

CONDORCET WINNING SETS OF SIZE THREE FOR ELECTIONS WITH SEVEN OR EIGHT CANDIDATES

PETER SEELMAN

ABSTRACT. A set W of candidates is a Condorcet winning set if every candidate outside W is ranked above all members of W by fewer than half of the voters. It is conjectured that every election has a Condorcet winning set of size at most 3. Elkind, Lang and Saffidine proved this for every election with at most 6 candidates, and for an odd number of voters up to 14 candidates; for an even number of voters with 7 or 8 candidates their bound is 4. For 7 candidates the statement also follows from their argument and the classical fact that every tournament on 6 vertices has a dominating set of size 2. Version 2 of the mixed-integer programming study of Zilberstein, Martins and Berker (arXiv:2604.19851v2, 10 October 2026) reports, from Gurobi computations with numerical tolerance 10^{-6} , that every election with at most 8 candidates has a Condorcet winning set of size 3. We give an exact, formally checked proof of this statement for 7 and 8 candidates, any number of voters and strict rankings; our certificates were timestamped on 29 September 2026. Hence the same holds for every election with at most 8 candidates, and a counterexample to the conjecture needs at least 9 candidates. The proof reduces the statement to the unsatisfiability of a propositional formula built from exact rational Farkas certificates and their images under relabeling of the candidates. For 8 candidates the formula has 280 variables and 4,161,137 clauses; it is split into 21 cubes, and each cube has an LRAT proof checked by the formally verified checker `cake_lpr`. The certificate, the proofs and a one-command verifier are public; a full check takes about 20 minutes on one machine with 32 GiB of memory.

1. INTRODUCTION

An election consists of a set C of m candidates and $n \geq 1$ voters, each with a strict linear order (ranking) of C . For $W \subseteq C$ and $c \in C \setminus W$ write $c \succ_v W$ if voter v ranks c above every member of W . Following Elkind, Lang and Saffidine [5, Definition 1], W *covers* c if more than $n/2$ voters rank some member of W above c , and W is a *Condorcet winning set* if it covers every candidate outside it. Equivalently, W is a Condorcet winning set if and only if

$$(1) \quad \#\{v : c \succ_v W\} < n/2 \quad \text{for every } c \in C \setminus W.$$

A tie at exactly $n/2$ counts against W . We say that c *overthrows* W if (1) fails for c . The family of Condorcet winning sets is closed upwards [5, Section 2], and the *Condorcet dimension* of an election is the smallest size of a Condorcet winning set. A Condorcet winner is a Condorcet winning set of size 1.

What was known. Elkind, Lang and Saffidine [5] introduced the notion. They exhibit a profile of dimension 3 with 15 candidates and an odd number of voters [5, Proposition 2], and they report that Geist found by SAT search a profile of dimension 3 with 6 candidates and 6 voters and showed that no smaller one exists [5, Section 4.1], [7]. Their upper bounds come from dominating sets of the majority tournament; the general one is logarithmic in m [5, Proposition 4], and its refinement reads as follows. With $f(m) = m/2 - 1$ for even m , $f(m) = (m - 1)/2$ for odd m , and $g(m) = \min\{k : f^k(m) = 0\}$, they prove:

Date: 30 September 2026; revised 14 October 2026. IRSI research note; not peer reviewed.

2020 Mathematics Subject Classification. 91B12, 05A05, 05C20, 68V15.

Key words and phrases. Condorcet winning set, Condorcet dimension, SAT, LRAT, Farkas certificate, `cake_lpr`.

Theorem 1.1 ([5, Proposition 4 and Corollary 1]). *Every m -candidate n -voter profile has Condorcet dimension at most $g(m)$ if n is odd and at most $g(m) + 1$ if n is even. Here $g(3) = \dots = g(6) = 2$ and $g(7) = \dots = g(14) = 3$.*

The even case is obtained by deleting one voter, applying the odd bound, and adding that voter’s top candidate [5, proof of Proposition 4]; this costs one extra candidate. So Theorem 1.1 gives size at most 3 for every election with $m \leq 6$, and for every election with an odd number of voters and $m \leq 14$; for an even number of voters and $m \in \{7, 8\}$ it gives only 4. Elkind, Lang and Saffidine knew no profile of dimension 4 or more and listed its existence as open [5, Sections 4.2 and 8].

Charikar, Lassota, Ramakrishnan, Vetta and Wang [3] proved that a Condorcet winning set of size 6 always exists, whatever m and n , and conjectured that for every k and every $\alpha \geq 2/(k+1)$ an α -undominated set of size k always exists [3, Conjecture 1]; for $k = 3$ and $\alpha = 1/2$ this says that every election has Condorcet dimension at most 3. Song, Nguyen and Lin [11] improved 6 to 5.

