RANK-MAXIMAL ASSIGNMENTS AND THE UNCOVERED SET: THE MCKELVEY AND BORDES INCLUSIONS THROUGH SEVEN AGENTS, AND THE FAILURE OF THE GILLIES VARIANT FROM FOUR

KOREA SUPERINTELLIGENCE LABS

Abstract. In the assignment (house allocation) domain of n agents, n houses and strict preferences, an assignment μ majority dominates λ when more agents prefer their house under μ than under λ; because this relation has ties, there are three covering relations — Bordes, Gillies and McKelvey — and three uncovered sets. Brandt, Chen, Dong, Lederer and Schlenga (arXiv:2602.14816) report that for $n \leq 5$ every rank-maximal assignment lies in the McKelvey uncovered set, and leave the general case open. We show the following. The McKelvey inclusion holds for $n = 6$ and $n = 7$, and in the stronger Bordes form; moreover the hypothesis can be weakened from rank-maximality to *local* rank-maximality — no exchange of two agents' houses and no cyclic exchange among three agents improves the count vector — and the conclusion still holds for $n \leq 7$; exchanges of two agents alone suffice for $n \leq 4$ and fail at $n = 5$, while the covering quantifier cannot be localised in the same way — at $n = 6$ a witness against a covering may have to move five of the six agents. These statements for $n \geq 5$ rest on unsatisfiability certificates from a SAT solver (one instance per conjugacy class of $\mathrm{Sym}(n)$, ten at $n = 6$ and fourteen at $n = 7$) and are not formalised; the statements for $n \leq 4$, the $n = 5$ sharpness witness, and a census of the 331 776 four-agent profiles are verified in Lean 4 from the definitions. The Gillies variant behaves differently: from $n = 4$ on there are profiles with a rank-maximal assignment outside the Gillies uncovered set — explicit witnesses for $n = 4, 5, 6, 7$ are verified in Lean, $n = 4$ is the smallest such size, exactly 9 504 of the 331 776 four-agent profiles are of this kind, and a padding construction gives a witness for every $n \geq 4$. The mechanism is the one the source names: a popular assignment Gillies covers everything it dominates. The general question remains open; we record why the obvious local arguments cannot settle it.

1. Introduction

The objects. There are n agents $N = \{1, \ldots, n\}$ and n houses H, and each agent x has a strict linear order $\succ_x$ over H; the profile $P = (\succ_1, \ldots, \succ_n)$ is the input. An *assignment* is a bijection $\mu \colon N \to H$, and $\mathcal{M}$ is the set of all $n!$ of them. Two assignments are compared by majority: μ *strictly majority dominates* λ, written $\mu \succ \lambda$, when strictly more agents prefer their house under μ to their house under λ than the other way round, agents receiving the same house being counted on neither side. This relation is asymmetric but has ties, and is not transitive; the *majority graph* on $\mathcal{M}$ is the object studied by Brandt, Chen, Dong, Lederer and Schlenga [1], who transfer the classical majoritarian solution concepts of social choice — the top cycle and the uncovered set — to this domain.

Because of the ties there is not one covering relation but three, all classical (Fishburn [5], Miller [6], Bordes [7], McKelvey [8], Gillies [9], Moulin [10]; Duggan [13] and Brandt, Brill and Harrenstein [14, 15] survey them), and the source uses all three: μ *Bordes covers* λ if $\mu \succ \lambda$ and every η with $\lambda \succ \eta$ has $\mu \succ \eta$; μ *Gillies covers* λ if $\mu \succ \lambda$ and every η with $\eta \succ \mu$ has $\eta \succ \lambda$; and μ *McKelvey covers* λ if it does both. The corresponding *uncovered sets* $\mathrm{UC_B}(P)$, $\mathrm{UC_G}(P)$, $\mathrm{UC_M}(P)$ consist of the assignments that nothing covers; the source writes UC for the McKelvey one and notes that the Bordes and the Gillies uncovered sets are refinements of it.

Two assignment rules from the matching literature enter. An assignment is *popular* (Gärdenfors [4]; Abraham, Irving, Kavitha and Mehlhorn [3]) if no assignment strictly majority dominates it; popular assignments need not exist. An assignment is *rank-maximal* (Irving, Kavitha, Mehlhorn, Michail and Paluch [2]) if it lexicographically maximises the number of agents receiving their first-ranked house, then the number receiving their second-ranked house, and so on; rank-maximal assignments always exist and can be computed in polynomial time.

The source and its question. The relevant part of [1] is its Section 4.2, "Uncovered Sets", whose Remark 6 reads, verbatim:

> "Our experiments suggest that, at least for small n, all rank-maximal assignments are contained in the uncovered set. We have verified this through exhaustive search for all profiles with $n \leq 5$. If this were true in general, one could obtain an element of the uncovered set by computing a rank-maximal assignment, which is possible in polynomial time. A set inclusion between the two rules would be interesting, as then UC would be a natural rule that is relatively decisive and yet contains all rank-maximal *and* all popular assignments."

Its Remark 5 lists the complexity of finding an uncovered assignment, and of deciding whether a given assignment is uncovered, as open; the remark above is the source's candidate route to the first of these. The source's own exhaustive computations stop at $n = 5$ (its $n = 6$ and $n = 7$ figures are from sampled profiles), so the first open case of the remark is $n = 6$. (Section, remark and example numbers of [1] are those of the compiled e-print of version 1, the only version as of 2026-09-07; see the bibliography.)

What is settled here. Write $\lambda \succ \mu$ for strict majority domination and $c(\mu)$ for the count vector of μ (Section 2). Everything below is stated for the n-agent, n-house domain with strict preferences, exactly as in the source.

- **The inclusion, through seven agents** (Theorems 3.2 and 3.3, Computations 5.1 and 5.2). For $n \leq 7$, every rank-maximal assignment is Bordes uncovered, hence McKelvey uncovered. For $n \leq 4$ this is verified in Lean from the definitions over all 216 and all 331 776 profiles; $n = 5$ is the source's own exhaustive claim, reproduced here from the definitions (Computation 4.6); for $n = 6$ and $n = 7$ it is established by unsatisfiability certificates from the SAT solver CaDiCaL, after a symmetry reduction to one instance per conjugacy class of $\mathrm{Sym}(n)$, and is not formalised.
- **A local hypothesis suffices.** Call μ *locally rank-maximal at radius k* if no assignment differing from μ on at most k agents has a lexicographically larger count vector; radius 2 means that no exchange of two agents' houses improves the count vector (*swap-optimal*), radius 3 adds the cyclic exchanges among three agents. For $n \leq 4$ every swap-optimal assignment is Bordes uncovered (Lean), and this is sharp: an explicit five-agent profile has a swap-optimal assignment that is McKelvey covered (Lean). For $n \leq 7$ every radius-3 locally rank-maximal assignment is Bordes uncovered (SAT). The covering quantifier, by contrast, cannot be localised: at $n = 6$ there is a profile and a $\nu \succ \mu$ with μ rank-maximal for which the Bordes condition holds for every η moving at most four agents and fails for some η moving more.
- **The Gillies variant fails from four agents** (Theorem 4.1, Propositions 3.4 and 4.3). The source's remark is about the McKelvey uncovered set, and its inclusion does not survive replacing McKelvey by Gillies covering. The mechanism is the source's own observation that a popular assignment "automatically Gillies-covers" everything it strictly dominates. What is new is that this fires on rank-maximal assignments: explicit profiles with $4, 5, 6, 7$ agents carrying a rank-maximal assignment μ and a popular λ with $\lambda \succ \mu$ are verified in Lean, and on the same profiles μ is Bordes uncovered for $n = 4, 5, 6$, so the Gillies uncovered set differs there from the Bordes and the McKelvey ones on a rank-maximal assignment; $n = 4$ is the smallest size at which this happens, since at $n = 3$ all three inclusions hold; exactly 9 504 of the 331 776 four-agent profiles (2.865%) have a rank-maximal assignment outside $\mathrm{UC_G}$; and a padding construction, proved in prose, gives such a profile for every $n \geq 4$.

Table 1 collects the status of every cell. What is *not* settled is the general question: whether every rank-maximal assignment is McKelvey (or Bordes) uncovered for all n. Section 6 records why the natural candidate witnesses — assignments built from the cycles of the would-be coverer — can never break a Bordes covering, which is why the finite cases needed a solver rather than an argument.

