enc.py`, about 190 lines including the sequential counter) and the lex-leader clauses (`sbp_clauses` in `code/frame/hard_test.py`); on the coverage check; and on `drat-trim`. It does not depend on CaDiCaL, on any colour argument or on any structure theorem

from the literature. The reduction (Section 2) and the completeness of the list are formalised in Lean (Section 7); the encoder and the SAT certificates are not.

4 Even boards

We ran the same code, with DRAT proofs, for $k = n/2$ and $n = 26, 28, 32, 34, 36, 40, 42, 44, 46$, and for $(n, k) = (35, 17)$ (Table 2). Every case is unsatisfiable and every proof was verified; no case timed out at the limit of 1800 s, so no case needed symmetry breaking or cubes, and the result for these boards rests only on Proposition 2.3 and Lemma 2.4. The encoder and the enumeration are the files of the $n = 35$ run, unchanged. Together with explicit sets of $n/2 + 1$ queens this proves Theorems 1.4 and 1.3. The case $(35, 17)$ reproduces the known bound $\gamma(Q_{35}) \geq 18$. The numbers of canonical frames for $n = 26, 28, 32, 34, 36$ equal the numbers of boxes reported in [11, Table 1]. For $n = 40, \ldots, 46$ the case list was also rebuilt by a separate enumeration that maps boxes geometrically and checks the number of frames by Burnside's lemma; the lists agree. As for $n = 35$, the verification script rebuilds each list, checks that every case has exactly one refuting record, rebuilds the formulas of a sample of records and compares their hashes.

n	k	frames	cases	CaDiCaL (s)	`drat-trim` (s)	proofs (GB)	slowest case (s)
26	13	1596	5663	413	549	0.59	17.8
28	14	2080	7300	652	787	0.93	17.8
32	16	3321	11 466	1597	1559	2.12	29.6
34	17	4095	14 049	2548	2193	3.06	81.1
36	18	5050	17 205	3604	2927	4.32	57.5
40	20	7381	24 871	7892	5395	8.15	84.8
42	21	8778	29 447	11 299	7154	10.85	132.7
44	22	10 440	34 854	13 243	8772	14.46	119.1
46	23	12 246	40 728	10 009	7724	18.55	135.3
35	17	330	330	13	36	0.01	0.1

Table 2: The frame method on further boards: every case UNSAT with a verified DRAT proof. Single-core CPU seconds on a shared machine, four to six processes; the times depend on the load (the run for $n = 46$ had the machine almost to itself).

The upper bounds come from sets found by a SAT search for a permutation-like placement on the odd lines plus one queen near a corner, checked directly, as (column, row):

- Q_{32}, 17 queens: $(1, 23)$, $(3, 11)$, $(5, 17)$, $(7, 27)$, $(9, 1)$, $(11, 7)$, $(13, 15)$, $(15, 31)$, $(17, 21)$, $(19, 25)$, $(21, 9)$, $(23, 3)$, $(25, 19)$, $(27, 5)$, $(29, 13)$, $(30, 30)$, $(31, 29)$;
- Q_{34}, 18 queens: $(1, 19)$, $(3, 7)$, $(5, 27)$, $(7, 9)$, $(9, 17)$, $(11, 1)$, $(13, 23)$, $(15, 15)$, $(17, 31)$, $(19, 5)$, $(21, 13)$, $(23, 21)$, $(25, 3)$, $(27, 33)$, $(29, 11)$, $(31, 25)$, $(32, 32)$, $(33, 29)$;
- Q_{36}, 19 queens: $(1, 27)$, $(3, 13)$, $(5, 9)$, $(7, 25)$, $(9, 3)$, $(11, 19)$, $(13, 35)$, $(15, 5)$, $(17, 31)$, $(19, 15)$, $(21, 23)$, $(23, 1)$, $(25, 11)$, $(27, 33)$, $(29, 21)$, $(31, 29)$, $(33, 7)$, $(34, 34)$, $(35, 17)$;
- Q_{40}, 21 queens: $(1, 17)$, $(3, 27)$, $(5, 33)$, $(7, 9)$, $(9, 5)$, $(11, 23)$, $(13, 19)$, $(15, 25)$, $(17, 37)$, $(19, 3)$, $(21, 15)$, $(23, 11)$, $(25, 1)$, $(27, 31)$, $(29, 21)$, $(31, 39)$, $(33, 13)$, $(35, 7)$, $(37, 35)$, $(38, 38)$, $(39, 29)$;
- Q_{42}, 22 queens: $(1, 23)$, $(3, 29)$, $(5, 11)$, $(7, 21)$, $(9, 39)$, $(11, 1)$, $(13, 7)$, $(15, 19)$, $(17, 25)$, $(19, 37)$, $(21, 17)$, $(23, 5)$, $(25, 35)$, $(27, 13)$, $(29, 27)$, $(31, 33)$, $(33, 3)$, $(35, 9)$, $(37, 15)$, $(39, 31)$, $(40, 40)$, $(41, 41)$;

- Q_{46}, 24 queens: $(0,0)$, $(1,11)$, $(3,25)$, $(5,19)$, $(7,35)$, $(9,39)$, $(11,9)$, $(13,31)$, $(15,1)$, $(17,43)$, $(19,21)$, $(21,17)$, $(23,41)$, $(25,29)$, $(27,5)$, $(29,23)$, $(31,37)$, $(33,3)$, $(35,45)$, $(37,13)$, $(39,7)$, $(41,15)$, $(43,33)$, $(45,27)$.

For Q_{44} we take Neuhaus's 22-queen dominating set of Q_{43} (a "0-1-cover", [17, App. B]; a file of Neuhaus's solutions repository [18], licensed CC BY 4.0 and included unchanged in our repository) and add the corner $(43,43)$, which covers the new row and column; the script checks both sets directly. For $n = 26, 28$ the sets of Section 8 are used. Upper bounds $\gamma(Q_n) \le n/2 + 1$ for these n are also in [20, 26].

5 Validation of the frame method

The certificates make the result independent of the solver, but not of the case list, the encoder or the proofs of Section 2. These were tested as follows; the verification script reruns or rechecks each item.

