LEXIMIN on CCA and HD

Original quotas and uniform-compatible quotas

Computational study for Section 6.4

10 October 2026

1 Main findings

We ran the LEXIMIN algorithm of Flanigan et al. [1] locally on the four instances from the preceding CCA and HD experiments. The accepted runs implement the formal primal formulation in Supplementary Information (SI), Section 10, Algorithm 2, with ILP column generation from Algorithm 1. They are not Panelot website runs. Section 3 specifies the implementation and numerical checks.

Every panel in all four computed lotteries satisfies every quota. Thus the mean quota violation and maximum quota miss are both exactly zero on the entire saved support, and in each of the 10,000 sampled panels per case. With the adjusted quotas, LEXIMIN also reproduces the uniform target probabilities to numerical precision. With the original quotas, its probabilities differ substantially from the prescribed targets.

	LEXIMIN probability (%)		Target difference (pp)	
Instance	Minimum	Maximum	Mean absolute	Maximum absolute
CCA original	2.401	100.000	8.005	64.629
CCA uniform	9.091	9.091	0.000	0.000
HD original	5.145	100.000	9.268	66.624
HD uniform	12.552	12.552	0.000	0.000

Here "pp" means percentage points: a probability difference of 0.01 is one percentage point. Differences are computed respondent by respondent using full-lottery marginals, not finite-sample frequencies. Displayed zeros for uniform cases conceal only numerical residuals: the largest absolute probability errors are 3.36×10^{-14} for CCA and 5.22×10^{-15} for HD.

For the original-quota instances, the mean absolute differences are 8.005 pp (CCA) and 9.268 pp (HD); the corresponding worst differences are 64.629 and 66.624 pp. These results do not support describing the prescribed targets and LEXIMIN probabilities as uniformly "within a few percentage points."

The original-quota minimum probabilities also agree, at the reported precision, with the 2.4% (CCA) and 5.1% (HD) entries in Table 1 of Flanigan et al. [1].

2 Instances and comparison targets

CCA contains 825 respondents, with a panel size of 75, four dimensions, 66 listed traits and 554 distinct trait profiles. Five listed traits are empty, so 61 are nonempty. HD contains 239 respondents, with a panel size of 30, seven dimensions, 33 nonempty traits and 202 profiles. Panel sizes are those used in the previous exercises and the supplied manuscript.

Original-quota case. We retain the original `categories.csv` and compare LEXIMIN with the profile-dependent target vector already constructed for the Theorem 2 experiment. For profile t with multiplicity m_t, that target solves

$$\min_{0\le p_t\le 1} \frac{1}{2}\sum_t m_t(p_t - k/n)^2, \quad \sum_t m_t p_t = k, \quad l_a \le \sum_t m_t A_{at} p_t \le u_a \quad \text{for every trait } a.$$

The previously certified exact rational targets are included in the package. We do not recalibrate them to fit LEXIMIN.

Uniform case. The target is $p_i = k/n$ for every respondent. If N_a respondents have trait a, the adjusted integer bounds are

$$l'_a = \lfloor kN_a/n \rfloor, \qquad u'_a = \lceil kN_a/n \rceil.$$

These are the same adjusted quota files as in the preceding experiments. The uniform probabilities are 1/11 for CCA and 30/239 for HD.

LEXIMIN receives the pool, quotas and panel size. It does not receive the prescribed probabilities as constraints or as an optimization objective. The target column is retained in the replication inputs only for comparison. No household restrictions are added, and no quota is relaxed.

3 Algorithm and verification

3.1 Formal LEXIMIN formulation

Let $\mathcal{P}$ denote the quota-feasible k-person panels, R the respondents whose probabilities are already fixed, and ρ_i their fixed values. At each stage, the restricted master problem is

$$\begin{aligned} &\max\ z \\ &\text{subject to } \sum_{P\in\mathcal{P}_0} \lambda_P = 1, \quad \lambda_P \ge 0, \\ &\qquad \sum_{P\ni i} \lambda_P \ge z && (i \notin R), \\ &\qquad \sum_{P\ni i} \lambda_P = \rho_i && (i \in R), \end{aligned}$$

where $\mathcal{P}_0$ is the current portfolio. We use the LP dual weights to solve an integer pricing problem: find a quota-feasible panel of maximum total weight. A panel with positive reduced cost is added to the portfolio. Pricing continues until its certified objective upper bound exceeds the master dual cap by at most 10^{-9}.

