A MONOTONE, CONDORCET CONSISTENT, 2-NM_1 TOURNAMENT RULE ON SEVEN AGENTS

KOREA SUPERINTELLIGENCE LABS

Abstract. A tournament rule turns the outcome of a round robin into a probability distribution over the players. Pennock, Schvartzman and Xue measure a rule's resistance to a two-player collusion by a selfishness parameter λ: a rule is 2-NM_λ if no two agents, by fixing the match between them, can raise their joint winning probability by more than λ times the loss suffered by the one who throws the match. They prove that no Condorcet consistent rule is 2-NM_λ for $\lambda < 1$, conjecture that $\lambda = 1$ is attainable, and report that a feasibility linear program finds such rules for tournaments on at most six agents before the search becomes too large. The question is restated as open — "a compelling open question" — in a December 2025 follow-up, the one paper on record as citing theirs.

We settle the next case. There is a tournament rule on seven agents that is monotone, Condorcet consistent and 2-NM_1. The rule is exhibited: its values are rationals with common denominator $55\,440 = 11 \cdot 7!$, given by a table of 456×7 integers indexed by the isomorphism classes of tournaments on seven agents, and the four axioms are verified directly at each of the $2\,097\,152$ labelled tournaments and at each of the $22\,020\,096$ single-match flips between them. With the published lower bound this pins the least λ admitting a monotone, Condorcet consistent, 2-NM_λ rule on seven agents to exactly 1. We also show that strengthening Condorcet consistency to top cycle consistency, which the source does not consider in this model and which the follow-up names as an open direction, costs nothing at four, five and six agents. Explicit rules and the exhaustive checks are machine-verified in Lean 4; the symmetry reduction that produced the numbers is a search device and is deliberately kept off the verified path. Nothing here bears on eight or more agents.

1. Introduction

Tournament rules and selfishness. A *tournament* on a finite set A of agents is a pair $T = (A, \succ_T)$ with $\succ_T$ a complete asymmetric relation: for every pair $i \neq j$ exactly one of $i \succ_T j$, $j \succ_T i$ holds, and it records who won the match between i and j. Write $\mathcal{T}_n$ for the set of tournaments on $A = \{0, 1, \dots, n-1\}$, so $|\mathcal{T}_n| = 2^{\binom{n}{2}}$. A *tournament rule on n agents* is a map $r\colon \mathcal{T}_n \to \Delta(A)$; the number $r_i(T)$ is the probability that the rule declares i the winner of T.

The tension this paper is about is between fairness and manipulability. On the fairness side, a rule is *Condorcet consistent* if $r_i(T) = 1$ whenever i beats everyone. On the manipulation side, one asks what two agents can gain by fixing the match between them. Pennock, Schvartzman and Xue [1] parametrise the second question by how selfish the colluders are. Their 2-NM_λ condition

$$r_i(T') + r_j(T') \;\le\; r_i(T) + r_j(T) + \lambda \max\bigl\{r_i(T) - r_i(T'),\, r_j(T) - r_j(T')\bigr\}$$

— quantified over all $i \neq j$ and all pairs $T \neq T'$ that agree on every match except the one between i and j — says, for a monotone rule, that when i throws its match to j the winner's gain is at most $\lambda + 1$ times the thrower's sacrifice; that reading is [1]'s own, recorded in the paragraph after [1, Proposition 3]. Both extremes are classical and both are recorded in [1], with the references: $\lambda = 0$ is strong non-manipulability, which no Condorcet consistent rule satisfies, and $\lambda = \infty$ is Pareto non-manipulability, which is achievable together with Condorcet consistency. The interesting range is in between, and [1, Theorem 2 and Corollary 1] locates its lower end: *no Condorcet consistent tournament rule is* 2-NM_λ *for* $\lambda < 1$.

Whether $\lambda = 1$ is attainable is [1, Conjecture 1]: "There exists a tournament rule that is monotone, Condorcet consistent, and 2-NM_1." For four agents the question is settled in [1, Figure 2] by an explicit description of the whole feasible set. Beyond that, the paper reports a computation and where it stopped:

> "Expressing and computationally solving the problem as a feasibility linear program show that such rules exist for tournaments of up to 6 agents. Unfortunately, as the number of tournaments on n agents grows exponentially with n, it became computationally difficult to check whether such rules exist for tournaments of larger size."

No rule is exhibited there for five or six agents; only the linear program's verdict is reported. Fifteen months later Pennock, Schoepflin and Wang [2] construct a rule that is monotone, Condorcet consistent and 2-NM_{11} for every n — the first Condorcet consistent rule with $\lambda = o(n)$ — and leave the sharp value open, in these words: "although we make strides toward closing the gap identified by [1], determining whether there exists a Condorcet consistent tournament rule satisfying 2-NM_1 is a compelling open question." Their introduction records the frontier as it stood: [1] "demonstrated empirically that a Condorcet consistent and 2-NM_1 rule exists for tournaments with at most six teams. Unfortunately, their experimental approach did not scale effectively to larger tournaments and did not suggest any obvious general rule."

What is proved here. We move the frontier by one agent, constructively.

Theorem 1.1. *There is a tournament rule on seven agents that is monotone, Condorcet consistent and* 2-NM_1. *One such rule is exhibited: its values are rationals with common denominator* 55 440*, and every instance of the four conditions is verified — at each of the* 2 097 152 *tournaments on seven agents and at each of the* 22 020 096 *pairs of tournaments differing in one match.*

Together with the lower bound of [1] this determines the sharp constant at $n = 7$: among monotone rules on seven agents, the least λ for which a Condorcet consistent 2-NM_λ rule exists is exactly 1. The same holds at $n = 4, 5, 6$, where the upper bound is the reproduction of [1]'s range recorded in Section 4.