What is new and what is not. The definitions, the remark, the mechanism behind the Gillies failure and the $n \leq 5$ verification of the McKelvey inclusion are the source's [1]; that the Bordes and Gillies uncovered sets refine the McKelvey one is stated there and is one line from the definitions.

n	rank-max. $\Rightarrow$ $\mathrm{UC_B}$ (so UC)	swap-opt. $\Rightarrow$ $\mathrm{UC_B}$	radius 3 $\Rightarrow$ $\mathrm{UC_B}$	rank-max. $\Rightarrow$ $\mathrm{UC_G}$
3	yes: Lean, 216 profiles	yes: Lean	yes: SAT	yes: Lean
4	yes: Lean, 331 776 profiles	yes: Lean	yes: SAT	**no**: 9 504 profiles, Lean
5	yes: source; C, 9 078 630 profiles; SAT	**no**: Lean witness	yes: SAT	no: Lean witness
6	yes: SAT, 10 certificates	no: SAT witness, re-checked	yes: SAT	no: Lean witness
7	yes: SAT, 14 classes	no: SAT witness, re-checked	yes: SAT	no: Lean witness
≥ 8	open	not addressed	open	no: padding, prose

Table 1. Status by number of agents. "Lean" means verified in Lean 4 from the definitions; "SAT" means a solver refutation, not formalised (Section 5; unsatisfiability certificates were emitted and kept for the ten $n = 6$ instances and for one $n = 7$ instance); "SAT witness, re-checked" means a satisfying assignment returned by the solver and re-verified from the definitions in Python (Computation 5.2). The $n = 5$ entry of the first column is the source's own claim, reproduced here both by an exhaustive C program over all profiles up to relabelling and by the solver.

Rank-maximal matchings and their polynomial-time computation are due to Irving et al. [2], popular matchings to Gärdenfors [4] and Abraham et al. [3]; the covering relations and their names go back to the papers cited above. None of these is consulted for anything beyond attribution, and no claim here rests on them. What we claim is: the cases $n = 6, 7$ of the source's remark (in the Bordes form, by certificate); the local strengthening with its sharp radius threshold; the failure of the Gillies variant, with its minimality, its witnesses, its census and its extension to all $n \geq 4$; the mechanical verification of the $n \leq 4$ cases; and the obstruction of Section 6. As of 2026-09-07 the source has no citing work in OpenAlex. A full-text sweep of a local corpus of 890 739 arXiv e-prints (376 shards; the computer-science and economics listings, the mathematics listings and a machine-learning slice, through early September 2026) finds 154 e-prints using the phrase "rank-maximal" (hyphenated or spaced) and 213 using "uncovered set", and exactly one using both: the source. The arXiv API returns nothing for the conjunction `"uncovered set" AND "rank-maximal"` over titles and abstracts.

Verification. Every theorem and proposition below that carries a formal counterpart in Appendix A is verified in Lean 4 against *Mathlib*, from the definitions of Section 2 transcribed literally; Section 8 records the modules, the use of compiled evaluation, and the axioms. The three items marked "Computation" and the proposition marked "not formalised" are outside the formal development and say so where they occur.

2. Preliminaries

2.1. **Profiles, assignments, ranks.** Throughout, N and H both have n elements, and in explicit examples both are labelled $0, 1, \dots, n-1$; a profile is then displayed as the list of the agents' rankings, agent 0's first, each ranking listing the houses best first, and an assignment μ as the list $(\mu(0), \dots, \mu(n-1))$. This is the representation of the formal development, and the letter notation of the source ($a, b, c, \dots$ for $0, 1, 2, \dots$) is used alongside it.

The *rank* of house p for agent x is $r(\succ_x, p) = 1 + |\{q \in H : q \succ_x p\}|$, so the favourite house has rank 1. The *count vector* of an assignment μ is

$$c(\mu) = (c_1, \dots, c_n), \qquad c_k = \big|\{x \in N : r(\succ_x, \mu(x)) = k\}\big|,$$

and μ is *rank-maximal* if $c(\mu) \geq_{\text{lex}} c(\lambda)$ for every $\lambda \in \mathcal{M}$, where $\geq_{\text{lex}}$ is the lexicographic order on $\mathbb{N}^n$. This is the source's gloss ("maximizes the number of agents who obtain their favorite house, subject to this condition it maximizes the number of agents who obtain their second-ranked house, and so on"); maximising the count vector lexicographically and minimising the sorted rank vector lexicographically are the same condition, and the count-vector form is the one transcribed.

For $k \geq 1$, μ is *locally rank-maximal at radius k* if $c(\mu) \geq_{\text{lex}} c(\lambda)$ for every $\lambda \in \mathcal{M}$ with $|\{x : \mu(x) \neq \lambda(x)\}| \leq k$. An assignment differing from μ on exactly two agents is obtained by exchanging those two agents' houses, and one differing on three agents by a cyclic exchange among the three, so radius 2 says that no exchange of two agents' houses improves the count vector — we say μ is *swap-optimal* — and radius 3 adds the three-cycles. Rank-maximality is the radius-n case and implies every radius.

2.2. **Majority, popularity, covering.** For assignments μ, λ let $N_{\mu,\lambda} = \{x \in N : \mu(x) \succeq_x \lambda(x)\}$ be the agents who weakly prefer μ to λ, where $\succeq_x$ is $\succ_x$ or equality; agents with $\mu(x) = \lambda(x)$ lie in both $N_{\mu,\lambda}$ and $N_{\lambda,\mu}$. Then, as in the source's Section 3,

$$\mu \succsim \lambda \iff |N_{\mu,\lambda}| \geq |N_{\lambda,\mu}|, \qquad \mu \succ \lambda \iff |N_{\mu,\lambda}| > |N_{\lambda,\mu}|,$$

the weak and the strict majority domination. Equivalently $\mu \succ \lambda$ iff the *margin* $|\{x : \mu(x) \succ_x \lambda(x)\}| - |\{x : \lambda(x) \succ_x \mu(x)\}|$ is positive; the formal development uses $|N_{\mu,\lambda}|$ itself. The relation $\succ$ is irreflexive and asymmetric, and it has ties: two agents with the same ranking who exchange houses produce two distinct assignments neither of which dominates the other (Proposition 7.1(v)). An assignment μ is *popular* if $\mu \succsim \lambda$ for all $\lambda \in \mathcal{M}$, i.e. if no λ has $\lambda \succ \mu$.

Following the source's Section 4.2 verbatim:

Definition 2.1. Let P be a profile and $\mu, \lambda \in \mathcal{M}$.

- μ *Bordes covers* λ if $\mu \succ \lambda$ and, for every $\eta \in \mathcal{M}$, $\lambda \succ \eta$ implies $\mu \succ \eta$;
- μ *Gillies covers* λ if $\mu \succ \lambda$ and, for every $\eta \in \mathcal{M}$, $\eta \succ \mu$ implies $\eta \succ \lambda$;
- μ *McKelvey covers* λ if it Bordes covers and Gillies covers λ.

For each covering relation C the *uncovered set* is $\mathrm{UC}_C(P) = \{\mu \in \mathcal{M} : \text{no } \lambda \in \mathcal{M}\ C\text{-covers } \mu\}$; we write $\mathrm{UC}_\mathrm{B}, \mathrm{UC}_\mathrm{G}, \mathrm{UC}_\mathrm{M}$, and $\mathrm{UC} = \mathrm{UC}_\mathrm{M}$ as the source does.

The source quantifies over "every third assignment η"; the two excluded instances $\eta = \mu$, $\eta = \lambda$ are vacuous when $\mu \succ \lambda$, so quantifying over all of $\mathcal{M}$, as the formal development does, is the same relation.

2.3. **Three one-line facts.** The following are used throughout; each is a direct consequence of the definitions and is verified in Lean for every n and every profile.

Lemma 2.2. *Let P be a profile on n agents.*

(a) *If $\mu \in \mathrm{UC}_\mathrm{B}(P)$ then $\mu \in \mathrm{UC}_\mathrm{M}(P)$; likewise if $\mu \in \mathrm{UC}_\mathrm{G}(P)$ then $\mu \in \mathrm{UC}_\mathrm{M}(P)$. (The Bordes and the Gillies uncovered sets refine the McKelvey one.)*

(b) *If λ is popular and $\lambda \succ \mu$, then λ Gillies covers μ.*

(c) *If μ is popular then $\mu \in \mathrm{UC}_\mathrm{B}(P) \cap \mathrm{UC}_\mathrm{G}(P)$, hence $\mu \in \mathrm{UC}(P)$.*

Proof. (a) McKelvey covering is the conjunction of the other two, so anything McKelvey covering μ Bordes covers it. (b) The Gillies condition "$\eta \succ \lambda$ implies $\eta \succ \mu$" is vacuous because nothing strictly dominates a popular λ. (c) Every covering relation requires the coverer to strictly dominate μ, and nothing strictly dominates a popular assignment. $\square$

Part (b) is the source's mechanism ("If μ is such a popular assignment, there exists no η with $\eta \succ \mu$, and hence μ automatically Gillies-covers all λ with $\mu \succ \lambda$"); part (c) is the popular half of the sentence closing the source's remark, and shows that the McKelvey and Bordes statements below are the stronger ones: proving Bordes uncoveredness settles the source's UC by (a).