Once a stage is complete, only respondents with a strictly positive dual multiplier on their unfixed lower-bound constraint are fixed. We use a 10^{-8} threshold to distinguish positive multipliers from numerical zero. Each new ρ_i is the corresponding marginal in the current primal lottery, as in Algorithm 2, line 10. We do not fix every respondent who happens to attain the current minimum. All previously fixed probabilities remain equality constraints at later stages.

3.2 Implementation choices

The new runner uses Python 3.12.14, NumPy 2.3.5 and Gurobi 13.0.3. Each solver uses one thread and seed zero. LPs use dual simplex, with primal feasibility and optimality tolerances 10^{-9}. Pricing MILPs use relative gap zero, absolute gap 10^{-10} and integrality tolerance 10^{-9}.

The initial portfolio uses the multiplicative-weight heuristic described in SI 10.2: weights begin at one, selected agents' weights are multiplied by 0.8, and weights are rescaled. We use $\min(3n, 900)$

rounds (900 for CCA and 717 for HD). This changes only the starting portfolio. Subsequent column generation still prices over all quota-feasible panels.

The authors' frozen practical implementation [2] uses a dual LP, barrier solutions and a fallback that decreases fixed probabilities when numerical difficulties arise. Initial tests of that implementation with tightened tolerances invoked this fallback on the original-quota instances. The reported results instead use the paper's formal primal formulation, implemented in `run_leximin.py`. No probability-relaxation fallback is used. This distinction is recorded explicitly; the reference source is included unchanged under `vendor/`.

3.3 Independent checks

For every optimization stage, a separate SciPy 1.17.0/HiGHS MILP recomputes the maximum feasible panel weight and its objective bound. We check normalization of the unfixed dual weights, the fixed sets and values, the objective identity, the pricing gap and agreement of fixed marginals with the final lottery. We also check equal marginals within each trait profile. The acceptance tolerance for these independent checks is 2×10^{-7}; actual residuals are much smaller:

Instance	Positive pricing gap	Fixed marginal error	Within-profile spread
CCA original	1.21×10^{-11}	7.55×10^{-12}	5.33×10^{-12}
CCA uniform	1.35×10^{-14}	3.62×10^{-14}	4.53×10^{-14}
HD original	2.44×10^{-11}	5.33×10^{-12}	5.15×10^{-12}
HD uniform	0.00×10^{0}	5.55×10^{-17}	4.14×10^{-15}

These are floating-point optimality diagnostics, not rational-arithmetic proofs of the exact LEXIMIN vector. Quota compliance has a stronger verification: all panel memberships and trait counts are integers, and every supported panel is checked directly against every bound. A separate verifier recomputes the exported CSV counts, marginals and all 40,000 sampled-panel metrics using respondent identifiers.

4 Full-lottery and sample results

For panel P, trait count $c_a(P)$ and bounds $[l_a, u_a]$, define

$$v_a(P) = \max\{l_a - c_a(P),\ c_a(P) - u_a,\ 0\}, \quad \overline{v}(P) = \frac{1}{M}\sum_{a=1}^{M} v_a(P), \quad v_{\max}(P) = \max_a v_a(P).$$

We average across all listed traits: $M = 66$ for CCA and $M = 33$ for HD. Excluding empty CCA traits still gives zero for LEXIMIN. For a lottery, the mean violation is $\sum_P \lambda_P \overline{v}(P)$, and its worst supported miss is $\max_{P:\lambda_P>0} v_{\max}(P)$.