In version 1 of [14] (21 April 2026), Zilberstein, Berker, Li and Martins model an electorate as a probability distribution x on the $m!$ rankings and study

$$(2) \quad \alpha^*(m, k) = \max_x \min_{|W|=k} \max_{c \notin W} \sum_{s: c \succ_s W} x[s],$$

For $1 \leq k < m$, every election with m candidates has a Condorcet winning set of size k if and only if $\alpha^*(m, k) < 1/2$ (cf. [14, v1, Sections 2–3; v2, Sections 3–4 and Proposition 1]). Indeed, the objective is continuous on the compact simplex. If its maximum is at least $1/2$, choose one challenger per committee at a maximizing distribution. The resulting inequalities, together with the simplex constraints, define a nonempty rational polytope, which has a rational point; clearing denominators gives a finite-electorate counterexample. Conversely, any such counterexample makes the maximum at least $1/2$. They compute (2) with the Gurobi mixed-integer solver. In Table 3 of version 1 (fixed first committee, no subsampling, 720 s limit) the values for $k = 3$ are 0.250, 0.300, 0.333 for $m = 4, 5, 6$ and the interval $[0.345, 0.429]$ for $m = 7$, meaning the solver timed out with a feasible value 0.345 and a solver-reported upper bound 0.429; there is no $m = 8$ entry for $k = 3$. Its Table 4 (a relaxed “infinite clone” model) has an $m = 8$ entry only for $k = 2$, its Table 5 reports subsampled searches at $m = 7, 8, 9$ that found no counterexample, and its Table 6 gives the root LP relaxation value 0.951 at $m = 8, k = 3$. Version 1 states that its results “remain empirical” [14, v1, Section 6]. Version 2 (10 October 2026; Section 5 and Table 2, 81,400 s limit) reports for $k = 3$ the intervals $[0.353, 0.400]$ for $m = 7$ and $[0.375, 0.444]$ for $m = 8$, and concludes that all elections with $m \leq 8$ and $3 \leq k < m$ admit a Condorcet winning set; these are floating-point solver bounds (tolerance 10^{-6}) rather than exact rational certificates.

The case $m = 7$ with an even number of voters is not stated in [5] or in version 1 of [14], but it follows from the proof of [5, Proposition 4]. Version 2 of [14] reports that case for every number of voters, from the floating-point bounds in its Table 2. Delete one voter, with top candidate t . The majority relation of the remaining odd electorate is a tournament, and its restriction to the six candidates other than t has a dominating set D with $|D| \leq 2$, because no tournament on fewer than 7 vertices has every pair of vertices beaten by a common vertex [6] (a direct check of all 2^{15} tournaments on 6 labeled vertices confirms this). Then $D \cup \{t\}$ is a Condorcet winning set of the remaining electorate that contains the deleted voter’s top candidate, so by the argument of [5, proof of Proposition 4] it is a Condorcet winning set of the whole election. For $m = 8$ with an even number of voters, the only earlier claim we found is the computational one of [14] version 2 (posted after our certificates were timestamped); we know of no earlier exact or formally verified proof. The search covered indexed citing papers and arXiv titles and abstracts (Section 7); the $m = 7$ deduction above is not a new existence result. A public problem tracker [10] lists, for the

same tie convention, an unattributed claim that a counterexample needs at least 8 candidates, marked as having incomplete dependencies, and suggests an exact route for 8 candidates through Farkas certificates and an unproved lemma; it gives no proof for 8 candidates. We learned of the tracker after the certificates of this paper were timestamped (Section 5).

Results.

Theorem 1.2. *Let $m \in \{7, 8\}$. Every election with m candidates, any number $n \geq 1$ of voters and strict rankings has a Condorcet winning set of size at most 3.*

For odd n this is contained in Theorem 1.1, and for $m = 7$ it also follows from the argument above. Relative to the published sources examined through 14 October 2026, the new content is an exact, formally verified proof of the case $m = 8$ with even n (which [14] v2 reports from floating-point MILP), together with exact certificates for $m = 7$ and $m = 8$ that give the quantitative bounds of Proposition 4.1. The certificate for $m = 8$ does not use Theorem 1.1 or the case $m = 7$.