The second result concerns a stronger fairness axiom. The *top cycle* $\mathrm{TC}(T)$ is the minimal set S of agents with $i \succ_T j$ for all $i \in S$, $j \notin S$; a rule is *top cycle consistent* if $r_i(T) = 0$ for $i \notin \mathrm{TC}(T)$, and this implies Condorcet consistency [1, Proposition 1]. In the 2-NM_λ model [1] studies only Condorcet consistency, and [2] names the strengthening as open: "Designing a tournament rule satisfying top cycle consistency, monotonicity, and 2-NM_λ for $\lambda = O(1)$ is an interesting and seemingly challenging open question." At small sizes the strengthening is free.

Theorem 1.2. *For each* $n \in \{4, 5, 6\}$ *there is a tournament rule on* n *agents that is monotone, top cycle consistent and* 2-NM_1*; three such rules are exhibited. Consequently, for those three* n *the least* λ *for which a monotone, top cycle consistent,* 2-NM_λ *rule on* n *agents exists is exactly* 1*, the same value as for Condorcet consistency.*

Both theorems are certificate results: what is new is a table of integers together with an exhaustive check of it, and the check is what the reader is asked to inspect. The tables were found by a linear program over isomorphism classes, using a neutrality reduction that takes the seven-agent program from $7 \cdot 2^{21} = 14\,680\,064$ variables down to 3 040; that reduction is standard, it is described in Section 6, and — this is the point of the design — *nothing in Theorems 1.1 and 1.2 depends on it.* The verification rebuilds the rule at every labelled tournament and checks that the stored representatives reach all of them, so an error in the class list, in the automorphism bookkeeping or in the averaging argument can only cause a rejection.

What is not proved here. Four disclaimers.

First, nothing here bears on $n \geq 8$, and nothing here bears on [1, Conjecture 1] in its intended form, which asks for a single rule defined for all n simultaneously. Our rules at $n = 4, \ldots, 7$ are unrelated tables; no pattern joins them, which is consistent with [2]'s report that the source's search "did not suggest any obvious general rule". Section 8 prices $n = 8$.

Second, the sharp-constant statements borrow their lower half from [1, Theorem 2] and we do not reprove it. Its proof there is written for monotone rules, which is the class our rules belong to; the extension to arbitrary rules is asserted in [1] and not used here.

Third, cover consistency — the strengthening of top cycle consistency that [2] also names — is *not* settled. Section 7 reports a floating-point measurement suggesting that no finite λ suffices already at five agents, and says explicitly why that is a measurement and not a theorem.

Fourth, the computation is not deep. What makes the seven-agent case a result is that the program the source solved has fourteen million variables, that the exact rational point had to be extracted from a polytope with empty interior, and that the resulting object is small enough to be checked in full by a third party.

Organisation. Section 2 fixes the definitions of [1] and records the linear form of 2-NM$_\lambda$ that the checker uses. Section 3 describes the certificate format and the exhaustive check. Section 4 states the results and displays the four-agent rule in full. Section 5 gives the controls. Section 6 describes the search that produced the tables, which is not part of the proofs. Section 7 collects three computations that lie outside the formal development. Section 8 prices eight agents. Section 9 describes the machine verification.

2. Preliminaries

Throughout, $n \geq 3$ and $A = \{0, 1, \dots, n-1\}$. All definitions in this section are those of [1], transcribed. Every number we attach to a statement of [1] — definition, proposition, theorem, corollary, conjecture, figure — was read off a recompilation of the arXiv v1 e-print source, the only version on arXiv; we did not have access to the ADT 2024 proceedings version, whose numbering could in principle differ.

Definition 2.1 ([1, Definitions 1 and 2])**.** A *tournament* on A is a pair $T = (A, \succ_T)$ with $\succ_T$ a complete asymmetric relation on A. A *tournament rule on n agents* is a map $r\colon \mathcal{T}_n \to \Delta(A)$, written $T \mapsto (r_0(T), \dots, r_{n-1}(T))$ with $r_i(T) \geq 0$ and $\sum_i r_i(T) = 1$.

Definition 2.2 ([1, Definitions 3, 4 and 5])**.** A rule r is *Condorcet consistent* if $r_i(T) = 1$ whenever $i \succ_T j$ for all $j \neq i$. The *top cycle* $\mathrm{TC}(T)$ is the minimal subset $S \subseteq A$ with $i \succ_T j$ for all $i \in S$ and $j \notin S$; it exists and is unique. The rule r is *top cycle consistent* if $r_i(T) = 0$ for every $i \notin \mathrm{TC}(T)$, and *cover consistent* if $r_j(T) = 0$ whenever some i *covers* j, that is $i \succ_T j$ and $j \succ_T k \Rightarrow i \succ_T k$ for all $k \neq i, j$.

Top cycle consistency implies Condorcet consistency and is implied by cover consistency [1, Proposition 1]. We use the first implication and record the standard reformulation of TC that our check is built on.