- *Identities.* Theorems 2.2(F), (V), (H), in the general form with the term U_X, and the facts (E3)–(E5) were checked by direct counting on every set of at most n squares with a frame for $n \le 5$ and on random and random dominating sets up to $n = 16$: 82 466 sets, 29 203 of them with $w = 1$ or $h = 1$. On random sets, the number of true waste literals of the encoder equals the corresponding left-hand side.

- *Exact solution sets.* For nine pairs (n,k) up to $(10,5)$, including $k > n/2$ and odd n, all models of all case formulas, closed under the symmetries, are exactly the dominating k-sets found by a plain enumeration with no frames and no identities. This tests the case list, the symmetry action, Proposition 2.3 and the encoder together.

- *Known values.* For $k = n/2$ and every even $n \le 30$ the formulas of all cases are satisfiable for some case exactly when $n \in \{4, 6, 10, 12, 18, 30\}$, that is, when $\gamma(Q_n) = n/2$. For $k = (n-1)/2$ and every odd $n \le 31$ they are satisfiable exactly for $n = 3, 11$. For $k = (n+1)/2$ they are satisfiable for $n = 5, 7, 9, 13, 17, 19, 21$ and unsatisfiable for $n = 15$, where $\gamma(Q_{15}) = 9$; the runs for $n = 5, 7, 9, 13, 15$ are complete. Every model is checked to be a dominating set with the case it was found for. The only negative control among odd boards with $k = (n+1)/2$ is therefore $n = 15$.

- *Residual symmetry breaking.* For all 3 566 cases with a non-trivial stabiliser for the eight pairs (n,k) from $(4,2)$ to $(10,5)$, the models of $\Phi \wedge \mathrm{SBP}$, closed under the stabiliser, are exactly the models of Φ.

- *Second encoding.* For $n = 32$, a second encoding written separately (totalizers, ladder at-most-one constraints, identities as equalities; `code/frame/enc_b.py`) solved with Glucose 4, and the first encoding solved with CaDiCaL 1.9.5, give UNSAT on all 11 466 cases.

- *Planted positives.* Thirty-five known dominating sets, among them Kamenetsky's best known sets for $n \le 26$, sets of $Q_{27}, \ldots, Q_{31}$, two 17-sets of Q_{32} and our sets of Q_{34} and Q_{36} (at $k = n/2 + 1$), are accepted by the formula of their case in every orientation with a canonical frame, and a non-dominating variant of each is rejected (recorded; not rerun by the script).

6 An independent cross-check for $n = 35$

Because the negative controls for odd boards are few, the $n = 35$ result was cross-checked with a second pipeline, written without reading the first encoder; it shares only the definitions of Section 2.

- *Case list.* A separate enumeration of frames and admissible cases computes the action of the symmetries on cases geometrically rather than from a table. For $(35, 18)$ it gives the same 128,941 cases.
- *Encoding X.* Cardinalities by odd-even merge sorting networks restricted to the needed outputs [1], at-most-one constraints by the product encoding, "exactly k queens" always, neediness by explicit conjunction variables, and all auxiliary variables fully defined. The identities are imposed as equalities in a form derived by a second double count: for a set X of border squares and $\mathrm{cap} \in \{1, 2\}$, $\sum_L o(L)\,(\mathrm{cap} - \nu_X(L)) + \#\{q \in X \text{ needy} : m(q) = 2\} = \mathrm{cap}(2k - d_T - d_S) - N_X$, where $\nu_X(L)$ counts needy squares of X on L and N_X is fixed by the case. Diagonals that meet X more than cap times are encoded exactly, so this form needs neither (E4) nor (E5), and (V) and (H) are imposed for every box shape.
- *Solver.* Kissat 4.0.4 [3], default options, without proofs, and *no symmetry breaking at all.* A formula that exceeds 4000 s is split into 16 cubes on the occupancy of four diagonals (those through the corners of the box first), recursively; a separate script checks from the records that every tree is complete.

Controls. Encoding X reproduces the exact solution sets of a plain enumeration for ten pairs (n, k) up to $(11, 5)$, and accepts 32 known dominating sets in every orientation with a canonical frame (recorded). In complete runs it refutes every case for $k = (n-1)/2$ and every odd $n = 5, \ldots, 35$ except for one satisfiable case at $n = 11$ (the known 5-set), all 330 cases of $(35, 17)$ included; for $(13, 7)$ it finds exactly the 26 satisfiable cases of the main pipeline among 5103; and for $(15, 8)$ it refutes all 7916 cases. On the zero-deficit cases of $(13, 7)$, $(15, 8)$, $(17, 9)$ and $(19, 10)$ (the type of the hardest $n = 35$ cases), the main encoder with CaDiCaL 1.9.5 and encoding X with Kissat give the same verdict on every case.

The hard cases and a sample. The cross-check was run on all 585 cases of phase 3, which the main computation refuted only with symmetry breaking or cubes, and on a stratified sample of 2200 of the other cases (2000 uniformly random and the 200 slowest in the main run). Every one of the 585 hard cases is unsatisfiable under encoding X: 14 of them needed cube splitting, and their trees have 795 unsatisfiable leaves in all; the formulas took 76.5 CPU hours. All 2200 sampled cases are unsatisfiable without splitting (50,142 CPU seconds). There is no satisfiable formula and no disagreement with the main computation. The verification script checks from the records that every tree is complete, and rebuilds all 809 formulas of the hard trees and all 2200 of the sample and compares their hashes. Kissat produced no proofs here: the cross-check is independent evidence for the encoding and the case list, while the certificates of Section 3 remain the proof.

7 Formalisation in Lean

The reduction of Section 2 is formalised in Lean 4 [7] with Mathlib [16] (directory `lean/`, Lean and Mathlib v4.35.0). Squares are elements of $\mathrm{Fin}\, n \times \mathrm{Fin}\, n$, domination is defined by attacks along lines and shown to be domination in the queen's graph (a `SimpleGraph`), and $\gamma(Q_n)$ is the least size of a dominating set.