Corollary 1.3. *Every election with at most 8 candidates has a Condorcet winning set of size at most 3. Consequently a counterexample to the conjecture of [3] for $k = 3$ has at least 9 candidates, and if it has at most 18 candidates then it has an even number of voters.*

Proof. If $m \leq 3$, take $W = C$. If $3 < m < 8$, add $8 - m$ new candidates, ranked last in a fixed order by every voter, and let W be a Condorcet winning set of size at most 3 of the new election (Theorem 1.2). Put $W' = W \cap C$, so $|W'| \leq 3$. Let $c \in C \setminus W'$; then $c \notin W$. If a voter v ranks c above every member of W' , then v ranks c above every member of W , because the members of $W \setminus W'$ are new candidates and every voter ranks them below c . Hence $\#\{v : c \succ_v W'\} \leq \#\{v : c \succ_v W\} < n/2$, and W' is a Condorcet winning set of the original election. For the last sentence: with an odd number of voters the majority relation is a tournament and each of its dominating sets is a Condorcet winning set [5, proof of Proposition 4], and no tournament on fewer than 19 vertices has every 3 vertices beaten by a common vertex [12], [8, Section 11, Corollary to Theorem 16], so every tournament on at most 18 vertices has a dominating set of size at most 3. (Theorem 1.1 gives the same for at most 14 candidates.) Bloks [2, Chapter 6] made this observation for profiles without majority ties. $\square$

The same certificates also give a quantitative bound (Proposition 4.1): $\alpha^*(8, 3) \leq 9/19$ and $\alpha^*(7, 3) \leq 2/5$. The second is a rigorous rational bound; it equals the upper value 0.400 that [14, v2, Table 2] reports numerically, and improves the value 0.429 of version 1. For $m = 8$ our bound $9/19 \approx 0.474$ is weaker than the numerical upper value 0.444 of version 2. We do not establish that either bound is tight. This consequence follows from the verified formulas and Theorem 2.5; an independent referee recomputed every released core bound exactly and checked the quantitative argument (Section 4).

2. REDUCTION TO A FINITE PROPOSITIONAL FORMULA

Fix $m \geq 4$ and let $C = \{0, \dots, m-1\}$. Let $\mathcal{S}$ be the set of $m!$ rankings and Δ the set of probability distributions on $\mathcal{S}$. An election with n voters defines $x \in \Delta$ by $x[s] = \#\{v : v \text{ has ranking } s\}/n$, and W is a Condorcet winning set if and only if $\mu_x(W, c) < 1/2$ for all $c \notin W$, where

$$\mu_x(W, c) = \sum_{s: c \succ_s W} x[s].$$

This covers odd and even n and ties at $n/2$ at once. An *event* is a pair $e = (W, c)$ with $|W| = 3$ and $c \in C \setminus W$; let E_m be the set of events, so $|E_m| = \binom{m}{3}(m-3)$, which is 140 for $m = 7$ and 280 for $m = 8$. For a ranking s let $[s \in e] = 1$ if $c \succ_s W$ and 0 otherwise.

Definition 2.1. A *core* is a set $K \subseteq E_m$ with weights $y_e > 0$ ($e \in K$), $\sum_{e \in K} y_e = 1$, such that

$$b(K, y) = \max_{s \in \mathcal{S}} \sum_{e \in K} y_e [s \in e] < \frac{1}{2}.$$

Lemma 2.2. If (K, y) is a core and $t > b(K, y)$, then no $x \in \Delta$ has $\mu_x(e) \geq t$ for all $e \in K$. In particular no x has $\mu_x(e) \geq 1/2$ for all $e \in K$.

Proof. $\sum_{e \in K} y_e \mu_x(e) = \sum_s x[s] \sum_{e \in K} y_e [s \in e] \leq b(K, y) < t$, while the left side is at least t if every $\mu_x(e) \geq t$. $\square$

A core is a Farkas certificate: it is a nonnegative combination of the inequalities $\mu_x(e) \geq 1/2$ that no distribution can satisfy. The permutation group S_m acts on events by $\sigma(W, c) = (\sigma(W), \sigma(c))$ and on distributions by relabeling the candidates in every ranking; then $\mu_{\sigma x}(\sigma e) = \mu_x(e)$, and σK with the transported weights is again a core with the same bound.

Example 2.3. The first core of the released $m = 8$ list is

$$K = \{(\{0, 1, 3\}, 2), (\{0, 2, 3\}, 1), (\{1, 2, 3\}, 0)\}$$

with weights $1/3$. Each event says that a given candidate is the highest of $\{0, 1, 2, 3\}$, so a ranking makes at most one of them true and $b = 1/3$. Three disjoint events cannot each have mass at least $1/2$.

Definition 2.4. Let $\mathcal{K}$ be a finite nonempty list of cores. The formula $F_m(\mathcal{K})$ has one Boolean variable v_e for each $e \in E_m$ and the clauses

- (1) for each 3-set W : $\bigvee_{c \notin W} v_{(W, c)}$ (at least one) and $\neg v_{(W, c)} \vee \neg v_{(W, c')}$ for $c \neq c'$ (at most one);
- (2) the unit clause $v_{(\{0, 1, 2\}, 3)}$;
- (3) for each $(K, y) \in \mathcal{K}$ and each $\sigma \in S_m$: $\bigvee_{e \in K} \neg v_{\sigma e}$, one clause per distinct image set σK .