Lemma 2.3. *For a tournament T, $\mathrm{TC}(T)$ is exactly the set of agents i from which every agent is reachable by a directed path in $\succ_T$.*

Proof. Contracting the strongly connected components of $\succ_T$ yields a tournament without directed cycles on the components, that is a linear order $S_1 \succ S_2 \succ \cdots \succ S_k$ in which every agent of S_p beats every agent of S_q for $p < q$. A set S satisfies $i \succ_T j$ for all $i \in S$, $j \notin S$ exactly when S is a union $S_1 \cup \cdots \cup S_p$, so the minimal non-empty such set is S_1. An agent of S_1 reaches every agent, since it reaches all of S_1 by strong connectivity and beats everything below; an agent of S_p with $p > 1$ reaches nothing in S_1. □

Definition 2.4 ([1, Definitions 7, 10 and 12])**.** Tournaments $T, T' \in \mathcal{T}_n$ are $\{i, j\}$-*adjacent* if they agree on every match except possibly the one between i and j. A rule r is *monotone* if $r_i(T) \geq r_i(T')$

for all $i \neq j$ and all $\{i,j\}$-adjacent $T \neq T'$ with $i \succ_T j$. For $\lambda \geq 0$, r is 2-*non-manipulable for* λ (2-NM$_\lambda$) if

$$r_i(T') + r_j(T') \;\leq\; r_i(T) + r_j(T) + \lambda \max\bigl\{r_i(T) - r_i(T'),\; r_j(T) - r_j(T')\bigr\} \tag{1}$$

for all $i \neq j$ in A and all $\{i,j\}$-adjacent $T \neq T'$ in $\mathcal{T}_n$.

One point of transcription needs saying, since everything below is quantified over adjacent pairs. [1, Definition 7] defines S-adjacency for a general $S \subseteq A$ by the displayed condition "$i \succ_T j \iff i \succ_{T'} j$ for $i \neq j \in [n] \setminus S$", which constrains only the matches with *both* ends outside S, and then glosses it as: "T and T' are S-adjacent if they coincide on every match except possibly those between agents in S." For $|S| = 2$ the gloss is the stronger of the two, and it is the gloss we have transcribed above. It is the standard reading — [3] defines S-adjacency as "they only differ on the outcomes of some subset of games played between members of S"; it is what makes 2-NM$_\lambda$ say what [1] says it says, namely that two agents fix the match between them; and it is the reading under which [1, Proposition 3] is stated. Under the weaker reading there would be more adjacent pairs, so monotonicity and 2-NM$_\lambda$ would be strictly stronger conditions, and nothing below would apply to them.

Two $\{i,j\}$-adjacent tournaments $T \neq T'$ then differ in exactly the match between i and j; if $i \succ_T j$ then $j \succ_{T'} i$, and we call (T, T') a *flip at* $\{i,j\}$. Write

$$a \;=\; r_i(T) - r_i(T'), \qquad b \;=\; r_j(T') - r_j(T)$$

for the sacrifice of the agent who throws the match and the gain of the agent who receives it. [1, Proposition 3] is the statement that a 2-NM$_\lambda$ rule, $\lambda > 0$, is monotone if and only if $a \leq (1+\lambda)b$ at every flip; the following packages all four inequalities in the form the check of Section 3 uses, and its proof is the one-line reading of (1) that [1] performs to obtain that proposition.

Lemma 2.5 (linear form). *Let $\lambda \geq 0$ and let r be a rule on n agents. Then r is monotone and* 2-NM$_\lambda$ *if and only if for every flip (T, T') at every pair $\{i,j\}$,*

$$a \geq 0, \qquad b \geq 0, \qquad b \leq (1+\lambda)a, \qquad a \leq (1+\lambda)b. \tag{2}$$

Proof. Suppose r is monotone and 2-NM$_\lambda$, and let (T,T') be a flip at $\{i,j\}$ with $i \succ_T j$. Monotonicity applied to (T, T', i, j) gives $a \geq 0$ and applied to (T', T, j, i) gives $b \geq 0$; hence $\max\{r_i(T) - r_i(T'), r_j(T) - r_j(T')\} = \max\{a, -b\} = a$, and (1) for the ordered pair (T, T') reads $b - a \leq \lambda a$, that is $b \leq (1+\lambda)a$. Exchanging the roles of the two tournaments, $\max\{r_j(T') - r_j(T), r_i(T') - r_i(T)\} = \max\{b, -a\} = b$ and (1) for (T', T) reads $a \leq (1+\lambda)b$. Conversely, assume (2) at every flip. Monotonicity is the inequality $a \geq 0$, read at the flip determined by the pair; and for any $\{i,j\}$-adjacent $T \neq T'$, one of the two orderings is a flip, in which case (1) is $b \leq (1+\lambda)a$ or $a \leq (1+\lambda)b$ as computed above. ☐

3. The certificate format

Encoding. Index the $\binom{n}{2}$ pairs (u,v) with $u < v$ in row-major order, so that $(0,1), (0,2), \ldots, (0, n-1), (1,2), \ldots$ receive the indices $0, 1, 2, \ldots$. A tournament on A is then encoded by an integer c with $0 \leq c < 2^{\binom{n}{2}}$: bit k of c is 1 exactly when, in the k-th pair (u,v), one has $u \succ v$. This is a bijection between codes and tournaments. The symmetric group S_n acts on codes through its action on A, and a single-match flip is an exclusive-or with a power of two.