- Lemma 2.1, Theorem 2.2 (as exact identities in $\mathbb{Z}$, for all n and all sets with a frame, dominating or not), the consequences for $k = n/2$ and for odd n, the action of the symmetries on frames and deficits, Proposition 2.3, the semantic part of Lemma 2.4 and Lemma 2.5 are proved for general n and $k \le n-2$. The semantic content of $\Phi(c)$ is a structure `CaseSat n k c D` with one field per clause group.
- The enumeration of $\mathcal{L}(n,k)$ is a Lean function, proved to list exactly the admissible cases with canonical frame. For $(32,16)$ and $(35,18)$ it is shown, by 343 kernel computations (`decide +kernel`, one per pair of horizontal margins), to equal the explicit lists of 11 466 and 128,941 cases of the runs.
- The 19-queen set dominates Q_{35} and the 17-queen set dominates Q_{32}.

The final theorem for $n = 35$, `gamma_35_eq_19_of_cases35_unsat_mixed`, states

$$\Big(\forall c \in \texttt{cases35}:\ \neg\exists D\ \texttt{CaseSat}\ 35\ 18\ c\ D\ \ \vee\ \neg\exists D\ \big(\texttt{CaseSat}\ 35\ 18\ c\ D \wedge \texttt{SymBreak}\ c\ D\big)\Big) \\ \Longrightarrow\ \texttt{gamma}\ 35 = 19,$$

where `cases35` is the list of the run and `SymBreak` the lex-leader condition; the analogous theorem for $n = 32$ has the hypothesis without `SymBreak`. All 116 theorems of the axiom report, and all kernel computations, depend only on `propext`, `Classical.choice` and `Quot.sound`; there is no `sorry` and no `native_decide`. The general theorems (`main_even`, `lt_gamma_of_cases_unsat`) apply to every even n, but the explicit case lists of $n = 26, 28, 34, 36, 40, \dots, 46$ are not in Lean; for those boards the identification of the run with $\mathcal{L}(n,k)$ rests on the Python enumeration, checked for $n = 40, \dots, 46$ by the separate enumeration of Section 4.

What is not formalised. The hypothesis `H` is what the computation establishes, by steps outside Lean: (i) that a set D with `CaseSat n k c D` yields a model of the CNF written by the encoder, i.e. the syntactic part of Lemma 2.4 (definitional variables, waste variables, sequential counters, at-most-one clauses), and the same for the lex-leader clauses and the exhaustiveness of cubes; (ii) that each CNF is unsatisfiable, which rests on the DRAT proofs and `drat-trim`, and the identification of each record with its formula by the CNF hash; (iii) that the Lean data file equals the set of refuted cases in the records, which a script checks (it also checks the list against the Python enumeration).

8 A second proof for $n = 26$ and $n = 28$: a line relaxation

This section gives a proof of $\gamma(Q_{26}) > 13$ and $\gamma(Q_{28}) > 14$ that is independent of the box method in its reduction and in its code. It keeps all the chosen empty rows and columns rather than only the outermost ones, forgets where the queens are, and records only which diagonals of each colour must be occupied to cover the grid of empty rows and columns. The notation of this section is separate from that of Sections 2–7.

8.1 The relaxation

Index rows and columns by $[0,n) = \{0,\dots,n-1\}$ and write (r,x) for the square in row r and column x. Its *colour* is $r + x \bmod 2$. Its *difference line* is $r - x$ and its *sum line* is $r + x$. Since $r - x \equiv r + x \pmod 2$, all squares of a line have the same colour, which we call the colour of the line. A queen on (r,x) lies on exactly one difference line and one sum line, both of its own colour, and attacks diagonally only squares of its own colour.

Let D be a dominating set of Q_n with $|D| = g$, and let z_0, z_1 be the numbers of queens of colour 0 and 1. Put $s = n - g$. Since D meets at most g rows and at most g columns, at least s rows and at least s columns contain no queen.

Definition 8.1. For $A, B \subseteq [0, n)$ and $q \in \{0, 1\}$ let $H_q(A, B)$ be the bipartite graph whose two vertex classes are the difference lines and the sum lines of colour q. Its edges are the squares $(r, x) \in A \times B$ of colour q, the square (r, x) joining the lines $r - x$ and $r + x$. For $z = (z_0, z_1)$ call (A, B) *z-feasible* if, for $q = 0, 1$, $H_q(A, B)$ has a vertex cover consisting of at most z_q difference lines and at most z_q sum lines.

Lemma 8.2. *If A is a set of empty rows and B a set of empty columns of D, then (A, B) is (z_0, z_1)-feasible.*

Proof. A square $(r, x) \in A \times B$ is unoccupied, and no queen shares its row or its column. Hence it is attacked along a diagonal, by a queen which then has the colour of (r, x). So the difference and sum lines of colour q that carry a queen cover all edges of $H_q(A, B)$. There are at most z_q of each kind, since each queen of colour q lies on one line of each kind and queens of colour $1 - q$ lie on no line of colour q. □

Feasibility is monotone: if $A' \subseteq A$ and $B' \subseteq B$, then $H_q(A', B')$ is a subgraph of $H_q(A, B)$, so every sub-pair of a feasible pair is feasible. A search that builds A and B element by element may therefore stop as soon as the partial pair is infeasible. Two certificates of infeasibility are used.

(M) *Matching bound.* A vertex cover of H_q has at least $\nu(H_q)$ vertices, where ν is the matching number. So $\nu(H_q(A, B)) > 2z_q$ implies infeasibility. A matching of $H_q(A, B)$ is a set of colour-q squares of $A \times B$, no two on a common diagonal.

(K) *Forced lines.* Distinct squares on one line lie on distinct lines of the other kind. So a line containing more than z_q squares of the graph belongs to every admissible cover, as otherwise more than z_q lines of the other kind would be needed. Removing forced lines, lowering the budgets and iterating, and then applying (M) to the remaining graph against the remaining total budget, gives a valid test.