3. Three and four agents, from the definitions

All statements in this section are verified by complete enumeration inside Lean: the $6^3 = 216$ three-agent profiles and the $24^4 = 331\,776$ four-agent profiles are generated as lists, every profile is proved to occur in the enumeration, and for each profile the $n!$ count vectors are computed once, the rank-maximal (or locally rank-maximal) assignments are read off, and each is tested against all $n!$ candidate coverers with the covering quantifier running over all $n!$ assignments. No symmetry reduction is used, so nothing has to be transported. The proof sketches say which claims rest on this evaluation.

Proposition 3.1 (three agents)**.** *Let P be a profile on three agents and μ a rank-maximal assignment. Then $\mu \in \mathrm{UC}_\mathrm{B}(P)$, $\mu \in \mathrm{UC}_\mathrm{G}(P)$ and $\mu \in \mathrm{UC}_\mathrm{M}(P)$.*

Proof sketch. Exhaustive evaluation over the 216 profiles, once for the Bordes relation and once for the Gillies relation; the McKelvey statement is Lemma 2.2(a). □

This is inside the source's own $n \leq 5$ verification and is not new; it is recorded because it is what makes $n = 4$ the *smallest* size at which the Gillies inclusion fails (Theorem 4.1).

Theorem 3.2 (four agents)**.** *Let P be a profile on four agents and μ a rank-maximal assignment. Then $\mu \in \mathrm{UC_B}(P)$, and hence $\mu \in \mathrm{UC}(P)$.*

Proof sketch. Exhaustive evaluation over all 331 776 profiles for the Bordes relation (370 processor-seconds of compiled evaluation, as recorded when the development was written), then Lemma 2.2(a). The same count was obtained independently by two C programs and a Python program, each written from the definitions alone; all three report 0 profiles with a Bordes- or McKelvey-covered rank-maximal assignment. □

The McKelvey form of Theorem 3.2 is the $n = 4$ case of the source's remark, and what is new here is only that a proof checker verifies it from the definitions rather than a reported experiment. Much less than rank-maximality is needed:

Theorem 3.3 (a local hypothesis suffices for $n \leq 4$, and radius 2 is sharp)**.**
(i) *Let P be a profile on three or on four agents and μ a swap-optimal assignment (locally rank-maximal at radius 2). Then $\mu \in \mathrm{UC_B}(P)$; for four agents, hence $\mu \in \mathrm{UC}(P)$.*
(ii) *In the five-agent profile*

$$P_{\mathrm{S}} = \big[\,[0,1,2,3,4],\ [0,1,2,3,4],\ [0,3,1,2,4],\ [1,0,2,3,4],\ [1,4,0,2,3]\,\big]$$

the assignment $\mu_{\mathrm{S}} = (2,4,0,3,1)$ is swap-optimal, is McKelvey covered (by $\lambda_{\mathrm{S}} = (1,0,3,2,4)$), is in particular Bordes covered, and is not rank-maximal.

Proof sketch. (i) is the same exhaustive evaluation as Theorem 3.2 with the rank-maximality test replaced by the radius-2 test; Theorem 3.2 is its corollary, since rank-maximality implies swap-optimality. (ii) is decided on the one profile: the radius-2 test, the Bordes and Gillies conditions for λ_{S} over all 120 assignments, and the failure of the radius-3 test (the cyclic exchange $(0,4,3,2,1)$ among agents $0,2,3$ raises the count vector from $(2,0,1,1,1)$ to $(2,1,1,0,1)$), which also refutes rank-maximality. The coverer λ_{S} is not popular, so this is a different mechanism from that of Section 4. □

The profile P_{S} is the first, in the enumeration order of the C program over the 9 078 630 five-agent profiles up to relabelling, of 2 680 such profiles; the SAT search of Section 5 finds a different one. Because μ_{S} is not rank-maximal, nothing in (ii) touches the source's remark or Computation 5.1.

Proposition 3.4 (census at four agents)**.** *Exactly 9 504 of the 331 776 four-agent profiles have a rank-maximal assignment outside $\mathrm{UC_G}(P)$, and 0 of them have a rank-maximal assignment outside $\mathrm{UC_B}(P)$.*

Proof sketch. The enumeration of four-agent profiles is proved duplicate-free and of length 331 776, so a count over it is a count of profiles, and the predicate "some rank-maximal assignment of P is Gillies covered" is proved equivalent to its evaluated form. Both counts are then compiled evaluations. The three programs of Theorem 3.2 all return 9 504 and 0. □

4. The Gillies variant fails from four agents

Theorem 4.1 (explicit witnesses, and separation)**.** *For $n = 4,5,6,7$ let P_n be the profile, μ_n and λ_n the assignments*

$$P_4 = \big[\,[0,1,2,3],\ [0,1,2,3],\ [0,3,2,1],\ [3,0,1,2]\,\big], \qquad \mu_4 = (2,1,0,3), \qquad \lambda_4 = (1,0,2,3),$$

and, for $n = 5,6,7$, P_n obtained from P_{n-1} by appending the new house $n-1$ at the end of every old agent's ranking and adding the new agent $n-1$ with ranking $n-1, 0, 1, \ldots, n-2$, with $\mu_n =$

$\mu_{n-1} \cup \{n-1 \mapsto n-1\}$ and $\lambda_n = \lambda_{n-1} \cup \{n-1 \mapsto n-1\}$; explicitly

$$P_5 = \big[\,[0,1,2,3,4],[0,1,2,3,4],[0,3,2,1,4],[3,0,1,2,4],[4,0,1,2,3]\,\big],$$
$$P_6 = \big[\,[0,1,2,3,4,5],[0,1,2,3,4,5],[0,3,2,1,4,5],[3,0,1,2,4,5],[4,0,1,2,3,5],[5,0,1,2,3,4]\,\big],$$
$$P_7 = \big[\,[0,1,2,3,4,5,6],[0,1,2,3,4,5,6],[0,3,2,1,4,5,6],[3,0,1,2,4,5,6],[4,0,1,2,3,5,6],$$
$$[5,0,1,2,3,4,6],[6,0,1,2,3,4,5]\,\big],$$

$\mu_5 = (2,1,0,3,4)$, $\mu_6 = (2,1,0,3,4,5)$, $\mu_7 = (2,1,0,3,4,5,6)$ and $\lambda_5 = (1,0,2,3,4)$, $\lambda_6 = (1,0,2,3,4,5)$, $\lambda_7 = (1,0,2,3,4,5,6)$. Then, for each $n \in \{4,5,6,7\}$: μ_n is rank-maximal in P_n, λ_n is popular in P_n, $\lambda_n \succ \mu_n$, and consequently λ_n Gillies covers μ_n, i.e. $\mu_n \notin \mathrm{UC_G}(P_n)$. Moreover $\mu_n \in \mathrm{UC_B}(P_n)$ for $n = 4,5,6$, and $\mu_4 \in \mathrm{UC}(P_4)$. In particular, at $n = 4,5,6$ the Gillies uncovered set differs on a rank-maximal assignment from the Bordes and from the McKelvey uncovered set; and, by Proposition 3.1, $n = 4$ is the smallest number of agents for which some rank-maximal assignment lies outside the Gillies uncovered set.

Proof sketch. For each n, rank-maximality of μ_n (its count vector against all $n!$ count vectors) and popularity of λ_n (all $n!$ margins against λ_n) are compiled evaluations on the one profile, and $\lambda_n \succ \mu_n$ is decided by the kernel; the Gillies covering is then Lemma 2.2(b), with no further computation. For $n = 4$, $\mu_4 \in \mathrm{UC_B}(P_4)$ and $\mu_4 \in \mathrm{UC}(P_4)$ are instances of Theorem 3.2; for $n = 5,6$ the Bordes uncoveredness of μ_n is a compiled evaluation over all $n!$ candidate coverers. All of these facts were re-derived for this paper, from the definitions alone, in Python (including $n = 8$ and $n = 9$ for the padded profiles, where μ_n is again rank-maximal and λ_n again popular with $\lambda_n \succ \mu_n$). ☐

In the source's letter notation P_4 is: agents 0 and 1 rank $a \succ b \succ c \succ d$, agent 2 ranks $a \succ d \succ c \succ b$, agent 3 ranks $d \succ a \succ b \succ c$; $\mu_4 = (c,b,a,d)$ and $\lambda_4 = (b,a,c,d)$ have the same count vector $(2,1,1,0)$, and $\lambda_4 \succ \mu_4$ by the margin $2-1 = 1$ (agents $0,1$ prefer λ_4, agent 2 prefers μ_4, agent 3 is indifferent). P_4 is the lexicographically first four-agent profile, in the enumeration order of the C program, carrying a rank-maximal assignment that is Gillies covered.