A certificate, and what is checked. A *certificate* for n and a rational $1 + \lambda = \mu_N/\mu_D$ consists of a positive integer D (the common denominator), a list $c_1, \ldots, c_m$ of codes (the *representatives*) and an array $(N_{s,i})$ of non-negative integers, $1 \leq s \leq m$ and $i \in A$. It determines a rule as follows: for each s and each permutation $\sigma \in S_n$, the code of $\sigma(T_{c_s})$ is recorded together with σ^{-1}, and if the code c carries the record (s, σ) then

$$r_i(T_c) \;=\; \frac{N_{s,\sigma^{-1}(i)}}{D}.$$

The check then runs over *every* code $c < 2^{\binom{n}{2}}$ and verifies:

(C0) *coverage*: some representative and permutation produced c, so r is defined at T_c;
(C1) $\sum_{i\in A} N_{s,\sigma^{-1}(i)} = D$, so $r(T_c)$ is a probability distribution — non-negativity is automatic, the entries being natural numbers;
(C2) Condorcet consistency at T_c;
(C3) for each pair $u < v$ such that $u \succ v$ in T_c, the four inequalities (2) of Lemma 2.5 at the flip $(T_c, T_{c'})$, where c' is c with the bit of (u, v) reversed, in the cleared form $\mu_D\, b \le \mu_N\, a$ and $\mu_D\, a \le \mu_N\, b$ over the numerators;

and, in the top cycle variant, additionally

(C4) $N_{s,\sigma^{-1}(i)} = 0$ for every i that does not reach every agent in T_c, which by Lemma 2.3 is top cycle consistency at T_c.

Every flip is inspected once, from the side on which the match is won; a pair $\{u, v\}$ with $v \succ u$ in T_c is skipped at c because it is the same flip seen at c'.

Lemma 3.1 (transport)**.** *If the check (C0)–(C3) succeeds for n, $D > 0$ and $\mu_N/\mu_D = 2$, then the rule r determined by the certificate is a probability distribution at every tournament in $\mathcal{T}_n$, Condorcet consistent, monotone and* 2-NM$_1$*. With (C4) it is in addition top cycle consistent.*

Proof. Immediate from Lemma 2.5 and Lemma 2.3, once (C0) is read as saying that r is total: the check is a conjunction over all codes and all pairs, and the codes biject with $\mathcal{T}_n$, so each condition of Section 2 is verified at each of its instances. The first sentence is formalised (Section 9); the addition of (C4) is not, and is the elementary step just described applied to one further clause. □

Remark 3.2. The representatives are one tournament from each isomorphism class, and the entries $N_{s,\cdot}$ are constant on the orbits of $\mathrm{Aut}(T_{c_s})$ on A, which makes the rule neutral. Neither fact is verified, and neither is used: the check quantifies over all labelled tournaments and (C0) is what rules out a table that misses a class. If the list of representatives were incomplete, or an automorphism orbit mishandled, or the whole reduction of Section 6 unsound, the check would fail; it cannot succeed vacuously.

4. Results

The published range, with explicit rules. The existence of monotone, Condorcet consistent, 2-NM$_1$ rules for $n \le 6$ is [1]'s, obtained there by a feasibility linear program. A rule is printed there only for $n = 4$, and none for $n = 5, 6$. We record explicit certificates for all three sizes, as the control on the encoding used at $n = 7$ and because a rule one can write down is more use than a verdict.

Proposition 4.1. *For each $n \in \{4, 5, 6\}$ the certificate with common denominator $n!$ — that is* 24*,* 120 *and* 720 *— listed in the accompanying development passes the check (C0)–(C3) at every one of the $2^{\binom{n}{2}}$ labelled tournaments. Hence, by Lemma 3.1, for each such n there is a monotone, Condorcet consistent,* 2-NM$_1$ *tournament rule on n agents.*

Proof, and what was computed. Exhaustive evaluation of the check, one n at a time: 64, 1 024 and 32 768 codes, carrying 192, 5 120 and 245 760 flips respectively, each contributing the four inequalities (2). Nothing is sampled. The certificates have 4, 12 and 56 representatives, the numbers of isomorphism classes of tournaments on 4, 5, 6 vertices [5]. □

At $n = 4$ the certificate is small enough to print in full. In the encoding of Section 3 the four classes are represented by the codes 0, 2, 4 and 5:

code	the tournament	$(r_0, r_1, r_2, r_3) \cdot 24$	
0	the transitive tournament $3 \succ 2 \succ 1 \succ 0$	$(0, 0, 0, 24)$	Condorcet winner 3
2	3 beats all; $0 \succ 2 \succ 1 \succ 0$	$(0, 0, 0, 24)$	Condorcet winner 3
4	$0 \succ 3 \succ 1 \succ 0$, and 2 beats $0, 1$ but loses to 3	$(7, 2, 4, 11)$	strongly connected
5	1 loses to all; $0 \succ 3 \succ 2 \succ 0$	$(8, 0, 8, 8)$	uniform on $\{0, 2, 3\}$

The rule at any other tournament on four agents is obtained by relabelling. In the strongly connected class the agents 0 and 1 have one win each and the agents 2 and 3 have two; the entry $(7, 2, 4, 11)/24$ is a point of the feasible set described in [1, Figure 2], namely $\frac{13}{16}$ of $(\frac{17}{39}, \frac{7}{39}, \frac{4}{39}, \frac{11}{39})$ plus $\frac{3}{16}$ of $(\frac{5}{9}, \frac{1}{9}, 0, \frac{3}{9})$, read in the reverse coordinate order (see Remark 7.1).

Seven agents.