Lemma 8.3 (symmetry)**.** *Let n be even and $\bar{X} = \{n - 1 - i : i \in X\}$.*

1. *The transpose $(r, x) \mapsto (x, r)$, the half-turn $(r, x) \mapsto (n-1-r, n-1-x)$ and the anti-transpose $(r, x) \mapsto (n-1-x, n-1-r)$ preserve colours. With the identity they form a group K of order 4. They map a dominating set with empty rows A and empty columns B to a dominating set with the same colour sizes and with empty rows and columns (B, A), $(\bar{A}, \bar{B})$ and $(\bar{B}, \bar{A})$ respectively.*

2. *z-feasibility of (A, B) is invariant under this action of K.*

3. *The reflection $(r, x) \mapsto (r, n - 1 - x)$ exchanges the two colours.*

Proof. (1) The colour of the image square is $x+r$, $2n-2-(r+x)$ and $2n-2-(r+x)$ respectively, all $\equiv r+x$. The statement about empty lines is immediate. (2) The three maps send difference lines to difference lines ($r-x \mapsto x-r$, $x-r$, $r-x$) and sum lines to sum lines ($r+x \mapsto r+x$, $2n-2-(r+x)$, $2n - 2 - (r + x)$), preserving colours. Hence they induce isomorphisms $H_q(A, B) \cong H_q(\sigma(A, B))$ which respect the two vertex classes, and admissible covers correspond. (3) The image of (r, x) has colour $r + n - 1 - x \equiv r + x + 1$ because n is even. □

Proposition 8.4. *Let n be even and $g < n$. Let $\mathcal{L}$ be a set of pairs (A, B, z_0), with $|A| = |B| = s$ and $z_0 \le \lfloor g/2 \rfloor$, such that*

(a) for every $z_0 \le \lfloor g/2 \rfloor$, every $(z_0, g - z_0)$-feasible pair of s-sets lies in the K-orbit of a pair (A, B) with $(A, B, z_0) \in \mathcal{L}$, and

(b) for every $(A, B, z_0) \in \mathcal{L}$, no set of g queens contained in $([0, n) \setminus A) \times ([0, n) \setminus B)$ with at most z_0 queens of colour 0 and at most $g - z_0$ of colour 1 dominates Q_n.

Then $\gamma(Q_n) > g$.

Proof. Suppose D is a dominating g-set. After the reflection of Lemma 8.3(3) if needed, $z_0(D) \le z_1(D)$, so $z_0(D) \le \lfloor g/2 \rfloor$. Choose s-sets A_0, B_0 of empty rows and columns. By Lemma 8.2, (A_0, B_0) is $(z_0(D), z_1(D))$-feasible, so by (a) there are $\sigma \in K$ and $(A, B, z_0(D)) \in \mathcal{L}$ with $\sigma(A_0, B_0) = (A, B)$. By Lemma 8.3(1), σD is a dominating g-set with the same colour sizes whose queens avoid the rows A and the columns B, contradicting (b). □

8.2 The computation

Stage 1 (pairs). A depth-first search decides membership of rows and columns in A and B in the order row 0, column 0, row $n-1$, column $n-1$, row 1, column 1, …, from the outside in. It maintains a maximum matching of $H_0 \cup H_1$ incrementally, prunes with (M) and (K), and after every block of four decisions (rows and columns t and $n - 1 - t$) discards prefixes that cannot extend to the canonical member of their K-orbit. The canonical member is the one whose sequence of block codes is lexicographically least, and the block code of an image depends only on the same block of the original, so this test is exact. At the leaves it decides the constrained vertex cover problem exactly, by branching. The output is a list of canonical feasible pairs. The search is split into parts by the index of the search node at depth 16 modulo the number of parts. For $n = 26$ there are 480 parts per colour split; for $n = 28$ there are 480 parts for each $z_0 \le 4$ and 4 800 for each $z_0 \ge 5$.

Stage 2 (queens). For each pair a branch-and-bound search finds all g-sets of queens inside the allowed region, with the prescribed colour sizes, that dominate the whole board. It picks an undominated square with the fewest available attackers and branches over those attackers, forbidding earlier siblings so that the branches partition the search space. It prunes with greedy packings of undominated squares whose sets of available attackers are pairwise disjoint, both overall and per colour.

The case $n = 26$. Here $g = s = 13$ and $z_0 \in \{0, \dots, 6\}$. Table 3 lists the work. Stage 1 leaves canonical feasible pairs only for the split $(6, 7)$, where there are 10,887 of them, and for the monochromatic split $(0, 13)$. In the latter case there is exactly one pair: A is the set of odd rows and B the set of even columns. This agrees with Weakley's structure theory [26], which the computation does not use. Stage 2 finds no dominating set for any of the 10,888 pairs. The SHA-256 hash of the survivor list begins with `58068a1dd390e350`. In total stage 1 took 7.0 and stage 2 2.6 CPU hours (single cores of a shared machine).

The case $n = 28$. Here $g = s = 14$ and $z_0 \in \{0, \dots, 7\}$. Table 4 lists the work for the 16,800 parts. Stage 1 leaves canonical feasible pairs only for the splits $(7, 7)$ (118,662 pairs) and $(6, 8)$ (101 pairs), and for $(0, 14)$ and $(1, 13)$. In the last two cases there is again exactly one pair, A the odd rows and B the even columns, whose grid $A \times B$ consists of squares of a single colour. This again

	stage 1			stage 2			certificate	
(z_0, z_1)	nodes	pairs	CPU s	nodes	CPU s	sets	cubes	CPU s
$(0, 13)$	792 319	1	0	9 367 246 608	7132	0	76	3
$(1, 12)$	3 426 552	0	0	–	–	–	76	3
$(2, 11)$	15 749 727	0	5	–	–	–	76	5
$(3, 10)$	75 767 491	0	27	–	–	–	76	15
$(4, 9)$	546 916 450	0	256	–	–	–	76	71
$(5, 8)$	4 296 637 439	0	2416	–	–	–	76	426
$(6, 7)$	29 237 608 861	10 887	22 318	3 408 018 489	2147	0	1072	6719

Table 3: The computation for $n = 26$, $g = 13$. Stage 1 and stage 2 are the search programs of Section 8.2. The last two columns refer to the stage-1 certificate of Section 8.3: the number of solved cubes and the CPU time of CaDiCaL plus `drat-trim`. Single-core CPU seconds on a shared machine.

agrees with Weakley's results: by Theorem 17 and Proposition 11 of [26] a 14-dominating set of Q_{28} would have colour classes $\{7, 7\}$ or $\{6, 8\}$. Altogether there are 118,765 pairs; the SHA-256 hash of the survivor list begins with `7bad471036c300f2`. Stage 1 took 83.1 CPU hours. Of the parts, 7457 (all but 16 of those with $z_0 = 7$, and 2673 of those with $z_0 = 6$) were run on rented single-core cloud containers (Modal), compiled there from the same source inside a pinned Debian image. The cost was about US$4. Every returned part was checked locally against its log and its SHA-256 hash before it was accepted, and the remaining parts, including 16 that exceeded the time limit of the cloud calls, ran locally. The cloud cores were about 1.8 times slower than the local ones, which inflates the CPU times of Table 4.

