

Auditing policy dependent benchmarks in allocation simulations

Pablo Loschi

Independent researcher, Berlin, Germany

ORCID 0009-0004-9455-4713

14 October 2026

Abstract

Allocation policies change which agents remain available for later decisions. A benchmark built from the best currently available agent therefore depends on the policy being evaluated. This note provides a worked audit of that measurement choice. In a two-period housing example, the moving-reference ratio rises while realised value falls from 16 to 11. A matched-seed housing sweep reproduces that disagreement. In a separate deadline queue, value rises from 19 to 27 while the moving ratio falls from 1.00 to 0.90; 15 of 200 fixed-seed cases show this direction of disagreement. A reusable routine reports numerator and denominator changes alongside a common reference. Feasibility checks distinguish an exact offline optimum from an upper bound. A completed-cohort check clarifies the population behind access gains. The contribution is a portable reporting diagnostic demonstrated in two illustrative simulations. It does not establish the prevalence of this problem in published models.

Keywords: simulation verification; policy evaluation; endogenous benchmark; dynamic allocation; agent-based model; reproducibility

1. Introduction

Consider a simulation that allocates homes to waiting buyers. A policy changes who wins today, and the winner leaves the queue. If the evaluator compares tomorrow's allocation with the highest valuation in its own queue, the policy changes the reference used to score it. An improvement in the resulting ratio can accompany a fall in total allocation value.

This note examines that problem in a fixed-offer housing simulation that compares limits on institutional purchases with adjustments to their offers. Its current-pool ratio measures selection along a realised path. Treating a change in that ratio as an aggregate value gain overlooks the policy's effect on the comparison set. A buyer who loses can be counted repeatedly among the best currently available alternatives; serving that buyer removes it from later maxima.

The audit retains the realised numerator and the moving reference for each policy path, then reports their changes separately. A fixed reference supplies a comparison with the same objective and exogenous inputs. The reference may be an attainable optimum or a labelled upper bound. These interpretations have different implications for the level of the ratio, even though either fixed positive denominator preserves the sign of a value change within a matched pair.

A complete two-period example makes the calculation inspectable. A larger housing sweep uses eight matched seeds at each of nine offer discounts. A separate deadline-based service queue tests the same diagnostic with different timing and an exactly calculated offline optimum. Its fixed-seed cases include agreement as well as disagreement between value and moving-ratio comparisons. A final check addresses a related reporting choice: access gains over all arrivals during a finite run differ from gains for a completed interior entry cohort.

The diagnostic returns the value change and reference change for each matched pair, then checks whether the reported ratio reverses their aggregate-value comparison. A pool-relative ratio can be useful for describing selection conditional on the pool encountered by a policy. Its change answers a different question from the change in total allocation value. Reporting both quantities lets readers distinguish those questions. A user can apply the same routine to saved totals from another simulation once the objective

and common exogenous inputs are justified. The ratio identity is elementary, and decision-dependent evaluation has established precedents. Both programs are illustrative applications; no external published implementation is audited. The supplied comparison routine accepts totals without importing either model; researchers still need to justify their input matching, objective and reference.

2. An exact two-period example

There is one listing in each of two periods. Each buyer can purchase at most once. Buyer I is an institution present only in period 1, with valuation 6 and certainty factor 1. Buyer O is a household with valuation 10 and certainty factor 0.5, present in either period until served. Buyer L arrives in period 2, with valuation 1 and certainty factor 0.5. Listings are allocated immediately; they cannot be held for later sale. Sellers rank offers by valuation multiplied by the certainty factor.

Without a charge, I's effective offer of 6 beats O's offer of 5 in period 1. O then beats L in period 2. The realised allocation value is $6 + 10 = 16$. The highest valuation in the pool is 10 at both decisions: O remains available after losing the first listing. Adding those two local references gives 20, so the moving-benchmark ratio is $16/20 = 0.80$. The denominator uses O twice, although O can purchase only once.

Table 1 compares these allocations. Now reduce institutional effective offers by 20%. I's offer becomes 4.8, so O wins the first listing. I expires after period 1. L is the only buyer left for the second listing, giving realised value $10 + 1 = 11$. The local reference is 10 followed by 1. The moving-benchmark ratio is $11/11 = 1.00$.

Table 1. Exact example with fixed arrivals and two immediate listings. The local reference includes all present buyers before the charge. The feasible optimum respects arrival, expiry and one purchase per buyer.