Theorem 2.5. If $F_m(\mathcal{K})$ is unsatisfiable, then every election with m candidates has a Condorcet winning set of size 3. If moreover $b(K, y) \leq \beta$ for every $(K, y) \in \mathcal{K}$, then $\alpha^*(m, 3) \leq \beta$.

Proof. Let $\beta_0 = \max_{(K, y) \in \mathcal{K}} b(K, y)$, so $\beta_0 < 1/2$. For $x \in \Delta$ let $t(x) = \min_{|W|=3} \max_{c \notin W} \mu_x(W, c)$; then $\alpha^*(m, 3) = \max_x t(x)$, and an election without a Condorcet winning set of size 3 has $t(x) \geq 1/2$. So both claims follow once we show: if some x has $t = t(x) > \beta_0$, then $F_m(\mathcal{K})$ is satisfiable. Every 3-set W has a challenger c with $\mu_x(W, c) \geq t$. Pick one such pair (W_0, c_0) and $\tau \in S_m$ with $\tau(W_0) = \{0, 1, 2\}$ and $\tau(c_0) = 3$, and replace x by τx ; this does not change t . Now choose $f(W) \notin W$ with $\mu_x(W, f(W)) \geq t$ for every W , with $f(\{0, 1, 2\}) = 3$, and set $v_{(W, c)}$ true if and only if $c = f(W)$. Clauses (1) and (2) hold. If the clause (3) of some (K, y) and σ failed, every event of σK would be true, so $\mu_x(\sigma e) \geq t$ for $e \in K$, hence $\mu_{\sigma^{-1}x}(e) \geq t > \beta_0 \geq b(K, y)$ for all $e \in K$, contradicting Lemma 2.2. So the assignment satisfies $F_m(\mathcal{K})$. $\square$

Only the at-least-one clauses are used for exhaustiveness in the cube split below; the at-most-one clauses are true under the assignment in the proof and only help the solver. The unit clause is a symmetry-breaking step whose soundness is the relabeling in the proof. The bound $b(K, y)$ is a maximum over the finite set $\mathcal{S}$ and is computed exactly, and ties and the parity of n are absorbed into the condition $\mu_x \geq 1/2$; no voter count is fixed.

Cube split. For $m = 8$ the formula is refuted in 21 pieces. A *leaf* is $F_8(\mathcal{K})$ plus positive unit clauses. By the at-least-one clause of $\{4, 5, 6\}$, a satisfying assignment picks some challenger $c \in \{0, 1, 2, 3, 7\}$ of $\{4, 5, 6\}$. If $c = 3$ it satisfies the leaf with the single unit $v_{(\{4, 5, 6\}, 3)}$. If $c \in \{0, 1, 2\}$ it also picks a challenger $d \in \{0, 1, 2, 3, 4\}$ of $\{5, 6, 7\}$ and satisfies the leaf with units $v_{(\{4, 5, 6\}, c)}, v_{(\{5, 6, 7\}, d)}$. If $c = 7$ it picks $d \in \{0, 1, 2, 3, 6\}$ for $\{4, 5, 7\}$ and satisfies the leaf with units $v_{(\{4, 5, 6\}, 7)}, v_{(\{4, 5, 7\}, d)}$. These $1 + 15 + 5 = 21$ leaves are exhaustive, so $F_8(\mathcal{K})$ is unsatisfiable if all 21 leaves are.

3. THE CERTIFICATES

Finding cores. The cores were found by counterexample-guided search, which is not part of the argument. A SAT solver proposes a challenger for every 3-set; a linear program over the $m!/3!$ ranking prefixes of length $m - 3$ (the order of the last three candidates never affects an event) tests whether some distribution gives every chosen event mass at least $1/2$; if not, a dual solution is shrunk by greedy deletion to a small support, and its weights are made rational and rechecked exactly. The new core’s orbit is added to the formula and the loop repeats. For $m = 8$ this was complemented by sampling SAT models of a fixed relaxation with different seeds and by local search. Only the final cores and their exact check enter the proof.

$m = 7$. Nine cores suffice: three of size 3 with weights $1/3$ (bound $1/3$) and six of sizes 4 and 5 with weights $1/5$ and $2/5$ (bound $2/5$). Their S_7 images give 21,980 distinct clauses, and F_7 has 140 variables and 22,226 clauses. CaDiCaL 3.0.1 [1] refutes it in about 38 s and writes a binary LRAT proof [4] of 109,076,017 bytes (55,056,808 bytes after xz). `cake_lpr` [13] verifies it in 8.4 s; `drat-trim` and `lrat-check` also verify the DRAT and text LRAT versions.