For $n = 28$ condition (b) of Proposition 8.4 rests on the SAT check of Section 8.3 alone. The branch-and-bound search of stage 2 was run only on the 102 pairs with $z_0 \in \{1, 6\}$, in 1501 s; it found no dominating set. Its running time has a heavy tail (up to 10^3 s for a single pair with $z_0 = 6$), so we did not run it on the 118,662 pairs with $z_0 = 7$. On the pair with $z_0 = 0$ we stopped it after 3.6 hours.

	stage 1			certificate		SAT stage 2
(z_0, z_1)	nodes	pairs	CPU s	cubes	CPU s	CPU s
$(0, 14)$	1 344 512	1	0	76	3	5910
$(1, 13)$	4 224 030	1	0	76	3	0
$(2, 12)$	21 160 768	0	5	76	6	–
$(3, 11)$	125 674 586	0	33	76	19	–
$(4, 10)$	998 940 852	0	313	76	86	–
$(5, 9)$	8 199 312 979	0	3536	76	483	–
$(6, 8)$	64 418 002 306	101	46 991	1072	5655	21
$(7, 7)$	251 094 656 771	118 662	248 194	1072	88 999	40 071

Table 4: The computation for $n = 28$, $g = 14$. Stage 1 is the search program of Section 8.2; most of the stage-1 time for $(7, 7)$ and part of it for $(6, 8)$ was spent on slower cloud cores. The certificate columns refer to the stage-1 certificate of Section 8.3, and the last column to the SAT check of condition (b), each CaDiCaL plus `drat-trim`. Single-core CPU seconds.

8.3 Certificates

The two searches of Section 8.2 are fast but intricate programs. Both conditions of Proposition 8.4 were therefore also established by SAT solving with proofs: CaDiCaL 3.1.0 [2] writes a DRAT proof of unsatisfiability for each formula and `drat-trim` [27] checks it. The formulas are produced by two short encoders that do not share code with the search programs. Cardinality constraints use the sequential counter [23].

Condition (b). For each listed pair, the formula has one variable per allowed square, one clause per square of the board stating that some allowed square attacking it (or equal to it) carries a queen, and constraints that at most z_q queens have colour q. For $n = 26$ all 10,888 formulas are unsatisfiable, and all 10,888 proofs were verified. CaDiCaL took 1137 s in total and `drat-trim` 1498 s. The monochromatic pair is by far the hardest instance (221 s and 408 s); each other instance takes less than 3 s for each of the two steps. For $n = 28$ all 118,765 formulas are unsatisfiable and all proofs were verified, in 21,985 s (CaDiCaL) and 24,017 s (`drat-trim`). The hardest instance is again the pair whose grid has a single colour, here with $z_0 = 0$ (1458 s and 4452 s); every other instance takes less than 32 s for each step.

Condition (a). Fix z_0. The formula F_{z_0} has variables a_r (row r is in A), b_x (column x is in B), y_d (difference line d in the cover) and w_t (sum line t in the cover), with the constraints $|A| = |B| = s$, at most z_q chosen difference lines and at most z_q chosen sum lines of colour q, and, for every square, the clause $\neg a_r \vee \neg b_x \vee y_{r-x} \vee w_{r+x}$. The projections of its models to the variables a, b are exactly the z-feasible pairs. For each listed pair (A, B, z_0) and each of its four images (P, Q) under K we add the clause $\bigvee_{r\in P} \neg a_r \vee \bigvee_{x\in Q} \neg b_x$, which excludes (P, Q). Condition (a) for this z_0 says exactly that the resulting formula F'_{z_0} is unsatisfiable.

We refute F'_{z_0} by cube and conquer [9], with a symmetry reduction. Fix the $4T$ variables $a_t, a_{n-1-t}, b_t, b_{n-1-t}$ $(t < T)$ in all 16^T ways; these cubes cover all assignments. The group K acts on the cubes, and $F'_{z_0} \wedge c$ is satisfiable if and only if $F'_{z_0} \wedge \sigma c$ is: the projections of the models of F'_{z_0} form a K-invariant set of pairs, by Lemma 8.3(2) and because the excluded set is a union of orbits. So it suffices to refute $F'_{z_0} \wedge c$ for one cube c in each orbit. We used $T = 2$ (76 orbits) for $z_0 \leq 5$ and $T = 3$ (1072 orbits) for $z_0 = 6$ and, for $n = 28$, $z_0 = 7$. A short script checks that the orbits of the chosen cubes cover all 16^T cubes. Every cube formula was refuted and every proof verified. For $n = 26$ this took 2.0 CPU hours and the proofs were about 25 GB in all. For $n = 28$ it took 26.5 CPU hours; for $z_0 = 7$ the formula has 477,629 clauses, most of them exclusion clauses, the largest proof was 1.26 GB and all proofs together about 182 GB. The proofs were deleted after checking. The encoder rebuilds the formulas deterministically, their SHA-256 hashes are recorded in the repository, and the verification script regenerates and checks a random sample of the proofs. For every $z_0 \geq 3$ and both n the certificate took less CPU time than the stage-1 search itself (Tables 3 and 4).

Complete replay. The certificate-only mode of the verification script (`-certify-all`) has been run to completion. It rebuilt every stage-1 cube formula and every stage-2 formula of both n from the committed survivor lists, solved each with CaDiCaL, checked each proof with `drat-trim` and deleted it. Its compacted records are in the repository; they contain every cube and every pair, all UNSAT with verified proofs and with the CNF hashes of the original records, and the default run of the script compares them with the original records.