	LEXIMIN: full support		Theorem 2: 10,000 draws	
Instance	Mean miss	Worst miss	Mean miss	Worst observed
CCA original	0.000000	0	0.430420	3
CCA uniform	0.000000	0	0.097873	3
HD original	0.000000	0	0.269539	4
HD uniform	0.000000	0	0.215327	4

The Theorem 2 columns are snapshots of the previous verified runs, not new simulations in this exercise. They preserve the prescribed marginals in the underlying lottery. LEXIMIN optimizes marginals subject to exact quotas. The comparison therefore includes both quota outcomes and the probability changes shown above.

Instance	Generated panels	Positive-weight panels	LP stages	Distinct sampled panels
CCA original	1591	748	78	638
CCA uniform	901	822	3	798
HD original	790	221	29	204
HD uniform	717	239	1	238

Zero-weight generated panels are retained in `lottery.json` for audit purposes. The file `lottery_support.csv` contains every strictly positive-weight panel; no positive weight is discarded by a support threshold. These panels and their weights specify the complete computed lottery.

For each case we independently draw 10,000 panels from the saved distribution using NumPy PCG64. A lottery can produce the same panel on multiple draws. In all four samples, the histogram of the group maximum miss has 10,000 observations at zero and none elsewhere. The full-support check establishes zero violation beyond these samples, so a Monte Carlo confidence interval for that zero is unnecessary. Empirical inclusion frequencies fluctuate around full-lottery marginals and are recorded separately.

Instance	Respondents	Panel size	Listed traits	Sampling seed
CCA original	825	75	66	20261020
CCA uniform	825	75	66	20261021
HD original	239	30	33	20261022
HD uniform	239	30	33	20261023

5 How the selection probabilities change

For prescribed target p_i and computed LEXIMIN marginal q_i, the reported person-weighted summaries are

$$\text{MAE} = \frac{1}{n}\sum_i |q_i - p_i|, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_i (q_i - p_i)^2}, \qquad \text{MaxAE} = \max_i |q_i - p_i|.$$

Respondents, rather than distinct profiles, receive equal weight. The mean signed difference is numerically zero because both vectors sum to the same panel size.

Instance	RMSE (pp)	Respondents with $\lvert q_i - p_i \rvert$ > 1 pp	> 5 pp	Target zeros	LEXIMIN ones
CCA original	13.951	785	264	424	21
CCA uniform	0.000	0	0	0	0
HD original	15.373	216	131	51	7
HD uniform	0.000	0	0	0	0

Zeros in the prescribed targets are exact. LEXIMIN "ones" use a numerical 10^{-7} tolerance. All respondents have positive LEXIMIN probability in all four instances.