Theorem 4.2. *There is a tournament rule on seven agents that is monotone, Condorcet consistent and* 2-NM_1*.*

Proof, and what was computed. The certificate has $D = 55\,440 = 11 \cdot 7!$, the 456 representatives of the isomorphism classes of tournaments on seven vertices [5], and $456 \times 7 = 3\,192$ non-negative integer numerators, each row summing to 55 440. The check (C0)–(C3) of Section 3 was evaluated at every one of the $2^{21} = 2\,097\,152$ labelled tournaments: at each, the coverage clause, the distribution clause, Condorcet consistency, and the four inequalities (2) at each of the 21 pairs of agents — so, over all codes, at each of the $2^{21} \cdot 21/2 = 22\,020\,096$ flips. Nothing is sampled and no tournament is skipped by an invariant. It succeeded; Lemma 3.1 converts that into the statement. The whole sweep costs 35.3 seconds of CPU time inside the proof assistant with the check compiled to a shared library (Section 9); an independently written C program, sharing no code and reading the certificate from a text file, reproduces the verdict in 1.1 seconds.

The completeness of the list of 456 representatives is *not* part of the proof: the coverage clause (C0) verifies directly that the relabelled representatives reach all 2 097 152 codes. How the numerators were found is Section 6, and plays no role here. □

Corollary 4.3. *For* $4 \leq n \leq 7$*, the least* λ *for which there exists a monotone, Condorcet consistent,* 2-NM_λ *tournament rule on* n *agents is exactly* 1*.*

Proof. The upper bound is Proposition 4.1 for $n \leq 6$ and Theorem 4.2 for $n = 7$, together with the observation that 2-NM_λ is monotone in λ. The lower bound is [1, Theorem 2] with $\alpha = 0$ — obtained there from the construction of [3] — whose three displayed inequalities are read at a single tournament on $[n]$ carrying a 3-cycle each of whose agents can be made undefeated by one flip, so the bound is available at each fixed $n \geq 3$; it is proved there for monotone rules, which is the class in question here. □

Top cycle consistency at small sizes.

Theorem 4.4. *For each* $n \in \{4, 5, 6\}$ *there is a tournament rule on* n *agents that is monotone, top cycle consistent and* 2-NM_1*. Consequently, for those* n *the least* λ *admitting a monotone, top cycle consistent,* 2-NM_λ *rule on* n *agents is exactly* 1*.*

Proof, and what was computed. The check (C0)–(C4), with the top cycle clause in the reachability form of Lemma 2.3, was evaluated at every labelled tournament for $n = 4, 5, 6$. At $n = 4$ and $n = 5$ the certificates of Proposition 4.1 pass it unchanged: those tables turn out to be top cycle consistent already. At $n = 6$ the Condorcet certificate does not, and it fails at exactly one place: in the class of the representative with code 19 the agent 2 receives 7/720 while lying outside the top cycle. A different certificate is used there, with the same 56 representatives and common denominator 360. For the sharpness, top cycle consistency implies Condorcet consistency [1, Proposition 1], so [1, Theorem 2] applies verbatim and gives $\lambda \geq 1$ as in Corollary 4.3. □

Remark 4.5. This says nothing about general n, which is the form in which [2] poses the question. What it does say is that the obstruction, if there is one, does not appear at the sizes where the Condorcet question is currently decided: the two columns agree, at the value 1, everywhere they are both known. Pennock, Schoepflin and Wang obtain a top cycle consistent, monotone rule for all n in a different, multiplicative model of non-manipulability [2], which is not comparable with 2-NM_λ term by term and is not used here.

5. Controls

A check that certifies a positive — "this table satisfies all four axioms" — fails safe in the wrong direction if some clause is vacuous or some parameter dead. The following four refutations close that gap; each is a claim whose truth would make the results above weaker than they look, and each is evaluated exhaustively and comes out false.

Proposition 5.1. *With $n = 4$ and the certificate of Section 4:*

1. *the check fails for $1 + \lambda = 3/2$, that is the exhibited four-agent rule is not 2-NM$_{1/2}$;*
2. *the check fails for the table obtained by moving one unit of probability inside the strongly connected class, from $(7, 2, 4, 11)$ to $(8, 2, 4, 10)$;*
3. *the check fails for the uniform table, every entry 6 over 24;*
4. *the check fails, at the coverage clause, for the list of representatives with the class of code 5 deleted.*

Item (1) says the selfishness parameter is live: the positive results do not silently prove more than 2-NM$_1$, as of course they cannot, since $\lambda \geq 1$ is [1, Corollary 1]. Item (2) says the check is not satisfied by every probability table. Item (3) isolates the Condorcet clause: the uniform rule is monotone and 2-NM$_\lambda$ for every λ, and it is the fairness clause alone that rejects it. Item (4) says that an incomplete list of isomorphism classes cannot pass unnoticed, which is what makes Remark 3.2 true.

6. How the tables were found

This section describes the search. None of it is used in Section 4, and none of it is machine-checked; it is recorded because it is the reason the seven-agent case is reachable at all. The argument is a standard convexity-and-symmetry averaging and we claim no novelty for it, only that this is not how the program of [1] was set up, and that the prior paper closest to an exhaustive approach discounts one: [4] observes that "the space of tournament rules for n teams lies in $n2^{\binom{n}{2}}$-dimensional space, and it is hard to imagine a successful exhaustive search beyond $n = 10$".