Remark 4.2 (the coverer is itself rank-maximal). Outside the formal development. In P_4 the four rank-maximal assignments are $(0,1,2,3)$, $(1,0,2,3)$, $(1,2,0,3)$ and $(2,1,0,3)$, all with count vector $(2,1,1,0)$; so λ_4 is itself rank-maximal, and popular, and it Gillies covers the rank-maximal μ_4. The Gillies failure is thus a rank-maximal assignment covered by another rank-maximal assignment, not by an assignment the rank-maximal rule would have rejected. λ_4 itself is Bordes and Gillies uncovered, as Lemma 2.2(c) requires.

Proposition 4.3 (every $n \geq 4$; not formalised)**.** *For every $n \geq 4$ there is a profile on n agents with a rank-maximal assignment that is Gillies covered by a popular assignment; in particular the Gillies uncovered set does not contain every rank-maximal assignment for any $n \geq 4$.*

The proof is an ordinary argument and is not formalised; the instances $n = 4,5,6,7$ of the construction are the ones verified in Theorem 4.1. It rests on the following padding lemma.

Lemma 4.4 (padding)**.** *Let P be a profile on agents and houses $\{0,\dots,n-1\}$. Define the profile P' on $\{0,\dots,n\}$ by giving every old agent $i < n$ its old ranking with the new house n appended (so n is the unanimously worst house among the old agents), and giving the new agent n the ranking $n \succ 0 \succ 1 \succ \cdots \succ n-1$. For an assignment a of P write $a' = a \cup \{n \mapsto n\}$. Then:*

(1) *the margin of a' over b' in P' equals the margin of a over b in P, for all assignments a, b of P; in particular $a' \succ b'$ in P' iff $a \succ b$ in P;*
(2) *if a is rank-maximal in P then a' is rank-maximal in P';*
(3) *if a is popular in P then a' is popular in P'.*

Proof. Ranks of old houses for old agents are the same in P and P', since the new house is appended at the end.

(1) Agent n receives n under both a' and b' and contributes nothing; every old agent compares the same two old houses as before.

(2) Let α be an assignment of P'. If $\alpha(n) = n$, then $\alpha = \alpha_0'$ for the assignment α_0 of P obtained by restriction, $c(\alpha) = c(\alpha_0) + e_1$ and $c(a') = c(a) + e_1$ (agent n is at rank 1, the old agents keep their ranks, and the new last coordinate is 0 for both), so $c(a') \geq_{\text{lex}} c(\alpha)$ because $c(a) \geq_{\text{lex}} c(\alpha_0)$. If $\alpha(n) = h \neq n$, then agent n is not at rank 1 (its rank-1 house is n), and exactly one old agent j has $\alpha(j) = n$, its worst house, so j is not at rank 1 either. Let β be the assignment of P with $\beta(j) = h$ and $\beta(i) = \alpha(i)$ for old $i \neq j$; this is a bijection onto the old houses. Every old agent $i \neq j$ at rank 1 under α is at rank 1 under β, so $c_1(\alpha) \leq c_1(\beta) \leq c_1(a) < c_1(a) + 1 = c_1(a')$, the middle inequality because a is rank-maximal in P. Hence $c(a') >_{\text{lex}} c(\alpha)$.

(3) Let α be an assignment of P'; we show that the margin of α over a' in P' is at most 0. If $\alpha(n) = n$ this is (1) and the popularity of a. Otherwise let $h = \alpha(n) \neq n$, let j be the old agent with $\alpha(j) = n$, and let β be as in (2). For an old agent $i \neq j$ let $\sigma_i \in \{-1, 0, 1\}$ be the sign of its preference for $\alpha(i)$ over $a(i)$; these are the same in P and P'. Popularity of a in P applied to β gives $\sum_{i \neq j} \sigma_i + \sigma_j^\beta \leq 0$, where $\sigma_j^\beta \in \{-1, 0, 1\}$ is agent j's sign for h against $a(j)$; hence $\sum_{i \neq j} \sigma_i \leq 1$. In P', agent j compares its worst house n with $a(j) \neq n$ and contributes -1, and agent n compares h with its favourite house n and contributes -1. So the margin of α over a' is at most $1 - 1 - 1 = -1 \leq 0$. □

Proof of Proposition 4.3. Start from (P_4, μ_4, λ_4) of Theorem 4.1 and apply Lemma 4.4 $n - 4$ times. At every stage μ is rank-maximal by (2), λ is popular by (3), and $\lambda \succ \mu$ by (1); so λ Gillies covers μ by Lemma 2.2(b). The profiles P_5, P_6, P_7 of Theorem 4.1 are the first three iterates. □

Remark 4.5. Outside the formal development. Padding does *not* carry the sharpness witness of Theorem 3.3(ii) upwards: applying Lemma 4.4 to $(P_{\text{S}}, \mu_{\text{S}}, \lambda_{\text{S}})$ once and twice, the padded μ_{S} is still swap-optimal (and still not radius-3 locally rank-maximal), but — checked from the definitions — the padded λ_{S} no longer Bordes covers it at $n = 6$ or at $n = 7$; Lemma 4.4 asserts nothing about covering, whose quantifier ranges over the unpadded assignments of the larger profile as well. The $n = 6$ and $n = 7$ entries of the second column of Table 1 therefore come from a direct solver search, Computation 5.2(ii), not from padding, and nothing is claimed for $n \geq 8$.

Computation 4.6 (Gillies failures at five agents, and the popular-coverer question)**.** Not formalised. A C program written from the definitions enumerates the five-agent profiles up to relabelling of agents and houses — agent 0's ranking is fixed to the identity and the remaining four rankings are taken in non-decreasing order, giving $\binom{123}{4} = 9\,078\,630$ profiles, the number the source prints in its Figure 1 for its own $n = 5$ computations — and reports, in 54 seconds: 0 profiles with a Bordes- or McKelvey-covered rank-maximal assignment (the source's own $n = 5$ claim, reproduced), and 544 720 with a Gillies-covered one. Counting the triples (profile, rank-maximal μ, coverer λ Gillies covering μ): at $n = 4$ all 12 672 of them have a popular coverer, so for four agents Lemma 2.2(b) characterises the Gillies failures; at $n = 5$, 698 800 have a popular coverer and 32 560 do not (out of 731 360), so the characterisation "the coverer is popular" fails from five agents. The $n = 4$ counts were reproduced by the Python program of Theorem 3.2; the $n = 5$ counts come from C only, and every count in this computation was re-derived for this paper by a second, independently written C program.

5. Beyond five agents: certificates

Nothing in this section is formalised. The statements are established by unsatisfiability certificates of the SAT solver CaDiCaL [16] on formulas whose *soundness* — every genuine counterexample yields a satisfying assignment — is argued in prose below; the size of the formalisation gap is priced at the end.

5.1. **The symmetry reduction.** Relabelling the agents by $\sigma \in \text{Sym}(N)$ and the houses by $\tau \in \text{Sym}(H)$ sends the profile P to the profile $P^{\sigma,\tau}$ in which agent $\sigma(x)$ ranks $\tau(p)$ above $\tau(q)$ iff x ranks p above q, and sends an assignment μ to $\tau \circ \mu \circ \sigma^{-1}$. Every rank is preserved, hence so are count vectors, the number of agents on which two assignments differ, all majority margins, rank-maximality, local rank-maximality at every radius, popularity, and the three covering relations. Given a counterexample — a profile P, a (locally) rank-maximal μ and a ν that Bordes covers it — take $\sigma = \text{id}$ and $\tau = \mu^{-1}$: now μ is the identity assignment. The residual freedom is the diagonal $\pi \mapsto (\pi, \pi)$, which fixes the identity and sends ν to $\pi\nu\pi^{-1}$; and $\nu \neq \mathrm{id}$ because $\nu \succ \mathrm{id}$. So it suffices to refute, for one representative ν of each non-identity conjugacy class of $\mathrm{Sym}(n)$, the existence of a profile in which id is (locally) rank-maximal and ν Bordes covers it: 2 classes at $n = 3$, 4 at $n = 4$, 6 at $n = 5$, 10 at $n = 6$ and 14 at $n = 7$.