What the proof rests on. This proof of $\gamma(Q_{26}) > 13$ and $\gamma(Q_{28}) > 14$ follows from Proposition 8.4, applied for $n = 26$ and for $n = 28$ with $\mathcal{L}$ the respective survivor list; with the upper bounds of [20, 19] for $n = 26$ and of [15] and Section 8.4 for $n = 28$ it gives the two values. Its correctness depends on Lemmas 8.2 and 8.3 and Proposition 8.4, which are proved above; on the two CNF encoders (`relax_cnf` in `code/refcheck.py`, about 30 lines using the cardinality encodings of PySAT [10], and `build` in `code/sat_place.py`, about 40 lines); on the short enumeration of cube orbit representatives and its check that their orbits cover all 16^T cubes; and on `drat-trim`. It does not depend on the correctness of the two search programs, which serve only to produce the lists $\mathcal{L}$; in particular it does not depend on the cloud computations. As a check that the encoders are not too strong, every listed pair, for both n, satisfies F_{z_0} without the exclusion clauses, the two known 14-queen dominating sets of Q_{26} satisfy the stage-2 encoding for $g = 14$, and the 15-queen dominating set of Q_{28} below satisfies it for $g = 15$.

8.4 Validation and independent checks

Known values. Table 5 shows the pipeline for $g = n/2$ and even $n \le 18$. The numbers of $n/2$-dominating sets up to the dihedral symmetries agree with the literature [4, 22], and for $n = 8, 14, 16$ no $n/2$-set exists. The recorded runs for $n = 20, 22, 24$ (23,888, 5087 and 56,957 feasible pairs) found no $n/2$-set, in agreement with [4]; the verification script reruns $n = 20, 22$ in its full mode.

n	4	6	8	10	12	14	16	18	20	22	24	26	28
number of $n/2$-sets up to symmetry	3	1	0	1	1	0	0	2	0	0	0	**0**	**0**

Table 5: Numbers of dominating sets of size $n/2$ of Q_n up to the dihedral group, as found by the pipeline. The values for $n \le 24$ agree with [4, 22]; those for $n = 26, 28$ are part of Theorem 1.4.

All $n/2$-sets in Table 5 with $n > 4$ are monochromatic. To test the pipeline on bichromatic sets, we also ran it with $g = \gamma(Q_n) = n/2 + 1$ for $n = 8, 14, 16$, where minimum dominating sets are plentiful and mostly bichromatic. Stage 2 examined 871, 571,200 and 555,156 feasible pairs and found 638, 588 and 371 classes. These agree with [22], and with [4] for $n = 8, 14$; for $n = 16$ Bird lists 43 classes, which Rostami and Bright corrected to 371.

Second implementations. Stage 1 was reimplemented separately. The second program uses another decision order, breadth-first augmenting paths, only bound (M), another exact cover algorithm and another canonical form. For every $n \le 18$ and for $n = 26$ the two programs produce the same set of orbits of feasible pairs; for $n = 26$ the second program needed 31,191,558,503 nodes and 15,019 s. For $n \le 12$ both also agree with a test of every pair (A, B), without pruning or symmetry. Stage 2 has the SAT check described above as its second implementation.

For $n = 28$ the second implementations were run only in part, since the certificates make the lower bound independent of all search programs. The second stage-1 program was run on every part with $z_0 \le 5$, where it finds the same orbits as stage 1 (the two single-colour pairs and nothing else), and on 1549 of the 4 800 parts with $z_0 = 6$, where its 90 pairs are all among those found by stage 1; it needed 38,258,356,242 nodes and 15,066 s. It was not run for $z_0 = 7$. The branch-and-bound search of stage 2 was run on the pairs with $z_0 \in \{1, 6\}$, as described in Section 8.2.

Independent referee checks. An independent AI referee, not involved in writing the search programs, reviewed the proofs of Section 8 and the two stage-1 programs. It also wrote the program `code/refcheck.py`, which uses only SAT solving and shares no code with the search programs.

This program enumerates all feasible pairs by incremental SAT solving with orbit-blocking clauses and finds the same orbit sets as stage 1 for every even $n \le 18$ and every z_0. It runs a complete independent pipeline (SAT enumeration of pairs, then SAT enumeration of all dominating sets inside each region), which reproduces the class counts above for $n \le 12$ and for $(n, g) = (8, 5)$. It also enumerates all g-dominating sets directly, without the relaxation, in the cases with $n \le 10$, again with the same counts. The referee also proposed the stage-1 certificate of Section 8.3; the same certificate is produced in the verification run for $n = 16$ and $n = 18$.

External review. An external AI review (ChatGPT Pro) of the code, which also regenerated all 15 base formulas of the stage-1 certificates and all 1 528 (for $n = 26$) and 2 600 (for $n = 28$) cube formulas from the committed survivor lists and found their SHA-256 hashes equal to the recorded ones, found three defects, now corrected, none of which affected the claimed values. When resumed, the certificate driver reused every cube already recorded in its output file without checking that the file was made for the same formula (the same n, g, z_0, T and base formula) or that the recorded formula hash matched, so that a rerun on a different, satisfiable formula could be reported as certified. Every certificate reported here was computed with one survivor list and one set of arguments, and each record's formula hash is computed from the exact text given to the solver. For $n = 28$, $z_0 = 7$ the run was stopped once, after 292 of the 1 072 cubes, and resumed with the same arguments and the same, unchanged survivor list (only the number of parallel processes changed); the regenerated hashes show that every record belongs to the formula it is filed under. The driver now compares the header and a freshly rebuilt base formula before reusing anything, rebuilds the formula of every reused cube and compares its hash, requires exactly the expected set of cubes, and solves failed, interrupted or mismatched cubes again. A regression test, run by the verification script, includes the reviewer's example ($n = 4$, $g = 2$, $z_0 = 0$, $T = 1$); the resume test of the cloud driver was tightened in the same way. The pinned `drat-trim` did not compile under GCC 14 (an implicit declaration of `getc_unlocked`) and is now built with `-D_DEFAULT_SOURCE`. Finally, an earlier version of this paper stated $\gamma(Q_n) \in \{\lceil n/2 \rceil, \lceil n/2 \rceil + 1\}$ for all $n \le 120$, omitting the exceptions $n = 3, 11$. The review's main suggestion, a verification mode that rebuilds every certificate formula of both stages from the survivor lists alone, solves it and checks the proof without rerunning the searches, is implemented as `verify.py -certify-all` and has been run to completion.