By Lemma 2.5, for fixed n and λ the set

$$P_n(\lambda) = \left\{ r \in \mathbb{R}^{n \cdot 2^{\binom{n}{2}}} \ : \ r \text{ is a rule, is Condorcet consistent, and satisfies (2) at every flip} \right\}$$

is a polytope: every defining condition is linear, the max in (1) having been removed by monotonicity. Now S_n acts on tournaments by $\sigma(i) \succ_{\sigma T} \sigma(j) \iff i \succ_T j$ and on rules by $(\sigma \cdot r)_i(T) = r_{\sigma^{-1}(i)}(\sigma^{-1}T)$, and each defining condition of $P_n(\lambda)$ is equivariant. Hence $P_n(\lambda)$ is S_n-invariant, and being convex it contains $\bar{r} = \frac{1}{n!}\sum_{\sigma \in S_n} \sigma \cdot r$ whenever it contains r. The average $\bar{r}$ is *neutral*: $\bar{r}_{\sigma(i)}(\sigma T) = \bar{r}_i(T)$. A neutral rule is determined by one number per pair (isomorphism class, orbit of Aut on the agents), and by equivariance it is enough to impose the constraints at those pairs. So $P_n(\lambda)$ is non-empty if and only if a much smaller polytope $Q_n(\lambda)$ is, of the following sizes.

n	classes [5]	labelled tournaments	variables of Q_n	variables of P_n
4	4	64	12	256
5	12	1 024	48	5 120
6	56	32 768	296	196 608
7	456	2 097 152	3 040	14 680 064

The class representatives come from `nauty-gentourng` [6], each replaced by the lexicographically least code in its S_n-orbit so that representatives are canonical; the identity $\sum_c n!/|\operatorname{Aut}(T_c)| = 2^{\binom{n}{2}}$ was checked at every $n \leq 8$ as a completeness test on the class list, and the variable counts $12, 48, 296, 3\,040$ in the table were recomputed directly from the representatives. At $n = 7$ the reduced program has 3 040 variables, 512 equalities and 17 408 inequalities. The equalities are one normalisation row $\sum_i r_i(T_{c_s}) = 1$ per class and one row $r_w(T_{c_s}) = 1$ per class with a Condorcet winner w; deleting the winner is a bijection from the second kind of class onto the classes on six agents, so their number is $456 + 56 = 512$ by [5]. The inequalities are the four of (2) at each of the $456 \cdot 21 = 9\,576$ (representative, pair) slots, that is 38 304 rows, of which 17 408 survive the removal of duplicates. Both counts, and the corresponding 27, 160, 1 338 at $n = 4, 5, 6$, were recomputed for this paper by rebuilding the program from the class list.

Extracting an *exact* rational point of $Q_7(1)$ was the one awkward step, and the recipe is worth recording. The polytope has empty interior — many values are forced by implicit equalities — so rounding a floating-point solution does not work directly, and taking every constraint tight at one vertex (3 942 of them) makes exact elimination blow up on fill-in. What worked: solve the linear program four times with different objectives (one plain, three with a small random linear term), intersect the four tight sets, which leaves 1 536 rows; eliminate that system exactly in rational arithmetic (925 pivots, 2 115 free coordinates, maximum row width 12); set the free coordinates to the rounded average of the four solutions over the denominator D; and verify every row exactly. With $D = 55\,440 = 11 \cdot 7!$ this succeeds; smaller denominators left between 12 and 991 violated rows.

An implementation error in this pipeline — a permutation used where its inverse was needed — survived the class-level program and was caught only by the labelled check, which is a concrete reason for the discipline of Remark 3.2.

7. Three computations outside the formal development

The following are recorded as computations, not as theorems; none of them is machine-checked and nothing above depends on any of them.

Remark 7.1 (the four-agent polytope of [1]). [1, Figure 2] states necessary and sufficient conditions for a rule on four agents to be Condorcet consistent and 2-NM_1: probability 1 on the winner in the two classes with a Condorcet winner, uniform over the three-cycle in the class with a Condorcet loser, and in the strongly connected class a convex combination of six explicit distributions. Exact rational vertex enumeration of $Q_4(1)$ — all triples of tight rows, in `Fraction` arithmetic — returns exactly six vertices in that class,

$$\begin{gathered}\left(\tfrac{1}{3}, 0, \tfrac{2}{9}, \tfrac{4}{9}\right), \quad \left(\tfrac{1}{3}, 0, \tfrac{1}{9}, \tfrac{5}{9}\right), \quad \left(\tfrac{10}{33}, \tfrac{2}{33}, \tfrac{8}{33}, \tfrac{13}{33}\right), \\ \left(\tfrac{7}{24}, \tfrac{1}{48}, \tfrac{13}{48}, \tfrac{5}{12}\right), \quad \left(\tfrac{5}{21}, \tfrac{1}{21}, \tfrac{4}{21}, \tfrac{11}{21}\right), \quad \left(\tfrac{11}{39}, \tfrac{4}{39}, \tfrac{7}{39}, \tfrac{17}{39}\right),\end{gathered}$$

which is the list of [1, Figure 2] with each vector read in the reverse coordinate order, and forces $(\frac{1}{3}, 0, \frac{1}{3}, \frac{1}{3})$ in the Condorcet-loser class. (The caption asks only for Condorcet consistency and 2-NM_1, whereas $Q_4(1)$ imposes monotonicity as well; that the two vertex lists coincide says that at $n = 4$ the extra axiom costs nothing.) Six four-vectors with denominators $9, 33, 48, 21, 39$ agreeing is not a coincidence: this is the evidence that the axioms as encoded in Section 3 are the axioms of [1], and it was run before anything was computed at $n \geq 5$.