5.2. **The encoding.** For fixed n, ν, a radius $k \leq n$ and a covering relation, the formula has Boolean variables $p_{i,a,b}$ $(a < b)$ for "agent i prefers house a to house b", with the transitivity clause $\neg p_{i,a,b} \vee \neg p_{i,b,c} \vee p_{i,a,c}$ for every ordered triple of distinct houses (so each agent's preference is a strict linear order); $\ell_{i,h,r}$ $(1 \leq r < n)$ reified in both directions as "at most $r-1$ of the $n-1$ houses $q \neq h$ beat h for agent i", i.e. $r(\succ_i, h) \leq r$, by direct subset clauses; $\gamma_{e,r,j}$ reified in both directions as "at least j agents i have $\ell_{i,e(i),r}$", i.e. $C_r(e) \geq j$ where $C_r(e) = c_1(e) + \cdots + c_r(e)$; and $m_{a,b}$ reified in both directions as "$a \succ b$", i.e. among the agents on which a and b differ a strict majority prefer $a(i)$ to $b(i)$, by subset clauses on those agents' preference literals. Local rank-maximality of id at radius k is asserted as $(C_1, \ldots, C_{n-1})(\mathrm{id}) \geq_{\mathrm{lex}} (C_1, \ldots, C_{n-1})(e)$ for every $e \neq \mathrm{id}$ moving at most k agents, through a chain of variables $g_r \Rightarrow \big(C_r(\mathrm{id}) > C_r(e)\big) \vee \big(C_r(\mathrm{id}) = C_r(e) \wedge g_{r+1}\big)$ with g_n true and g_1 asserted; lexicographic order of the cumulative vectors is lexicographic order of the count vectors, and $C_n \equiv n$ is omitted. Covering is $m_{\nu,\mathrm{id}}$ together with, for every $e \notin \{\mathrm{id}, \nu\}$, $\neg m_{\mathrm{id},e} \vee m_{\nu,e}$ (Bordes) or $\neg m_{e,\nu} \vee m_{e,\mathrm{id}}$ (Gillies). A genuine counterexample, after the reduction, satisfies every clause when the variables are set to their intended values (the chain variables to the truth of the conditions they abbreviate); hence unsatisfiability for every class representative proves the statement. Restricting the rank-maximality quantifier to assignments moving at most k agents *removes* clauses, so an unsatisfiable formula at radius k proves the statement with the weaker local hypothesis; restricting the covering quantifier likewise removes clauses and proves a statement with a weaker conclusion, which is how the localisation of the covering quantifier is tested.

5.3. **Validation.** At $n = 3$ and $n = 4$ the formulas reproduce the exhaustive results of Section 3 in both directions: unsatisfiable for Bordes covering on every class; for Gillies covering unsatisfiable on every class at $n = 3$, and at $n = 4$ satisfiable on exactly the class of a 3-cycle, with the returned profile $[[0,1,3,2],[1,0,2,3],[0,3,2,1],[1,3,0,2]]$, $\nu = (0,2,3,1)$, in which — re-checked from the definitions — id is rank-maximal, ν is popular, ν Gillies covers id and does not Bordes cover it. At $n = 5$ the Bordes formulas are unsatisfiable on all six classes, matching the 0 of Computation 4.6.

Computation 5.1 ($n = 6$ and $n = 7$)**.** For $n = 6$ and for $n = 7$, every rank-maximal assignment of every profile is Bordes uncovered, hence McKelvey uncovered. This extends the source's remark, in the Bordes form, by two sizes. Evidence: with the rank-maximality quantifier restricted to radius 3 (which proves more; see Computation 5.2), all 10 class formulas at $n = 6$ (6 138 variables, 79 052–79 084 clauses each) and all 14 at $n = 7$ (20 389 variables, 604 021–604 088 clauses each) are unsatisfiable. Both sweeps were regenerated and refuted again for this paper, one instance at a time on one core of a shared machine: the ten $n = 6$ instances in 0.3–0.9 seconds each (6.1 seconds in all) and the fourteen $n = 7$ instances in 3.6–33.6 seconds each (334 seconds in all), with clause counts identical to those recorded when the development was written (where the same sweeps took 7.0 and 415 processor-seconds). Without the radius restriction the $n = 6$ sweep is 59 258 variables and 591 thousand clauses per class and 72 seconds in all, again all unsatisfiable.

Computation 5.2 (the local strengthening and its threshold)**.** (i) For $3 \leq n \leq 7$, every assignment that is locally rank-maximal at radius 3 — no exchange of two agents' houses and no cyclic exchange among three agents improves its count vector — is Bordes uncovered, hence McKelvey uncovered. Evidence: the formulas of Computation 5.1 at radius 3 are exactly these, and all are unsatisfiable.

(ii) Radius 2 does not suffice beyond $n = 4$: at $n = 5$ the radius-2 Bordes formula is satisfiable on the class of the 5-cycle (and unsatisfiable on the other five), with the returned profile $W = [[2,1,0,3,4],[2,1,0,4,3],[2,3,0,1,4],[4,1,0,3,2],[4,0,1,3,2]]$ and $\nu = (1,2,3,4,0)$; re-checked from the definitions, id is swap-optimal in W, is not locally rank-maximal at radius 3, and is McKelvey (in particular Bordes) covered by ν. Radius 2 is unsatisfiable at $n = 3$ and $n = 4$, matching Theorem 3.3(i). The kernel-verified sharpness witness is Theorem 3.3(ii), a different profile. Searched for this paper: at $n = 6$ the radius-2 Bordes formula is satisfiable on exactly the classes of the 5-cycle and the 6-cycle (unsatisfiable on the other eight), returning the profiles

$$[[0, 5, 4, 3, 2, 1], [2, 4, 3, 1, 0, 5], [2, 3, 4, 0, 1, 5], [5, 4, 3, 0, 1, 2], [5, 4, 0, 1, 2, 3], [5, 1, 0, 2, 3, 4]],$$
$$\nu = (0, 2, 3, 4, 5, 1);$$
$$[[3, 5, 1, 2, 0, 4], [3, 2, 1, 5, 0, 4], [3, 2, 1, 0, 5, 4], [3, 4, 1, 5, 0, 2], [5, 3, 1, 2, 0, 4], [5, 0, 3, 1, 2, 4]],$$
$$\nu = (1, 2, 3, 4, 5, 0);$$

in both, re-checked from the definitions, id is swap-optimal, not locally rank-maximal at radius 3, not rank-maximal, and McKelvey covered by ν, which here happens to be popular. At $n = 7$ the radius-2 formula (12 829 variables, 483 831–483 898 clauses) is satisfiable on exactly the classes of a 5-cycle with two fixed points and of a 6-cycle with one fixed point, and unsatisfiable on the other twelve (201 seconds for the fourteen); the first returns

$$[[0, 1, 6, 2, 5, 4, 3], [1, 3, 0, 4, 2, 6, 5], [4, 6, 3, 1, 2, 0, 5], [4, 1, 3, 0, 2, 6, 5],$$
$$[4, 5, 1, 6, 2, 3, 0], [6, 1, 3, 4, 0, 2, 5], [6, 2, 4, 3, 0, 1, 5]], \qquad \nu = (0, 1, 3, 4, 5, 6, 2),$$

and the second

$$[[0, 5, 1, 4, 3, 6, 2], [1, 2, 5, 0, 6, 3, 4], [5, 1, 3, 2, 4, 0, 6], [1, 4, 3, 5, 0, 6, 2],$$
$$[5, 4, 3, 1, 2, 0, 6], [5, 6, 1, 2, 0, 4, 3], [1, 4, 3, 2, 6, 0, 5]], \qquad \nu = (0, 2, 3, 4, 5, 6, 1),$$

both re-checked in the same way (swap-optimal, not radius 3, not rank-maximal, McKelvey covered by the popular ν).

(iii) The covering quantifier cannot be localised. At $n = 6$, restricting the Bordes condition to assignments η moving at most four agents makes the formula satisfiable already on the class of the 6-cycle: in the profile where all six agents rank $5 \succ 4 \succ 3 \succ 2 \succ 1 \succ 0$, with $\nu = (1, 2, 3, 4, 5, 0)$, the identity is rank-maximal (all count vectors are $(1, 1, 1, 1, 1, 1)$), $\nu \succ \mathrm{id}$ with margin 4, the Bordes condition holds for all η moving at most four agents, and it fails for 117 of the η moving five or six — all re-checked from the definitions.

The localisation in (i) is also what makes $n = 7$ affordable: without it the $n = 7$ formula for the 7-cycle class has 554 773 variables and 9 099 737 clauses (223 MB of CNF; generated for this paper, not solved), against 604 021 clauses.