Upper bounds. The verification script checks that Kamenetsky's 14-queen set and the independent 14-queen set of Östergård and Weakley dominate Q_{26}, and that their 14-queen set of Corollary 1.5 dominates Q_{27}. For $n = 28$ we found a 15-queen dominating set by simulated annealing; it is in the repository, and the script checks it. (The value $i(Q_{28}) = 15$ of [15] already gives $\gamma(Q_{28}) \le 15$.) The script also checks the sets used for $n = 29, 30, 31$ in the proof of Corollary 1.5.

8.5 Comparison with the box method

The relaxation is much slower than the box method: for $n = 28$ its stage 1 took 83.1 CPU hours and its stage-1 certificate 26.5, against well under one CPU hour for the frame method (Table 2). Its value is its independence. It keeps all s chosen empty rows and columns, not only the outermost ones, records only which diagonals of each colour must be occupied, and places the queens in a second stage; it shares neither the reduction nor any code with the frame method or with [11]. For $n = 26$ and $n = 28$ the two methods agree.

9 Remarks

For even n and $k = n/2$ the identities leave no slack when the box is not square, and the frame method is fast: under two CPU hours for $n = 36$ and about six for $n = 44$, including proof checking, with the square boxes of deficit vector 0 taking most of the time. The cost grows by a factor of roughly 1.5 per step of 2 in n, so somewhat larger even boards appear to be within reach of the same code; the even values from $n = 48$ on remain open in the certified literature. For odd n and $k = (n + 1)/2$ there are two units of slack in each side identity and up to four in the border identity; the cases with deficit vector 0 and an almost square box, where the 18 queens would be pairwise non-attacking, account for about 40% of the time for $n = 35$, and the hardest single case took about an hour. Two further sound constraints that we tried on such cases, a colour balance between the diagonals and Weakley's moment identities [26] taken modulo small integers, did not help. The odd values of $n < 120$ that remain open, 47, 51, 59, 63, 67, 75, 79, 87, 95, 103, 111 and 119 (by [20] and the sets of [17]), are beyond the present computation without new ideas.

The relaxation of Section 8 keeps only the line structure of the squares that must be covered diagonally, yet it is strong enough that stage 1 leaves only 10,888 candidate pairs for $n = 26$ and 118,765 for $n = 28$. Its cost grows quickly: from $n = 26$ to $n = 28$ the stage-1 search grew from 7.0 to 83.1 CPU hours (part of the latter on slower cores), the number of pairs by a factor of about 11, and the stage-1 certificate from 2.0 to 26.5 CPU hours.

Reproducibility

The repository[2] contains, for the frame method, the case enumeration, the encoder, the drivers and checkers, the records of every case for $n = 26, 28, 32, 34, 35, 36, 40, 42, 44, 46$ (one JSON line per case or cube with the SHA-256 of the formula, the verdict, the checker's status and the CPU times), the records of the validation runs, the code and records of the cross-check, and the Lean sources; and, for the relaxation, the C sources of both stages and of the second stage-1 program, the CNF encoders, the referee's checks, the driver used for the cloud runs, and for $n = 26$ and $n = 28$ the survivor lists, per-part SHA-256 manifests, the logs of the searches, the per-pair SAT records, the per-cube records of the stage-1 certificates and the records of the complete replay. The script `scripts/verify.py` rebuilds the programs and, in about an hour on three cores, reruns the validation of the relaxation for small n, the referee's checks and the stage-1 certificate for $n = 16, 18$, the tests of the frame method, and seeded random samples of the computations; it checks every record set against its manifest or against the case list rebuilt from first principles, rebuilds the formulas of a sample of records and compares their hashes, checks the Lean data files against the records, the upper bounds, and the records of the cross-check. Every number quoted in this paper is generated from a file of claims that the script compares with the computed values, and each of fifteen altered copies of the records must make it fail. Optional modes rebuild every formula hash of the frame records (`-frame-rehash-all`), build the Lean formalisation and check its axioms (`-lean`), rerun the whole $n = 26$ relaxation (`-full`) or a larger part of $n = 28$ (`-heavy28`), and replay every certificate formula of the relaxation (`-certify-all`, about 45 CPU hours). The DRAT proofs themselves (about 232 GB for $n = 35$, and 25 and 182 GB for the relaxation certificates for $n = 26$ and $n = 28$) are not published; they are regenerated by solving the recorded formulas again.

[2] https://github.com/jasondavies/queen-domination

Declaration of AI-generated research and authorship

This manuscript is the product of an AI-driven research process. During a broader investigation of open problems, initiated by the human author, AI systems identified the problems, devised the methods, wrote and ran the code for the computations, wrote the Lean formalisation, and wrote this manuscript. The frame/deficit case split was first proposed in an earlier, uncertified attempt by Claude Fable 5.2 (Claude for web), run by the author; it was re-derived, proved and reimplemented here, and compared with the logs of that attempt only afterwards (the case lists for $n = 28, 30, 32$ agree). The line relaxation of Section 8 was devised first, when the AI system believed $\gamma(Q_{26})$ and $\gamma(Q_{28})$ to be open: its literature search missed [11] and [17], and an earlier version of this paper and of the OEIS entry presented those values as new. This was corrected when the prior work was pointed out. For $n = 35$ and $n = 40, \ldots, 46$ the AI system then searched the literature, preprint servers, code repositories and the OEIS history; it found no statement of $\gamma(Q_{35})$, and for $n = 40, \ldots, 46$ only the uncertified claims of [17]. Such a search cannot be exhaustive. AI systems produced the results and their presentation, including the mathematical exposition, the computations and the organisation and writing of the text; the human author directed the AI systems but wrote no part of the manuscript. The human author also launched the main cloud computation for $n = 28$, using commands prepared by the AI system. The system was Claude Opus 5.5 (Anthropic), used through Claude Code.