$m = 8$. The released list has 160 exact certificates with 145 distinct supports, which form 141 orbits under S_8 . Support sizes range from 3 to 10 events (Table 1); most supports involve 6 or 7 candidates. The largest bound is $b = 9/19$. The orbit clauses number 4,160,520; with 56 at-least-one clauses, 560 at-most-one clauses and the unit clause, F_8 has 280 variables and 4,161,137 clauses. This formula uses no clauses derived from smaller cases and no lexicographic symmetry breaking beyond the unit clause.

events per core	3	4	5	6	7	8	9	10
distinct supports	5	4	74	25	29	5	2	1

TABLE 1. Sizes of the 145 distinct $m = 8$ core supports.

Each of the 21 leaves was solved by CaDiCaL 3.0.1 with `--lrat` (seed 1), and each binary LRAT proof was compressed with `zstd`. Table 2 lists the leaves. Solving took 60,948 s of summed solver wall time (16.9 solver-hours), and the proofs take 5,045,711,702 bytes compressed. Leaves in which 3 challenges $\{4, 5, 6\}$, $\{5, 6, 7\}$ or $\{4, 5, 7\}$, or 6 challenges $\{4, 5, 7\}$, are easy (148 to 882 s); the others take 50 to 83 minutes.

4. WHAT MUST BE TRUE, AND HOW EACH STEP IS CHECKED

Theorem 1.2 for $m = 8$ follows from Theorem 2.5 and the cube split once the following hold.

- (C1) Every listed (K, y) is a core: positive weights summing to 1 with $b(K, y) < 1/2$ over all $8! = 40,320$ rankings.
- (C2) The base file `A145-none.cnf` is exactly $F_8(\mathcal{K})$ under the stated numbering of events.
- (C3) Each leaf file is the base file plus the stated positive units, and the leaves cover all cases.
- (C4) Each leaf file is unsatisfiable.

The verified step. `cake_lpr` [13] is an LRAT/LPR checker verified in HOL4 through CakeML compilation to machine code. For its two-file invocation, the theorem states that printing `s VERIFIED UNSAT` implies unsatisfiability of the formula parsed from the DIMACS input, under its filesystem, FFI and target-machine assumptions. Our use of pre-generated assembly additionally trusts its correspondence to the proved code and the assembler, linker and concrete C FFI wrapper. We use the pre-built assembly `cake_lpr.S` of https://github.com/tanyongkiam/cake_lpr at commit

a36874a8b750b43fe4b385b8ddbf5b033e46a3fa

leaf	cube	solve (s)	proof (MB)	check A (s)	check B (s)
Lcube-456-3	(456, 3)	882	66.4	38	23
Lc456-0-567-0	(456, 0), (567, 0)	4,499	368.2	90	66
Lc456-0-567-1	(456, 0), (567, 1)	3,527	267.1	71	57
Lc456-0-567-2	(456, 0), (567, 2)	3,015	245.1	87	50
Lc456-0-567-3	(456, 0), (567, 3)	249	27.5	25	17
Lc456-0-567-4	(456, 0), (567, 4)	4,750	381.7	83	84
Lc456-1-567-0	(456, 1), (567, 0)	3,250	238.9	82	48
Lc456-1-567-1	(456, 1), (567, 1)	3,997	315.2	71	58
Lc456-1-567-2	(456, 1), (567, 2)	3,441	277.2	85	52
Lc456-1-567-3	(456, 1), (567, 3)	268	23.4	23	16
Lc456-1-567-4	(456, 1), (567, 4)	4,165	341.7	73	61
Lc456-2-567-0	(456, 2), (567, 0)	3,168	271.0	93	50
Lc456-2-567-1	(456, 2), (567, 1)	3,294	262.7	65	50
Lc456-2-567-2	(456, 2), (567, 2)	3,663	337.5	71	58
Lc456-2-567-3	(456, 2), (567, 3)	310	35.9	25	17
Lc456-2-567-4	(456, 2), (567, 4)	4,195	370.6	75	63
Lc456-7-457-0	(456, 7), (457, 0)	4,171	330.3	84	59
Lc456-7-457-1	(456, 7), (457, 1)	4,997	441.4	86	77
Lc456-7-457-2	(456, 7), (457, 2)	4,735	396.7	87	69
Lc456-7-457-3	(456, 7), (457, 3)	148	17.3	22	16
Lc456-7-457-6	(456, 7), (457, 6)	224	30.0	20	17
total		60,948	5,045.7	1,356	1,008