5.4. **Certificates, and the gap to formalisation.** CaDiCaL emits an LRAT proof for each refutation. All eleven certificates quoted here were re-emitted and re-measured for this paper, in text format. For the reduced encoding at $n = 6$ the ten certificates are 514 533–2 137 170 bytes each (16 308 471 bytes in all); the class of the 5-cycle is 79 067 clauses, 1 492 859 bytes of CNF and a 2 137 170-byte certificate produced in 0.8 seconds. At $n = 7$ the certificate for the 7-cycle class is 33 258 105 bytes, produced in 19 seconds from 13.5 MB of CNF, so all fourteen are of the order of 0.4 GB. Replaying such a certificate through the verified checker of the Lean toolchain (`Std.Tactic.BVDecide.Reflect.verifyCert`) is cheap: on the smallest and the largest of the ten $n = 6$ certificates it returns `true` in 235 ms and 283 ms, interpreted, after about 0.7 seconds of parsing the DIMACS file inside Lean. Checking is therefore not the obstacle. What a kernel-checked version of Computations 5.1 and 5.2 needs is the clause list *rebuilt inside the proof assistant* from the definitions of Section 2, with soundness proofs for the five gadgets of the encoding (transitivity, the rank-threshold subset clauses, the cumulative-count subset clauses, the lexicographic chain, the majority subset clauses) and for the relabelling reduction; a comparable development of ours elsewhere, covering two of the five gadgets (transitivity and the majority subset clauses) and with no relabelling lemma, runs to about 450 lines, so the estimate is 1 500–2 500 lines of Lean. This is why those two computations are reported as computations and not as theorems. Larger n is out of reach of this route as it stands: at $n = 8$ the covering part alone is $\Theta(n!)$ majority reifications, about six million clauses per class before the rank part, and the measured growth from $n = 6$ to $n = 7$ (eightfold in clauses, fortyfold in time) prices the 21 classes at 7–20 processor-hours; exhaustive enumeration is hopeless much earlier, the six-agent profiles up to relabelling numbering $\binom{724}{5} \approx 1.6 \cdot 10^{12}$.

6. Why there is no cheap proof

To show that no ν Bordes covers a rank-maximal μ one must produce, for every $\nu \succ \mu$, an η with $\mu \succ \eta$ and $\nu \not\succ \eta$. Take $\mu = \mathrm{id}$ and let $s_i \in \{-1, 0, 1\}$ be the sign of agent i's preference for $\nu(i)$ over i, so that $\sum_i s_i > 0$. The natural candidates are the *partial* versions of ν: for a union A of cycles of ν let η_A agree with ν on A and with the identity off A — these are assignments. Then

$$\mathrm{margin}(\mathrm{id}, \eta_A) = -\sum_{i \in A} s_i, \qquad \mathrm{margin}(\nu, \eta_A) = \sum_{i \notin A} s_i,$$

so $\mathrm{id} \succ \eta_A$ forces $\sum_A s_i < 0$, hence $\sum_{i \notin A} s_i = \sum_i s_i - \sum_A s_i > 0$, i.e. $\nu \succ \eta_A$: *no union of cycles of ν is ever a witness*, for any n and any profile, and this uses nothing about rank-maximality. Together with Computation 5.2(iii) — a witness may have to move five of six agents — this says that the witness, when it exists, is far from both id and ν, and that a proof of the general inclusion will need an idea rather than a local exchange argument. Whether the inclusion holds for all n is open.

7. Controls

The theorems of Section 3 and the computations of Section 5 say "nothing covers a rank-maximal assignment"; they would be empty if nothing ever covered anything, or trivial if rank-maximal assignments were secretly popular. The following statements are verified in Lean and rule this out.

Proposition 7.1 (controls)**.** (i) *(The source's Example 5, reproduced.) In the three-agent profile P_{E} with agent 0: $a \succ c \succ b$, agent 1: $a \succ b \succ c$, agent 2: $b \succ a \succ c$ (as lists, $[[0,2,1],[0,1,2],[1,0,2]]$), the assignment $\mu_{\mathrm{E}} = (c, a, b) = (2, 0, 1)$ McKelvey covers $\lambda_{\mathrm{E}} = (a, b, c) = (0, 1, 2)$; so $\lambda_{\mathrm{E}} \notin \mathrm{UC}(P_{\mathrm{E}})$, and the covering relation is non-empty already at the smallest size considered.*

(ii) *λ_{E} is Pareto-optimal. So "every Pareto-optimal assignment is uncovered" is false — the source's point in its Example 5 — and the hypothesis of Proposition 3.1 bites: λ_{E} is not rank-maximal (count vector $(1, 1, 1)$ against $(2, 1, 0)$ for μ_{E}, which is rank-maximal and, by Proposition 3.1, uncovered).*

(iii) *μ_4 of Theorem 4.1 is rank-maximal and not popular. So "every rank-maximal assignment is popular" is false, and the inclusions are not instances of Lemma 2.2(c).*

(iv) *A popular assignment is Bordes uncovered and Gillies uncovered (Lemma 2.2(c)).*

(v) *The majority relation has ties: for two agents with the ranking $0 \succ 1$, the assignments $(0, 1)$ and $(1, 0)$ are distinct and neither strictly majority dominates the other.*

Proof sketch. (i) is two compiled evaluations (the Bordes and the Gillies condition over the six assignments); (ii) is a case split over the six assignments decided by the kernel, and a compiled evaluation for the count vectors; (iii) is the rank-maximality evaluation of Theorem 4.1 together with $\lambda_4 \succ \mu_4$; (iv) is Lemma 2.2(c); (v) is decided by the kernel. □

8. Verification

Lemma 2.2, Proposition 3.1, Theorems 3.2, 3.3 and 4.1, and Propositions 3.4 and 7.1 rest on declarations verified in Lean 4 against *Mathlib* [17, 18] (Lean 4.32.0, Mathlib at its `v4.32.0` tag), whose statements are reproduced verbatim in Appendix A. The development is eight modules, 1 001 lines, 52 definitions and 98 theorems. The definitions of Section 2 are transcribed on plain lists of natural numbers (a ranking is a list of houses, best first; a profile a list of rankings; an assignment a list of houses; all permutations of $[0, \ldots, n-1]$), with the source's sentences quoted beside them, and the enumerations of all assignments and of all profiles are proved complete (`mem_assigns`, `mem_profiles`) and, for the census, duplicate-free (`profiles4_nodup`, `profiles4_length`). The exhaustive statements are proved by compiled evaluation (`native_decide`) of a Boolean sweep over the enumeration, bridged to the quantified statement by proved equivalences (`good_of_goodB`, `goodL_of_goodLB`, `gilliesBadB_iff`, and the `_iff` lemmas for each Boolean transcription); the general lemmas (`gillies_of_popular`, `mckelvey_uncovered_of_bordes`, `bordes_uncovered_of_popular`, `gillies_uncovered_of_popular`, `localRankMaximal_` `of_rankMaximal`) are proved without any evaluation. Accordingly the axiom print (`#print axioms`) of every statement in Appendix A, re-run read-only for this paper, is `propext`, `Classical.choice`, `Quot.sound` (for some only `propext`) plus, for the evaluated statements, exactly the named `native_decide` axioms of the sweeps or witnesses they rest on (e.g. `sweep4B` for Theorem 3.2, `pop4` for the Gillies failure at $n = 4$, and `pop4`, `rmax4`, `sweep4B` together for the separation of Theorem 4.1); of the 80 declarations printed, 45 carry no `native_decide` axiom at all. There is no `sorry` and no axiom declared by hand. Compiled evaluation trusts the Lean compiler in addition to the kernel; the $n = 4$ counts were reproduced outside Lean by three independent programs, two in C (0.4 and 2 seconds) and one in Python (42 seconds). As recorded when the development was written, the $n = 4$ Bordes sweep takes 370 processor-seconds, the radius-2 sweeps about 500, and the census about 770; every other module checks in seconds. Outside the formal development, and labelled as such where they occur, are Remark 4.2, Proposition 4.3 with Lemma 4.4 (prose), Remark 4.5, Computations 4.6, 5.1 and 5.2 (C and SAT), Section 6 (prose), and the bibliographic and corpus searches. The development is currently in a private repository: repository reference to be added at submission.

Appendix A. The formal statements

The Lean statements corresponding to the results above, verbatim (statements only; proofs, docstrings and attributes omitted); the namespace `LeanProblemSpec.RankmaxUncovered` is omitted. `IsPerm n l` says that the list `l` is a permutation of `[0,…,n-1]`, `IsProf n P` that `P` is a list of `n` such permutations (a profile), and `UncoveredFor n C P a` that no assignment `C`-covers `a`.