The computational claims are reproduced by `scripts/verify.py`. At the time of writing, the mathematics, the code and the formalisation have been checked only by AI systems and by computer verification. These checks include an adversarial review by an independent AI referee, who re-derived the proofs of Section 8, read the search programs, wrote the independent SAT-based checks of Section 8.4 and the stage-1 certificate of Section 8.3; an external AI review of the code of the relaxation (ChatGPT Pro), whose findings are corrected (Section 8.4); the Lean formalisation of the frame method; and the separately written cross-check of Section 6. No independent human expert has reviewed the work. The human author has not read this manuscript or checked its arguments. The author takes responsibility for the accuracy of this disclosure.

References

[1] K. E. Batcher, *Sorting networks and their applications*, in: AFIPS Spring Joint Computer Conference, 1968, 307–314. doi:10.1145/1468075.1468121.

[2] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, F. Pollitt, *CaDiCaL 2.0*, in: Computer Aided Verification (CAV 2024), Lecture Notes in Comput. Sci., Springer, 2024, 133–152. doi:10.1007/978-3-031-65627-9_7.

[3] A. Biere et al., *Kissat SAT solver*, version 4.0.4, https://github.com/arminbiere/kissat.

[4] W. H. Bird, *Computational methods for domination problems*, PhD thesis, University of Victoria, 2017. https://dspace.library.uvic.ca/handle/1828/8634.

[5] S. Bozóki, P. Gál, I. Marosi, W. D. Weakley, *Domination of the rectangular queen's graph*, Electron. J. Combin. **26**(4) (2019), #P4.45. doi:10.37236/6026.

[6] E. J. Cockayne, *Chessboard domination problems*, Discrete Math. **86** (1990), 13–20. doi:10.1016/0012-365X(90)90344-H.

[7] L. de Moura, S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction (CADE 28), Lecture Notes in Comput. Sci., Springer, 2021, 625–635. doi:10.1007/978-3-030-79876-5_37.

[8] D. Finozhenok, W. D. Weakley, *An improved lower bound for domination numbers of the queen's graph*, Australas. J. Combin. **37** (2007), 295–300.

[9] M. J. H. Heule, O. Kullmann, S. Wieringa, A. Biere, *Cube and conquer: guiding CDCL SAT solvers by lookaheads*, in: Hardware and Software: Verification and Testing (HVC 2011), Lecture Notes in Comput. Sci., Springer, 2012, 50–65. doi:10.1007/978-3-642-34188-5_8.

[10] A. Ignatiev, A. Morgado, J. Marques-Silva, *PySAT: a Python toolkit for prototyping with SAT oracles*, in: Theory and Applications of Satisfiability Testing (SAT 2018), Lecture Notes in Comput. Sci., Springer, 2018, 428–437. doi:10.1007/978-3-319-94144-8_26.

[11] Anonymous, *Hint and Conquer: new queen domination numbers by SAT solving*, preprint, October 2026, file `doc.pdf` on branch `hint-and-conquer` of the repository https://github.com/TahaRostami/Gamma.

[12] jkolantree, *BSC* (repository), directory `formal/q26_grid_annihilator` and files `applications/Q26_*`, commits of 11–12 August 2026, https://github.com/jkolantree/BSC.

[13] A. Karandikar, A. Dutta, *Improved lower bounds for queen's domination via an exactly-solvable relaxation*, Electron. J. Combin. **31**(4) (2024), #P4.38. doi:10.37236/12003.

[14] M. D. Kearse, P. B. Gibbons, *Computational methods and new results for chessboard problems*, Australas. J. Combin. **23** (2001), 253–284.

[15] A. Langlois-Rémillard, M. Müssig, É. Roldán, *Complexity of chess domination problems*, Res. Math. Sci. **12** (2025), no. 1, paper no. 14. doi:10.1007/s40687-024-00492-5.

[16] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), 2020, 367–381. doi:10.1145/3372885.3373824.

[17] S. Neuhaus, *Eigenschaften kleinster dominierender Mengen und Dominanzzahlen von Damengraphen*, Dissertation, Universität zu Köln, 2009. https://kups.ub.uni-koeln.de/2916/.

[18] S. Neuhaus, *Solutions for queen's domination* (repository), 2017, https://github.com/stefanneuhaus/queens-domination-solutions.

[19] OEIS Foundation Inc., *The On-Line Encyclopedia of Integer Sequences*, sequence A075458, with D. Kamenetsky's file of best known placements for $n \leq 26$ (2019), https://oeis.org/A075458.

[20] P. R. J. Östergård, W. D. Weakley, *Values of domination numbers of the queen's graph*, Electron. J. Combin. **8** (2001), #R29. doi:10.37236/1573.

[21] V. Raghavan, S. M. Venkatesan, *On bounds for a board covering problem*, Inform. Process. Lett. **25** (1987), 281–284. doi:10.1016/0020-0190(87)90201-8.

[22] T. Rostami, C. Bright, *Queen domination by SAT solving*, Art Discrete Appl. Math. (2026). doi:10.26493/2590-9770.2033.c47; arXiv:2508.11945.

[23] C. Sinz, *Towards an optimal CNF encoding of Boolean cardinality constraints*, in: Principles and Practice of Constraint Programming (CP 2005), Lecture Notes in Comput. Sci., Springer, 2005, 827–831. doi:10.1007/11564751_73.

[24] J. J. Watkins, *Across the Board: The Mathematics of Chessboard Problems*, Princeton University Press, 2004. doi:10.1515/9781400840922.

[25] W. D. Weakley, *A lower bound for domination numbers of the queen's graph*, J. Combin. Math. Combin. Comput. **43** (2002), 231–254.

[26] W. D. Weakley, *Queen domination of even square boards*, Electron. J. Combin. **29**(2) (2022), #P2.50. doi:10.37236/10617.

[27] N. Wetzler, M. J. H. Heule, W. A. Hunt Jr., *DRAT-trim: efficient checking and trimming using expressive clausal proofs*, in: Theory and Applications of Satisfiability Testing (SAT 2014), Lecture Notes in Comput. Sci., Springer, 2014, 422–429. doi:10.1007/978-3-319-09284-3_31.