TABLE 2. The 21 leaves for $m = 8$. A cube entry $(456, c)$ is the unit clause $v_{(\{4,5,6\},c)}$; proof sizes are zstd-compressed binary LRAT. Check A is the builder’s cake_lpr run; check B is the independent recomputation’s run under a 28 GiB address-space cap, on the same proof files with the same checker binary.

step	how it is checked	status
Reduction (Thm. 2.5), cube split, Cor. 1.3, Prop. 4.1	proofs in this paper	argued, not formally verified
(C1)	exact integer arithmetic over all rankings, in two independently written programs (<code>gen.py</code> and <code>recompute.py</code>)	computed exactly; the code is trusted
(C2)	two independently written generators produce byte-identical files (SHA-256 <code>348efbbb...44cc</code>)	computed; Python generators and verification scripts trusted
(C3)	<code>cover.py</code> , and separately <code>recompute.py</code> , check the leaf contents, the at-least-one clauses and exhaustiveness	computed; the code is trusted
(C4)	cake_lpr prints <code>s VERIFIED UNSAT</code> for each leaf	formally verified checker

TABLE 3. Status of each link. “Trusted” means the code is not formally verified but was written twice or read in full.

(SHA-256 `2f3af32d...7d4a`), compiled with `gcc -O2 basis_ffi.c cake_lpr.S`; the resulting binary had SHA-256 `1822ca1e...09ca` in two independent builds. CaDiCaL and its proofs are not trusted: a wrong proof is rejected. cake_lpr exits with status 0 even when it rejects a proof, so the verifier requires the exact output line. In controlled tests a truncated proof was rejected at the

hash check, and an empty proof whose hash had been entered in the manifest reached `cake_lpr` and was rejected there.

The unverified steps. There is no Lean or HOL statement of the encoding. The link between Theorem 2.5 and the files (C1 to C3) rests on short Python programs: event indexing in the order of `itertools.combinations` with ascending challengers, the action of S_8 , an exact core check with Python integers and `Fraction`, and clause writing. For $m = 8$ these programs were written twice, by different AI agents from different providers, and the two write byte-identical base and leaf files; the recomputation ran in a sandbox without network access. For $m = 7$ two independently written programs produce the same clause set, and the independent one checks cores on ranking prefixes while the other checks all 7! full rankings. Controls: removing all core clauses from F_7 leaves a satisfiable formula; dropping any one of seven of the nine $m = 7$ cores makes it satisfiable; a mutated weight is rejected by the exact check; for $m = 8$, 300 random integer profiles never make all events of a core clause hold, and a planted invalid clause is flagged.

The quantitative bound.

Proposition 4.1. $\alpha^*(8, 3) \leq 9/19$ and $\alpha^*(7, 3) \leq 2/5$. Thus every election with 8 candidates has a 3-set W such that every $c \notin W$ is ranked above all of W by at most 9/19 of the voters, and the same holds for 7 candidates with 2/5.

Proof. The largest core bound is 9/19 for $m = 8$ and 2/5 for $m = 7$, and F_8 and F_7 are unsatisfiable (for F_8 through the cube split, which uses only the at-least-one clauses). Apply the second statement of Theorem 2.5. $\square$

Table 3 of version 1 of [14] reports a numerical lower value of approximately 0.345 (Table 2 of version 2 reports 0.353); our rigorous upper bound is $\alpha^*(7, 3) \leq 2/5$. The proposition uses the same formulas and LRAT checks as Theorem 1.2; its quantitative argument and all per-core bounds were independently rechecked for this manuscript. That recomputation was done by a referee agent from a different provider, with its own code, in exact rational arithmetic over all 7! and 8! rankings for the 9 and 160 released cores; it found the maxima 2/5 and 9/19.

5. REPRODUCTION

Everything is at <https://github.com/Aspect5/condorcet-certificates> (commit 7e68ba6); the 22 proofs (21 for $m = 8$, one for $m = 7$; 5,100,768,510 bytes) are assets of release v1. Commit 7e68ba6 hardens the verifier of release v1 (commit 1982b4f), whose exact checks could be skipped by running Python with optimization enabled; the cores, formulas and proofs are unchanged.

```
git clone https://github.com/Aspect5/condorcet-certificates
cd condorcet-certificates
git checkout --detach 7e68ba60a69e3ac83ec79e90e78c9094287a3c3a
./verify.sh --quick # hashes and CNF regeneration
CAKE_HEAP_MB=26000 ./verify.sh # also downloads and checks all 22 proofs
```

`verify.sh` checks that the SHA-256 of `MANIFEST.sha256` is the timestamped value given below, checks every file listed there (four scripts changed after the timestamp have new hashes, listed in `MANIFEST.post-stamp.sha256`), refuses to run Python with optimization, requires the exact worst bounds 2/5 and 9/19, builds `cake_lpr` from the pinned commit and checks the hash of `cake_lpr.S`, regenerates the $m = 7$ formula and the $m = 8$ base formula from the cores (rechecking every core exactly) and compares their hashes, writes the 21 leaves and compares their hashes, downloads any missing proof and checks its hash, runs `cake_lpr` on each proof and requires `s VERIFIED UNSAT`, and finally runs `cover.py`. It needs python3 with numpy, gcc, git, curl, zstd and xz.