Evaluation	No charge	20% charge	Change
Winners, periods 1 and 2	I, O	O, L	Different allocation path
Realised value	16	11	-31.25%
Sum of highest present valuations	20	11	-45.00%
Ratio to highest present valuations	0.80	1.00	+20 percentage points
Ratio to feasible optimum of 16	1.00	0.6875	-31.25 percentage points

The feasible schedules can be enumerated. They include (I, O), (I, L) and (O, L), with values 16, 7 and 11. Thus 16 is the true dynamic optimum here. Giving O the first listing improves that listing's realised value by 4, but sacrifices the chance to serve I. The later loss of 9 more than offsets the immediate gain. The example needs timing and one-purchase constraints; it does not rely on strategic bids, stochastic failures or uncertain parameter estimates.

For any policy p , let V_p denote total realised value and B_p the sum of highest present valuations. Define $E_p = V_p / B_p$. For positive quantities, the comparison satisfies

$$\frac{E_1}{E_0} = \frac{V_1/V_0}{B_1/B_0}, \log \frac{E_1}{E_0} = \log \frac{V_1}{V_0} - \log \frac{B_1}{B_0}. (1)$$

This is a ratio identity. If $B_1/B_0 < V_1/V_0 < 1$, realised value falls while E rises. A corresponding reversal is possible when both quantities increase and the benchmark grows faster than the numerator. Calling E an efficiency gain without reporting B conceals this possibility.

A fixed positive reference F gives $V_1/F - V_0/F$ the same sign as $V_1 - V_0$ for each run. No optimality property of F is required for that sign agreement. Interpreting V/F as proximity to an economically

relevant optimum does require a justified feasible benchmark. The small example supplies one; the larger simulation uses a looser upper bound and labels it accordingly.

The executable file `replication/benchmark_toy.py` constructs both paths and enumerates the feasible schedules using only Python’s standard library. Its assertions check the values in Table 1. It is a readable diagnostic of an elementary ratio comparison.

3. Relationship to existing work

The service queue belongs to the established bounded-delay scheduling problem. Kesselman et al. (2004) analyse packets with values and deadlines, compare online policies against an offline optimum on the same input, and give efficient offline algorithms. An optimum respecting timing and capacity is therefore a standard reference for this problem. The small dynamic program used here is a verification tool. It supplies no new scheduling algorithm or competitive bound. The measurement question is what a sum of current maxima records when the policy changes the queue.

Arora, Dekel and Tewari (2012) show why regret comparisons can mislead when losses depend on an action history. Their policy-regret comparator accounts for the history induced by the alternative policy. The queue examples share the concern about policy-dependent histories, but contain no adaptive adversary or learning algorithm. Their regret results are not applied here.

Perdomo et al. (2020) study prediction rules that change the distribution on which they are evaluated. The present comparison set changes because served agents leave a queue. There is no retraining, stability or convergence result. The connection locates this diagnostic within established work on decision-dependent evaluation.

Bertsimas, Farias and Trichakis (2011) measure the price of fairness against a utilitarian allocation in a specified feasible problem. That common reference helps distinguish the feasible optimum in the service queue from the temporally unconstrained upper bound used in the larger housing model. Windrum, Fagiolo and Moneta (2007) distinguish the empirical validation of agent-based models from program verification. The checks here concern implementation and measurement. They provide no evidence that the maintained rules predict a city’s housing outcomes.

Furtado’s (2022) PolicySpace2 includes interacting labour, credit and housing markets with empirical spatial inputs. It illustrates the broader modelling work needed for a policy application. No benchmark error is alleged in that model. Appendix A describes the smaller housing program using the reporting concerns of the ODD protocol (Grimm et al. 2020).

4. The housing simulation and audit design

4.1 Buyers, listings and timing

The inherited model has five districts and 120 periods. In each district-period, household arrivals average 17 and small-investor arrivals average 4. Their valuations are lognormal, with medians 400,000 and 372,000 and log standard deviations 0.34 and 0.30. Households can remain for 36 periods and small investors for 24. A household leaving at its deadline without buying is recorded as an unserved exit.

Three institutional entities generate a total of 42 fresh buyer-instances per district-period. Their valuation median is 366,000 and log standard deviation 0.18. Each instance can win once and is available only in the current period. An entity can win repeatedly through different instances. This is a model of repeated institutional demand rather than a population of persistent firms with optimised budgets.

Fresh listings are Poisson draws averaging 20 per district-period. Buyers bid in a common local market and each listing is allocated sequentially. Valuations are fixed and bids are not strategic. A seller scores an offer as valuation times the buyer-class certainty factor, then applies any policy adjustment. The baseline factors are 0.88 for households, 0.95 for small investors and 1 for institutions. The main audit sets actual

transaction failures to zero. The certainty factors are therefore maintained offer discounts; they are not estimated failure probabilities or realised completion costs.