Definitions.

```
def IsPerm (n : Nat) (l : List Nat) : Prop := l.Perm (List.range n)
def IsProf (n : Nat) (P : List (List Nat)) : Prop := P.length = n ∧ ∀ r ∈ P, IsPerm n r
def rank (r : List Nat) (h : Nat) : Nat := 1 + r.idxOf h
def supp : List (List Nat) → List Nat → List Nat → Nat
  | r :: P, x :: a, y :: b => (if rank r x ≤ rank r y then 1 else 0) + supp P a b
  | _, _, _ => 0
def mdom (P : List (List Nat)) (a b : List Nat) : Bool := decide (supp P b a < supp P a b)
def rcount : List (List Nat) → List Nat → Nat → Nat
  | r :: P, x :: a, k => (if rank r x = k then 1 else 0) + rcount P a k
  | _, _, _ => 0
def rvec (P : List (List Nat)) (a : List Nat) : List Nat :=
  (List.range P.length).map (fun k => rcount P a (k + 1))
def lexGE : List Nat → List Nat → Bool
  | x :: xs, y :: ys => if x = y then lexGE xs ys else decide (y < x)
  | _, _ => true
def RankMaximal (n : Nat) (P : List (List Nat)) (a : List Nat) : Prop :=
  ∀ b, IsPerm n b → lexGE (rvec P a) (rvec P b) = true
def BordesCovers (n : Nat) (P : List (List Nat)) (m l : List Nat) : Prop :=
  mdom P m l = true ∧ ∀ e, IsPerm n e → mdom P l e = true → mdom P m e = true
def GilliesCovers (n : Nat) (P : List (List Nat)) (m l : List Nat) : Prop :=
  mdom P m l = true ∧ ∀ e, IsPerm n e → mdom P e m = true → mdom P e l = true
def McKelveyCovers (n : Nat) (P : List (List Nat)) (m l : List Nat) : Prop :=
  BordesCovers n P m l ∧ GilliesCovers n P m l
def UncoveredFor (n : Nat) (C : Nat → List (List Nat) → List Nat → List Nat → Prop)
    (P : List (List Nat)) (a : List Nat) : Prop :=
  ∀ l, IsPerm n l → ¬ C n P l a
def Popular (n : Nat) (P : List (List Nat)) (m : List Nat) : Prop :=
  ∀ e, IsPerm n e → mdom P e m = false
def diffCount (a b : List Nat) : Nat := (a.zip b).countP (fun p => !(p.1 == p.2))
def LocalRankMaximal (n k : Nat) (P : List (List Nat)) (a : List Nat) : Prop :=
```

```
  ∀ b, IsPerm n b → diffCount a b ≤ k → lexGE (rvec P a) (rvec P b) = true
```

Lemma 2.2.

```
theorem mckelvey_uncovered_of_bordes {n P a}
    (h : UncoveredFor n BordesCovers P a) : UncoveredFor n McKelveyCovers P a
theorem mckelvey_uncovered_of_gillies {n P a}
    (h : UncoveredFor n GilliesCovers P a) : UncoveredFor n McKelveyCovers P a
theorem gillies_of_popular {n P m l} (hp : Popular n P m) (h : mdom P m l = true) :
    GilliesCovers n P m l
theorem bordes_uncovered_of_popular {n P m} (h : Popular n P m) :
    UncoveredFor n BordesCovers P m
theorem gillies_uncovered_of_popular {n P m} (h : Popular n P m) :
    UncoveredFor n GilliesCovers P m
theorem localRankMaximal_of_rankMaximal {n k P a} (h : RankMaximal n P a) :
    LocalRankMaximal n k P a
```

Proposition 3.1 and Theorem 3.2.

```
theorem three_bordes (P : List (List Nat)) (hP : IsProf 3 P)
    {a : List Nat} (ha : IsPerm 3 a) (hrm : RankMaximal 3 P a) :
    UncoveredFor 3 BordesCovers P a
theorem three_gillies (P : List (List Nat)) (hP : IsProf 3 P)
    {a : List Nat} (ha : IsPerm 3 a) (hrm : RankMaximal 3 P a) :
    UncoveredFor 3 GilliesCovers P a
theorem three_mckelvey (P : List (List Nat)) (hP : IsProf 3 P)
    {a : List Nat} (ha : IsPerm 3 a) (hrm : RankMaximal 3 P a) :
    UncoveredFor 3 McKelveyCovers P a
theorem four_bordes (P : List (List Nat)) (hP : IsProf 4 P)
    {a : List Nat} (ha : IsPerm 4 a) (hrm : RankMaximal 4 P a) :
    UncoveredFor 4 BordesCovers P a
theorem four_mckelvey (P : List (List Nat)) (hP : IsProf 4 P)
    {a : List Nat} (ha : IsPerm 4 a) (hrm : RankMaximal 4 P a) :
    UncoveredFor 4 McKelveyCovers P a
```

Theorem 3.3.

```
theorem three_bordes_local (P : List (List Nat)) (hP : IsProf 3 P)
    {a : List Nat} (ha : IsPerm 3 a) (hrm : LocalRankMaximal 3 2 P a) :
    UncoveredFor 3 BordesCovers P a
theorem four_bordes_local (P : List (List Nat)) (hP : IsProf 4 P)
    {a : List Nat} (ha : IsPerm 4 a) (hrm : LocalRankMaximal 4 2 P a) :
    UncoveredFor 4 BordesCovers P a
theorem four_mckelvey_local (P : List (List Nat)) (hP : IsProf 4 P)
    {a : List Nat} (ha : IsPerm 4 a) (hrm : LocalRankMaximal 4 2 P a) :
    UncoveredFor 4 McKelveyCovers P a

def PS5 : List (List Nat) :=
  [[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 3, 1, 2, 4], [1, 0, 2, 3, 4], [1, 4, 0, 2, 3]]
def muS5 : List Nat := [2, 4, 0, 3, 1]
def lamS5 : List Nat := [1, 0, 3, 2, 4]
theorem local_radius_two_sharp :
    LocalRankMaximal 5 2 PS5 muS5 ∧ ¬ UncoveredFor 5 McKelveyCovers PS5 muS5 ∧
      ¬ UncoveredFor 5 BordesCovers PS5 muS5 ∧ ¬ RankMaximal 5 PS5 muS5
theorem coversS5 : McKelveyCovers 5 PS5 lamS5 muS5
theorem notLocal3S5 : ¬ LocalRankMaximal 5 3 PS5 muS5
```

```
theorem lamS5_not_popular : ¬ Popular 5 PS5 lamS5
```

Proposition 3.4.

```
theorem profiles4_nodup : (tuples 4 (assigns 4)).Nodup
theorem profiles4_length : (tuples 4 (assigns 4)).length = 331776
theorem gilliesBadB_iff {n : Nat} {P : List (List Nat)} :
    gilliesBadB (assigns n) P = true ↔
      ∃ a, IsPerm n a ∧ RankMaximal n P a ∧ ¬ UncoveredFor n GilliesCovers P a
theorem census_four :
    (tuples 4 (assigns 4)).countP (gilliesBadB (assigns 4)) = 9504
theorem census_four_bordes :
    (tuples 4 (assigns 4)).countP (fun P => !(goodB (assigns 4) P)) = 0
```

Theorem 4.1.

```
def P4 : List (List Nat) := [[0, 1, 2, 3], [0, 1, 2, 3], [0, 3, 2, 1], [3, 0, 1, 2]]
def mu4 : List Nat := [2, 1, 0, 3]
def lam4 : List Nat := [1, 0, 2, 3]
theorem rmax4 : RankMaximal 4 P4 mu4
theorem pop4 : Popular 4 P4 lam4
theorem dom4 : mdom P4 lam4 mu4 = true
theorem gillies_fails_4 : ¬ UncoveredFor 4 GilliesCovers P4 mu4
theorem separation_four :
    UncoveredFor 4 McKelveyCovers P4 mu4 ∧ UncoveredFor 4 BordesCovers P4 mu4 ∧
      ¬ UncoveredFor 4 GilliesCovers P4 mu4

def P5 : List (List Nat) :=
  [[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 3, 2, 1, 4], [3, 0, 1, 2, 4], [4, 0, 1, 2, 3]]
def mu5 : List Nat := [2, 1, 0, 3, 4]
def lam5 : List Nat := [1, 0, 2, 3, 4]
theorem rmax5 : RankMaximal 5 P5 mu5
theorem pop5 : Popular 5 P5 lam5
theorem gillies_fails_5 : ¬ UncoveredFor 5 GilliesCovers P5 mu5
theorem separation_five :
    UncoveredFor 5 BordesCovers P5 mu5 ∧ ¬ UncoveredFor 5 GilliesCovers P5 mu5

def P6 : List (List Nat) := [[0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5], [0, 3, 2, 1, 4, 5],
  [3, 0, 1, 2, 4, 5], [4, 0, 1, 2, 3, 5], [5, 0, 1, 2, 3, 4]]
def mu6 : List Nat := [2, 1, 0, 3, 4, 5]
def lam6 : List Nat := [1, 0, 2, 3, 4, 5]
theorem rmax6 : RankMaximal 6 P6 mu6
theorem pop6 : Popular 6 P6 lam6
theorem gillies_fails_6 : ¬ UncoveredFor 6 GilliesCovers P6 mu6
theorem separation_six :
    UncoveredFor 6 BordesCovers P6 mu6 ∧ ¬ UncoveredFor 6 GilliesCovers P6 mu6

def P7 : List (List Nat) := [[0, 1, 2, 3, 4, 5, 6], [0, 1, 2, 3, 4, 5, 6],
  [0, 3, 2, 1, 4, 5, 6], [3, 0, 1, 2, 4, 5, 6], [4, 0, 1, 2, 3, 5, 6],
  [5, 0, 1, 2, 3, 4, 6], [6, 0, 1, 2, 3, 4, 5]]
def mu7 : List Nat := [2, 1, 0, 3, 4, 5, 6]
def lam7 : List Nat := [1, 0, 2, 3, 4, 5, 6]
theorem rmax7 : RankMaximal 7 P7 mu7
theorem pop7 : Popular 7 P7 lam7
theorem gillies_fails_7 : ¬ UncoveredFor 7 GilliesCovers P7 mu7
```