Measured runs on a 32-vCPU cloud host (full runs used `CAKE_HEAP_MB=26000`):

run	wall time	memory
--quick, public clone, 30 Sep 2026 [†]	34 s	1.8 GiB peak RSS
full, independent replay, 30 Sep 2026 [†]	22 min 18 s	cake_lpr heap 26,000 MB, 32 GiB cap
full, fresh public clone, 30 Sep 2026 [†]	17 min 47 s	same settings, proofs downloaded
full, pinned commit 7e68ba6, 1 Oct 2026	23 min 2 s	25.4 GiB peak RSS
cake_lpr on the 21 $m = 8$ leaves (recomputation)	1,008 s	25.4 GiB peak RSS, 28 GiB cap

[†] Verifier of release v1 (commit 1982b4f). All full runs ended with ALL CHECKS PASSED. The leaf formulas need about 2.7 GB of disk.

Timestamp. MANIFEST.sha256 lists 64 files, including every formula and proof, and has SHA-256 ae319c72fe646532c836856be0f86572b8c92d10b84c6de9050815d572c77555.

Its OpenTimestamps [9] file commits this hash to Bitcoin block 969212, whose header time is 2026-09-29 22:18:32 UTC; the commitment was checked against the block header’s merkle root as served by two public block explorers. This shows that the manifest existed by that block, and matching hashes tie the proofs and formulas to it. Header times are set by miners and are not an exact wall clock. An earlier manifest, which differs only in three documentation files, is anchored in block 969208.

6. LIMITS, AND $m = 9$

The result is a computer proof of finitely many cases and has not been peer reviewed. Its trusted base includes the execution assumptions for cake_lpr described in Section 4, the Python programs and the mathematical reduction; no one has checked the LRAT derivations by hand, and the formal guarantee stops at “the parsed formulas are unsatisfiable.” The proof explains little: we have no short proof of the $m = 8$ case that avoids the computation, and we do not know how many of the 141 orbits are necessary. The result does not settle any case with $m \geq 9$, nor does it improve the general size-5 guarantee to size 4. The same conclusion for $m \leq 8$ was reported independently from floating-point MILP computations [14] (v2); the two methods share no code.

Bounded trials for $m = 9$ remained inconclusive. Lifting the released $m = 8$ core orbits gives 19,747,224 clauses; tested unsplit and shallow-cube instances did not return a verdict within limits of 0.5 to 3 hours. These internal measurements have not been independently rechecked. Materializing the orbits poses a scaling challenge. Generating needed orbit clauses during search, stronger symmetry handling and linear-programming checks of partial assignments are possible directions.

7. HOW THIS WORK WAS DONE

The problem was taken from the list FrontierMath: Open Problems of Epoch AI, which asks for an election in which every Condorcet winning set has size at least 4. Two other problems from that list were attempted by agents of the lab in the same week without a result (neither a counterexample nor a proof). The search, the certificates and the proofs were produced by AI agents run by IRSI, under rules that a result stays a candidate until an evaluator that its builder does not control recomputes it. The recorded tasks name Claude Opus 5.5 (Anthropic) for the $m = 8$ certificate build and GPT-6 Sol (OpenAI, through Codex) for the $m = 7$ and $m = 8$ recomputations; the internal referee of this manuscript used GPT-6.1 Sol. The records do not name the model of the initial $m = 7$ build or of the first part of the $m = 8$ search. The core search is the loop described in Section 3: it started from the nine $m = 7$ cores and earlier trial cores, grew the core list by counterexample-guided search, first with additional clauses for smaller cases and lexicographic symmetry breaking and then in the self-contained form used here, and ended with the 21-way cube split. From queuing the $m = 8$ search (28 September 2026, 10:58 UTC) to