Remark 7.2 (no rule on four agents is a function of the score sequence). Every one of the six vertices above has $r_0 - r_1 \geq 7/39 > 0$, the minimum being attained at the last of them. In the strongly connected tournament on four agents the agents 0 and 1 both have exactly one win, so no monotone, Condorcet consistent, 2-NM_1 rule on four agents can give equal probabilities to agents with equal scores: already at the smallest interesting size, the number of wins is not enough information. The two agents are of course distinguishable — in the certificate of Section 4, agent 0 beats an agent with two wins and agent 1 beats an agent with one — and the exhibited rule separates them by 5/24. This is an immediate corollary of [1, Figure 2] and is not new; it is recorded because it is the smallest concrete obstruction to the "obvious general rule" whose absence [2] remarks on.

Remark 7.3 (cover consistency: a measurement, not a theorem). Cover consistency was introduced, in a different model of non-manipulability, by [4]. Replacing Condorcet consistency by cover consistency in $Q_n(\mu - 1)$ and testing feasibility by maximising the minimum slack, with the HiGHS simplex solver in floating point, the program is infeasible at $n = 5$ and $n = 6$ for every $\mu = 1 + \lambda$ tried up to 10^7, while the optimal minimum slack tends to zero from below: $-1.8 \cdot 10^{-2}$, $-9.2 \cdot 10^{-4}$ and $-8.5 \cdot 10^{-6}$ at $\mu = 2, 10, 100$. The honest reading is "no finite λ appears to work, with a vanishing margin". It is not a theorem: there is no exact certificate at any single μ and no argument parametric in μ, and a vanishing margin is exactly the regime in which floating point is least trustworthy. A search for the qualitative obstruction — a flip at which monotonicity and cover consistency together force $a = 0$ while $b > 0$, which would kill every λ at once — found none at $n = 5$ or $n = 6$, so any proof will have to be quantitative. For $n = 3$ and $n = 4$ the cover-consistent program is feasible, and bisection on μ brackets its threshold at $\mu^* \in [1.999425, 2.000008]$, the same value $\mu^* = 2$ that the same bisection returns for Condorcet and for top cycle consistency at every $n \leq 6$. All of these bisection figures are floating point and none of them is used above: the exact statements of Section 4 come from the certificates, and the exact lower bound from [1, Theorem 2].

8. The frontier at eight agents

There are 6 880 isomorphism classes of tournaments on eight vertices [5], and running the count of Section 6 over them — which also confirms the identity $\sum_c 8!/|\operatorname{Aut}(T_c)| = 2^{28}$ — gives the reduced program of Q_8 exactly 54 256 variables and $4 \cdot 6\,880 \cdot 28 = 770\,560$ inequalities before duplicates are removed, against 3 040 and 38 304 at $n = 7$. That is larger than the seven-agent program, which takes about 70 seconds of CPU time, but not by a factor that matters. The obstacle is on the other side. The verification of Section 3 would sweep all $2^{28} = 268\,435\,456$ labelled tournaments, 128 times as many codes as at $n = 7$ and 28 rather than 21 pairs at each — extrapolating the measured 35.3 CPU-seconds of the seven-agent sweep, on the order of two CPU-hours, which is affordable — but it builds a lookup table with one machine word per labelled tournament, some two gigabytes inside the proof assistant, which is not. A check at the level of isomorphism classes would avoid that, but it would then depend on the completeness of the 6 880-element class list, which at $n = 8$ is out of reach of the direct enumeration that (C0) performs at $n = 7$; it would rest instead on the orbit-count over a group of 40 320 elements, which is exactly the piece of bookkeeping that Remark 3.2 keeps off the trusted path at $n \leq 7$. Independently, extracting an exact rational point at that size is unlikely to be routine: at $n = 7$ it already required the four-objective intersection recipe of Section 6. We have not attempted $n = 8$: the class count, the variable count and the row count above were computed, but no program was solved and no sweep was run at that size, so the time and memory figures are extrapolations from the measured $n \leq 7$ runs and nothing in this section is a claim about eight agents.

Nothing in this paper suggests what the answer at $n = 8$, or in general, should be. The four certificates exhibited here have denominators 24, 120, 720 and 55 440 and no visible family resemblance.

9. Verification

Machine-checked in Lean 4 [7] with Mathlib [8] are: the exhaustive checks behind Proposition 4.1, Theorem 4.2, Theorem 4.4 and Proposition 5.1, in each case as an evaluation of the decision procedure of Section 3 over every labelled tournament of the relevant size; and the transport of Lemma 3.1, in the form stated there, from the check to the statement "there exists a rule that is a distribution at every tournament, is Condorcet consistent, is monotone and is 2-NM$_1$" phrased in the definitions of Section 2, which are transcriptions of [1, Definitions 1, 2, 3, 7, 10 and 12]. Theorem 4.2 is that statement at $n = 7$. Not formalised, and written out above as ordinary mathematics, are: Lemma 2.3; Lemma 2.5, of which only the direction used — from (2) to monotonicity and 2-NM$_1$ — is part of the formalised transport; the top cycle half of Lemma 3.1, so that Theorem 4.4 rests on a machine-checked evaluation of (C0)–(C4) plus that elementary reading of clause (C4); Corollary 4.3 and the sharpness half of Theorem 4.4, which combine our certificates with [1, Theorem 2 and Proposition 1]; the whole of Section 6; and the three computations of Section 7, which are labelled as measurements where they occur. There is no `sorry` anywhere in the development.