For each district-period, the program generates civilian arrivals and valuations, removes expired buyers, generates institutional instances and draws fresh listings. It calculates a policy signal from earlier exits, processes listings and records current exits. After each success, it removes the winner from the relevant pool. At the end of a period, unused institutional instances disappear while civilian buyers remain subject to patience. This sequence explains how a policy changes the comparison set even with unchanged arrivals.

4.2 Instruments used to generate alternative paths

The main audit compares an unconstrained regime with a contention-linked charge. The charge multiplies institutional effective offers by $1 - 0.22s$, where s is the fraction of household exits that were unserved in the district's trailing six periods. Purchases count as served exits. When the denominator is empty the signal is zero. The signal is updated between periods. This is an offer-ranking adjustment. The model does not calculate tax revenue or determine which participant bears a statutory charge.

The access-population check also includes a per-entity cap, a fair-share rule and a constant charge. The cap limits cumulative institutional acquisitions by each entity within a district over the full run. The fair-share rule restricts institutional wins within each period while households are waiting. It releases that restriction if the household pool is empty. The constant charge reduces institutional offers by a fixed fraction. Settings for these instruments are inherited from an exercise targeting a five-percentage-point reduction in the full-run unserved-exit fraction.

These rules supply contrasting allocation paths. Their comparison is conditional on the model's information, eligibility and fixed bids. In particular, the contention signal presumes that entry, purchase and withdrawal can be observed. Routine completed-transaction records do not supply that registry. The audit does not establish that the charge could be implemented at these settings.

4.3 Outcomes and references

Total realised value V is the sum of successful buyers' valuations. The model stores it under the legacy key welfare; the manuscript uses the more restricted interpretation. It excludes transfer payments, administrative costs, housing consumption after purchase and differences in the marginal utility of income. It is neither observed transaction revenue nor a full social-welfare measure.

The moving reference B adds the highest valuation among all bidders present at each listing, before the policy eligibility or charge is applied. A buyer who remains in the pool can appear in several of these maxima. B is consequently not the value of a feasible allocation, even when the current pool is fully observed. V/B can describe selection against the local pool encountered by the policy. It cannot isolate changes in total allocation value across policies that induce different pools.

The fixed reference F is the sum of the L highest valuations among all buyer-instances generated during the run, where L is the number of fresh listings. It can use future entrants for earlier listings and ignore buyer expiry and location constraints. It also ignores sequential availability. With zero failures and exogenous inputs, feasible realised allocations cannot exceed this upper bound. If V^* is the best feasible allocation value, then $V \leq V^* \leq F$ and $V / F \leq V / V^* \leq 1$. A ratio close to one therefore does certify a small possible gain under this valuation objective. It does not identify the exact feasible optimum or the precise avoidable loss. The audit reports V directly alongside this common normalisation.

All ratios are calculated for each seed before taking averages. The realised-value index is separately calculated as 100 times the ratio of mean policy value to mean unconstrained value. Averages of ratios weight runs differently from ratios of totals. The exact sign agreement for a fixed F is therefore stated per seed. The audit additionally reports pooled ratios so the aggregation choice is visible.

4.4 Fresh reproduction checks

The offer-discount grid is $w = 0, 0.04, 0.06, 0.07, 0.08, 0.10, 0.12, 0.16$ and 0.20 . The household certainty factor is $1 - w$; the small-investor factor is $\min(1, 1 - w + 0.07)$. Each grid point uses eight seeds beginning at 20260824 and increasing by 100. The two policy paths share a seed. This gives 72 pairs and 144 runs. Changing w also changes small-investor certainty, so this is a sensitivity path rather than an isolated variation of household certainty.

The original `sim_v6.py` is imported unchanged. Its value and ratio outputs allow recovery of B and F. The audit saves their levels, checks equality of F within each pair and verifies equation (1). Every numerical field in the earlier zero-failure sweep is compared with its fresh counterpart. Appendix A explains why full search and zero failures matter to common-input matching.

For the access check, the existing cohort diagnostic adds counters to the inherited model. It follows entrants from periods 36 through 83 inclusive; the last entrant reaches its 36-period deadline by period 119. Fresh results are compared field by field with the archived diagnostic, and hashes of valuation draws, fresh listings and fixed references are checked across policies. The instrument settings are frozen; there is no fresh parameter search on the evaluation seeds.

These checks address reproducibility and the stated comparison. They do not test all structural assumptions or prove that any policy is optimal. Online Resource 1 supplies the simulation code and seed-level outputs, including the numerator, references and completed-cohort counts. It also contains the portable routine and the exhaustive service-solver check.

5. Results of the audit

5.1 A gain created by a falling denominator

Table 2 reports the housing sweep.

Table 2. Fresh matched-seed sweep, eight seeds per offer discount. Value and reference changes are percentages based on ratios of totals. Ratio gains are percentage-point differences of mean per-seed ratios. F is the fixed top-valuations upper bound; B is the policy-dependent sum of current maxima. The final column counts opposite signs for realised value and the moving-ratio change within a seed.