(In the source file the lists `P6` and `P7` are written on one line each; they are broken here for width.)

Proposition 7.1.

```
def PE : List (List Nat) := [[0, 2, 1], [0, 1, 2], [1, 0, 2]]
def muE : List Nat := [2, 0, 1]
def lamE : List Nat := [0, 1, 2]
theorem muE_covers_lamE : McKelveyCovers 3 PE muE lamE
theorem lamE_covered : ¬ UncoveredFor 3 McKelveyCovers PE lamE
theorem lamE_pareto : ParetoOptimal 3 PE lamE
theorem lamE_not_rankMaximal : ¬ RankMaximal 3 PE lamE
theorem muE_rankMaximal : RankMaximal 3 PE muE
theorem rankMaximal_not_popular : RankMaximal 4 P4 mu4 ∧ ¬ Popular 4 P4 mu4
theorem majority_tie :
    mdom [[0, 1], [0, 1]] [0, 1] [1, 0] = false ∧
      mdom [[0, 1], [0, 1]] [1, 0] [0, 1] = false ∧ ([0, 1] : List Nat) ≠ [1, 0]
```

Here `ParetoOptimal n P a` says that no assignment weakly improves every agent and strictly improves some agent relative to `a`.

References

[1] F. Brandt, H. Chen, C. Dong, P. Lederer and A. Schlenga, *Majoritarian assignment rules*, arXiv:2602.14816v1 [econ.TH; cs.GT, cs.MA], 16 February 2026; the arXiv comment states that it appears in the Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026). Read in full. Section, remark, example, figure and fact numbers are those of the e-print of version 1 (the only version as of 2026-09-07) compiled from its LaTeX source (17 pages): the majority relation is defined in Section 3, rank-maximality and popularity in Section 4, the three covering relations, the uncovered sets, Example 5, Remark 5 and Remark 6 in Section 4.2, the open question on mixed popular assignments in Section 5, and Fact 1 (the $n = 5$ top-cycle case analysis, "brute forcing through all possible profiles (up to symmetry) with the help of a household computer in two days") in Appendix B. The count 9 078 630 of five-agent profiles up to symmetry is printed in the captions of its Figures 1 and 3. All quotations are verbatim from the compiled e-print.

[2] R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail and K. E. Paluch, *Rank-maximal matchings*, ACM Transactions on Algorithms **2** (2006), no. 4, 602–610. Bibliographic data checked against Crossref (doi:10.1145/1198513.1198520). The origin of rank-maximal matchings and of their polynomial-time computation; cited from the source's bibliography, not consulted for this work.

[3] D. J. Abraham, R. W. Irving, T. Kavitha and K. Mehlhorn, *Popular matchings*, SIAM Journal on Computing **37** (2007), no. 4, 1030–1045. Bibliographic data checked against Crossref (doi:10.1137/06067328X). Cited from the source's bibliography, not consulted for this work.

[4] P. Gärdenfors, *Match making: Assignments based on bilateral preferences*, Behavioral Science **20** (1975), no. 3, 166–173. Bibliographic data checked against Crossref (doi:10.1002/bs.3830200304). The origin of popular matchings; not consulted for this work.

[5] P. C. Fishburn, *Condorcet social choice functions*, SIAM Journal on Applied Mathematics **33** (1977), no. 3, 469–489. Bibliographic data checked against Crossref (doi:10.1137/0133030). Not consulted for this work.

[6] N. R. Miller, *A new solution set for tournaments and majority voting: Further graph-theoretical approaches to the theory of voting*, American Journal of Political Science **24** (1980), no. 1, 68–96. Crossref (doi:10.2307/2110925) records only the first page; the page range is that given in the reference list of [13] as deposited with Crossref. Not consulted for this work.

[7] G. Bordes, *On the possibility of reasonable consistent majoritarian choice: Some positive results*, Journal of Economic Theory **31** (1983), no. 1, 122–132. Bibliographic data checked against Crossref (doi:10.1016/0022-0531(83)90024-8). Not consulted for this work.

[8] R. D. McKelvey, *Covering, dominance, and institution-free properties of social choice*, American Journal of Political Science **30** (1986), no. 2, 283–314. Crossref (doi:10.2307/2111098) records only the first page; the page range is that given in the reference list of [13] as deposited with Crossref. Not consulted for this work.

[9] D. B. Gillies, *Solutions to general non-zero-sum games*, in: A. W. Tucker and R. D. Luce (eds.), Contributions to the Theory of Games, Volume IV, Annals of Mathematics Studies 40, Princeton University Press, Princeton, 1959, pp. 47–85. The page range is zbMATH's (Ann. Math. Stud. 40, 47–85); the publisher's chapter record for the 2016 reissue (doi:10.1515/9781400882168-005) gives 47–86, with the next chapter beginning at page 87, and the reference list of [13] as deposited with Crossref gives no pages. Not consulted for this work.

[10] H. Moulin, *Choosing from a tournament*, Social Choice and Welfare **3** (1986), no. 4, 271–291. Bibliographic data checked against Crossref (doi:10.1007/BF00292732). Not consulted for this work.

[11] G. Bordes, M. Le Breton and M. Salles, *Gillies and Miller's subrelations of a relation over an infinite set of alternatives: General results and applications to voting games*, Mathematics of Operations Research **17** (1992), no. 3, 509–518. Bibliographic data checked against Crossref (doi:10.1287/moor.17.3.509). Cited from the source's bibliography, not consulted for this work.

[12] F. Brandt and F. Fischer, *Computing the minimal covering set*, Mathematical Social Sciences **56** (2008), no. 2, 254–268. Bibliographic data checked against Crossref (doi:10.1016/j.mathsocsci.2008.04.001). Cited from the source's bibliography, not consulted for this work.

[13] J. Duggan, *Uncovered sets*, Social Choice and Welfare **41** (2013), no. 3, 489–535. Bibliographic data checked against Crossref (doi:10.1007/s00355-012-0696-9; published online 2012, in print 2013). The survey that names and compares the Bordes, Gillies and McKelvey covering relations; not consulted for this work beyond its deposited reference list.

[14] F. Brandt, M. Brill and P. Harrenstein, *Tournament solutions*, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang and A. D. Procaccia (eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, Chapter 3, pp. 57–84. Bibliographic data checked against Crossref (doi:10.1017/CBO9781107446984.004). Not consulted for this work.

[15] F. Brandt, M. Brill and P. Harrenstein, *Extending tournament solutions*, Social Choice and Welfare **51** (2018), no. 2, 193–222. Bibliographic data checked against Crossref (doi:10.1007/s00355-018-1112-x). Not consulted for this work.

[16] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks and F. Pollitt, *CaDiCaL 2.0*, in: Computer Aided Verification (CAV 2024), Part I, Lecture Notes in Computer Science 14681, Springer, 2024, pp. 133–152. Bibliographic data checked against Crossref (doi:10.1007/978-3-031-65627-9_7). The solver used is the build shipped with the Lean toolchain.

[17] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction – CADE 28, Lecture Notes in Computer Science **12699**, Springer, 2021, pp. 625–635. Bibliographic data checked against Crossref (doi:10.1007/978-3-030-79876-5_37).

[18] 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. Bibliographic data checked against Crossref (doi:10.1145/3372885.3373824).