The modules carrying the check import no library at all: a tournament is a natural number, a certificate is two strings of decimal digits, and every object in sight is a natural number or an array of them. That is what allows the check to be compiled to a shared library and run by Lean's compiled evaluator (`native_decide`), which is the only trusted step and the reason the seven-agent sweep costs 35.3 CPU-seconds rather than an estimated ten to twenty CPU-minutes interpreted. The audit that matters is therefore the axiom list. The seven-agent theorem and the evaluation it rests on depend on `propext`, `Classical.choice`, `Quot.sound` and exactly one evaluation axiom, the one generated by that single sweep; each smaller check depends on the same three plus its own single evaluation axiom. The transport lemma itself declares no evaluation axiom — it depends only on `propext`, `Classical.choice` and `Quot.sound` — so every appeal to the compiled evaluator in the paper is one of the finite sweeps and none is hidden inside a proof. The bookkeeping facts about the pair indexing of Section 3 at $n = 7$ are discharged by the kernel's own decision procedure, with no evaluation axiom at all. An independently written C program, sharing no code and no data path with the Lean development, reproduces every verdict reported above. Timings were measured on a shared machine under load and are quoted as CPU seconds. The repository is private; repository reference to be added at submission.

References

[1] D. Pennock, A. Schvartzman and E. Xue, *Toward fair and strategyproof tournament rules for tournaments with partially transferable utilities*, in: R. Freeman and N. Mattei (eds.), Algorithmic Decision Theory — 8th International Conference, ADT 2024, New Brunswick, NJ, USA, October 14–16, 2024, Proceedings, Lecture Notes in Computer Science **15248**, Springer, Cham, pp. 174–188; `doi:10.1007/978-3-031-73903-3_12`. E-print: arXiv:2408.10346v1 (cs.GT), 19 August 2024. All definition, proposition, theorem, corollary, conjecture and figure numbers cited above are those of the arXiv v1 e-print — the only version on arXiv, and the only version we read — as obtained by recompiling its source; in particular Definition 7 is $\{i, j\}$-adjacency, Definition 10 is 2-NM$_\lambda$, Definition 12 is monotonicity, Proposition 1 is the fairness hierarchy, Proposition 3 is the monotonicity criterion behind Lemma 2.5, Theorem 2 with its Corollary 1 is the lower bound $\lambda \geq 1$, Conjecture 1 is the attainability of $\lambda = 1$, and Figure 2 is the four-agent characterisation.

[2] D. Pennock, D. Schoepflin and K. Wang, *Strategyproof tournament rules for teams with a constant degree of selfishness*, in: V. Bilò, Y. Cai and Z. Huang (eds.), Web and Internet Economics — 21st International Conference, WINE 2025, New Brunswick, NJ, USA, December 8–11, 2025, Proceedings, Lecture Notes in Computer Science **16266**, Springer, Cham, pp. 502–520; `doi:10.1007/978-3-032-18660-7_27`. E-print: arXiv:2512.05235v1 (cs.GT), 4 December 2025. The quotations above are from the introduction and the conclusions of that e-print, and from its appendix on top cycle consistency. On 3 September 2026, when the literature check behind this paper was last run, [1] had exactly one arXiv version, this paper had exactly one, and this was the only paper recorded anywhere as citing [1].

[3] J. Schneider, A. Schvartzman and S. M. Weinberg, *Condorcet-consistent and approximately strategyproof tournament rules*, in: 8th Innovations in Theoretical Computer Science Conference (ITCS 2017), Leibniz International Proceedings in Informatics **67**, Schloss Dagstuhl, 2017, pp. 35:1–35:20; `doi:10.4230/LIPIcs.ITCS.2017.35`; arXiv:1605.09733. The lower-bound construction reused by [1, Theorem 2] is theirs.

[4] A. Schvartzman, S. M. Weinberg, E. Zlatin and A. Zuo, *Approximately strategyproof tournament rules: on large manipulating sets and cover-consistence*, in: 11th Innovations in Theoretical Computer Science Conference (ITCS 2020), Leibniz International Proceedings in Informatics **151**, Schloss Dagstuhl, 2020, pp. 3:1–3:25; `doi:10.4230/LIPIcs.ITCS.2020.3`; arXiv:1906.03324v2 (cs.GT). This is the prior paper that joins cover consistency to incentive compatibility; it works in the k-SNM-α model, which is not the model of the present paper, and it does not use a neutrality reduction, remarking instead that "the space of tournament rules for n teams lies in $n2^{\binom{n}{2}}$-dimensional space, and it is hard to imagine a successful exhaustive search beyond $n = 10$" (quoted from v2, the post-ITCS camera-ready e-print).

[5] OEIS Foundation Inc., *Sequence A000568: number of outcomes of unlabeled n-team round-robin tournaments*, The On-Line Encyclopedia of Integer Sequences, `https://oeis.org/A000568`. The values used above are $a(4) = 4$, $a(5) = 12$, $a(6) = 56$, $a(7) = 456$ and $a(8) = 6\,880$; the ones at $n \leq 8$ were reproduced here with `nauty-gentourng` [6].

[6] B. D. McKay and A. Piperno, *Practical graph isomorphism, II*, J. Symbolic Comput. **60** (2014), 94–112; `doi:10.1016/j.jsc.2013.09.003`. The program used here is `nauty-gentourng`, and only in the search of Section 6.

[7] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: A. Platzer and G. Sutcliffe (eds.), Automated Deduction — CADE 28, Lecture Notes in Computer Science **12699**, Springer, Cham, 2021, pp. 625–635; `doi:10.1007/978-3-030-79876-5_37`.

[8] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), ACM, 2020, pp. 367–381; arXiv:1910.09336.