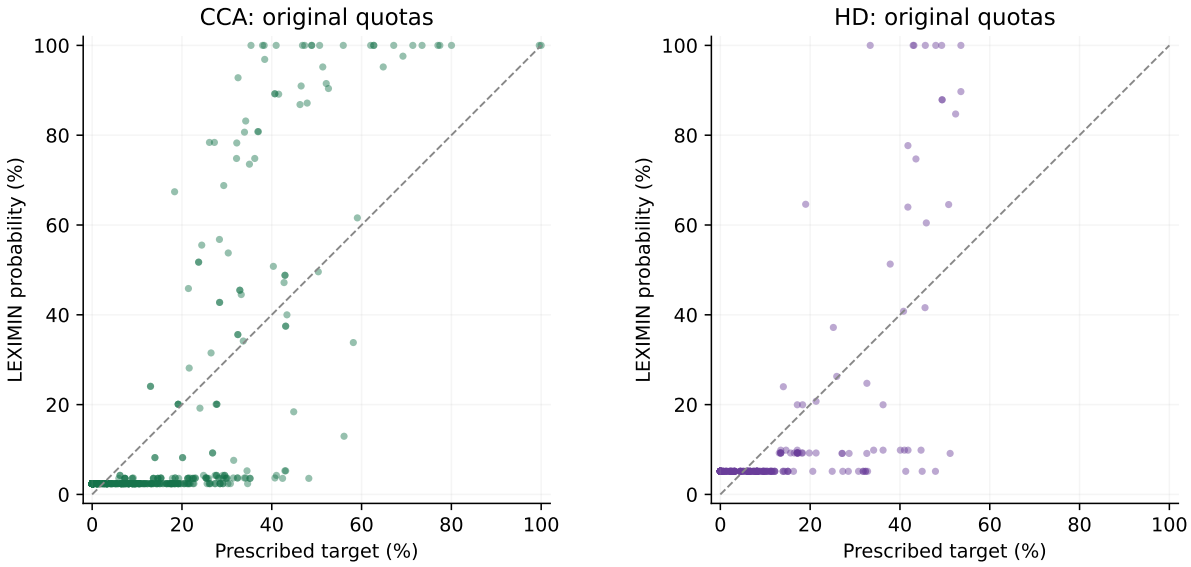


Figure 1: Original quotas: prescribed targets against full-lottery LEXIMIN marginals. Each point represents a respondent; some points overlap. The dashed diagonal denotes equal probabilities.

In the adjusted-quota cases, the computed lotteries simultaneously satisfy all integer quotas and implement the uniform vector to numerical precision. Since every lottery on k-person panels has average marginal k/n, no minimum marginal can exceed k/n. Thus feasibility of the uniform vector also identifies the LEXIMIN optimum in these two cases.

For the original quotas, changing from the previously chosen least-squares calibration to LEXIMIN changes the fairness objective. The observed discrepancies measure that change. They do not establish that every exact-quota lottery must differ from the prescribed targets: testing whether those particular vectors lie in the convex hull of feasible panels is a separate feasibility question. These four instances also do not establish general performance or runtime dominance of either approach.

6 Replication files and use

The package includes all inputs, full lotteries, sampled panels, per-trait outcomes, respondent-level probability comparisons, solver settings and independent checks. The concise summary is `results_summary.csv`; the same information is available as `results.json`. Each instance directory includes a `summary.json` with numerical diagnostics.

Reproduce the computations. From the unzipped package directory, create a Python 3.12 environment and install `requirements.txt`. Then run:

```
OPENBLAS_NUM_THREADS=1 OMP_NUM_THREADS=1 python run_leximin.py
OPENBLAS_NUM_THREADS=1 OMP_NUM_THREADS=1 python analyze_results.py
python verify_saved_files.py
python build_report.py
pdflatex LEXIMIN-results-report.tex
pdflatex LEXIMIN-results-report.tex
```

A Gurobi license is needed to rerun optimization. The bundled restricted non-production license sufficed for these recorded models. Gurobi is not needed to analyze or verify the supplied lotteries. Exact supporting panels can differ with solver versions or tie-breaking; the relevant reproducibility checks concern marginals, certified pricing gaps and quota compliance. Recorded elapsed times are not controlled runtime benchmarks.

The input preparation script documents how the earlier experiments were snapshotted. Its outputs are already included, so the earlier directories are not needed to reproduce this benchmark. The exact target fractions, original-case calibration certificates and previous Theorem 2 summaries are included to make the comparison auditable. The manuscript snippet is optional; this exercise does not modify the revised paper or response letter automatically.

Panelot cross-check. Each instance also includes `pool.csv` with identifier header `pool_member_identifier`, followed by the trait columns. At `https://panelot.org/compute_lottery`, upload that file together with the corresponding `categories.csv`; enter panel size 75 for CCA or 30 for HD. The targets are omitted because Panelot determines its own probabilities. Download the complete resulting lottery if an independent comparison is desired. No data were submitted to Panelot in this exercise.

References

[1] B. Flanigan, P. Gölz, A. Gupta, B. Hennig and A. D. Procaccia. *Fair algorithms for selecting citizens' assemblies*. Nature **596**, 548–552 (2021). doi:10.1038/s41586-021-03788-6. Supplementary Information, Sections 8 and 10, Algorithms 1 and 2.

[2] P. Gölz and collaborators. *citizensassemblies-replication*. `https://github.com/pgoelz/citizensassemblies-replication`. Reference snapshot: commit `6ad2d5fb2b22dc27d66b1d6e7c93c141967802de`; accessed 10 October 2026. Vendored source retains its GPL-3.0 license.