the independent recheck (29 September 2026, 20:29 UTC) took 33.5 hours of wall time on one 32-vCPU cloud instance; solver runs were capped at 24 threads. Pricing that whole interval at the instance’s on-demand list price (1.428 US dollars per hour) gives about 48 US dollars, an estimate rather than an invoice; the model use was covered by flat-rate subscriptions and existing cloud credits, and every task had a marginal model-spending cap of zero. The task files given to the agents are available from the author. For $m = 8$, one agent searched for cores and a second built the self-contained certificate; a third agent, from the other provider and with no part in the build, wrote its own generator, reproduced the formulas byte for byte and reran `cake_lpr` on all 21 proofs. That rerun used the same core file, the same 21 proof files and a `cake_lpr` binary with the same SHA-256 as the build: it is one proof artifact checked twice by the same checker, not two independent proofs. For $m = 7$, the certificate was built, independently recomputed with a different encoding, and proof-checked by three separate agents. Separate agents read the prior literature in full, searched the papers citing [5] and [14] through 29 September 2026, and reviewed the public text; the literature check was repeated through 13 October 2026, and a release check on 14 October 2026 found version 2 of [14] (10 October 2026), which that repeat had missed; this revision describes it. This manuscript was drafted by an AI agent from the certificate’s documentation. For the 13 October revision, an independent GPT-6 Astra agent (through Codex) reread every theorem and number against the verifier logs and reran the exact recomputation offline, and a Claude Opus 5.5 agent applied the resulting corrections. For the 14 October revision, a Claude Opus 5.5 agent applied the release check’s changes describing version 2 of [14], and a separate Claude Sonnet 5.5 agent and a Composer agent (through Cursor) each reread the changed lines against the text of that version and the verifier logs. Peter Seelman directed the work.

REFERENCES

1. Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt, *CaDiCaL 2.0*, Computer Aided Verification (CAV 2024), Lecture Notes in Computer Science, vol. 14681, Springer, 2024, pp. 133–152.
2. Suzanne Bloks, *Condorcet winning sets*, MSc dissertation, London School of Economics and Political Science, 2018, https://suzannebloks.com/wp-content/uploads/2020/12/Bloks_2018_Condorcet-Winning-Sets_Dissertation-LSE.pdf.
3. Moses Charikar, Alexandra Lassota, Prasanna Ramakrishnan, Adrian Vetta, and Kangning Wang, *Six candidates suffice to win a voter majority*, Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC 2025), 2025, arXiv:2411.03390, pp. 1590–1601.
4. Luís Cruz-Filipe, Marijn J. H. Heule, Warren A. Hunt, Jr., Matt Kaufmann, and Peter Schneider-Kamp, *Efficient certified RAT verification*, Automated Deduction (CADE-26), Lecture Notes in Computer Science, vol. 10395, Springer, 2017, pp. 220–236.
5. Edith Elkind, Jérôme Lang, and Abdallah Saffidine, *Condorcet winning sets*, Social Choice and Welfare **44** (2015), 493–517.
6. Paul Erdős, *On a problem in graph theory*, The Mathematical Gazette **47** (1963), no. 361, 220–223.
7. Christian Geist, *Finding preference profiles of Condorcet dimension k via SAT*, 2014, arXiv:1402.4303.
8. John W. Moon, *Topics on tournaments*, Holt, Rinehart and Winston, New York, 1968, Public-domain edition: Project Gutenberg EBook 42833, <https://www.gutenberg.org/ebooks/42833>.
9. *OpenTimestamps*, <https://opentimestamps.org>.
10. *ProofAtlas: Condorcet winning sets (equality overthrow)*, <https://www.proofatlas.ai/collaboration/condorcet-winning-sets-equality-overthrow/>, status checked 14 August 2026, accessed 13 October 2026.
11. Haoyu Song, Thành Nguyen, and Young-San Lin, *A few good choices*, Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2026, arXiv:2506.22133, pp. 4861–4874.
12. Esther Szekeres and George Szekeres, *On a problem of Schütte and Erdős*, The Mathematical Gazette **49** (1965), no. 369, 290–293.
13. Yong Kiam Tan, Marijn J. H. Heule, and Magnus O. Myreen, *Verified propagation redundancy and compositional UNSAT checking in CakeML*, International Journal on Software Tools for Technology Transfer **25** (2023), 167–184, Checker: https://github.com/tanyongkiam/cake_lpr.
14. Itai Zilberstein, Ruben Martins, and Ratip Emin Berker, *MILP the gap: An automated reasoning approach to Condorcet’s paradox*, 2026, arXiv:2604.19851v2, 10 October 2026 (v1, 21 April 2026, with G. Li: “Is four

enough? Automated reasoning approaches and dual bounds for Condorcet dimensions of elections”, 8th Games, Agents, and Incentives Workshop (GAIW-26)).

INSTITUTE FOR RECURSIVE SELF IMPROVEMENT (IRSI)

Email address: `peter@irsi.ai`