Offer discount	Value change (%)	B change (%)	Gain vs F (pp)	Gain vs B (pp)	Opposite signs
0%	-0.561	+1.134	-0.559	-1.676	0/8
4%	-0.433	+0.131	-0.431	-0.558	0/8
6%	-0.352	-0.573	-0.351	+0.218	8/8
7%	-0.307	-0.946	-0.306	+0.632	8/8
8%	-0.249	-1.332	-0.248	+1.071	8/8
10%	-0.117	-2.096	-0.117	+1.950	8/8
12%	+0.024	-2.754	+0.024	+2.725	1/8
16%	+0.356	-3.726	+0.353	+3.936	0/8
20%	+0.749	-4.483	+0.740	+4.925	0/8

At $w = 6\%$, the charge lowers mean realised value by 0.352% and lowers the pooled moving reference by 0.573%. The moving ratio rises by 0.218 percentage points. The fixed-reference ratio falls by 0.351 points. At $w = 10\%$, the value loss is smaller, 0.117%, while the moving reference falls by 2.096%. The moving-ratio gain is 1.950 points. The ratio records a change in the queue as well as a change in allocation value.

All eight seeds at each of the 6%, 7%, 8% and 10% nodes give these opposite signs. At 12%, both mean comparisons are positive, although one seed has a negative value change and a positive moving-ratio change. The grid contains 33 opposite-sign pairs out of 72. That count describes these selected scenarios and seeds; it is not a frequency estimate for housing policies or arbitrary model parameters.

At zero discount both mean comparisons are negative. At 16% and 20% both are positive. The sign reversal is therefore not universal. The moving reference can make a loss appear as a gain for intermediate inputs, while agreeing in sign elsewhere. The diagnostic reports the agreement cases as well as the reversals; the offer-discount grid does not establish how frequently either pattern occurs in other models.

Fig. 1 separates the ratio changes from the changes in their components.

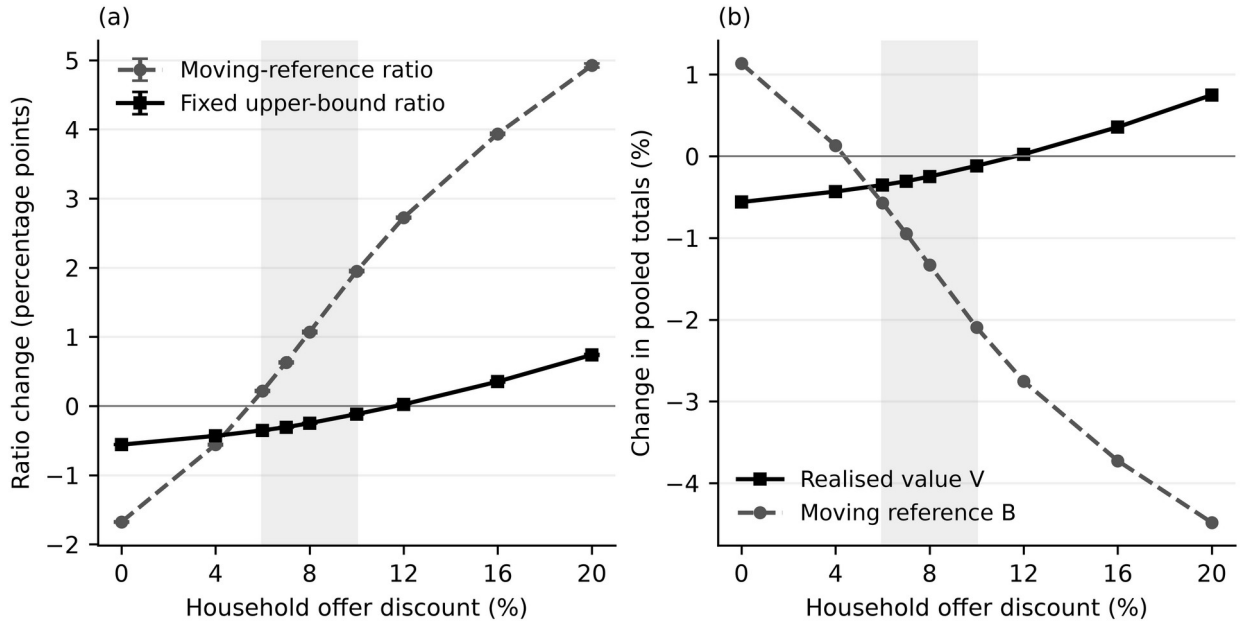


Fig. 1 Changes in the two evaluation ratios and their components. Panel a reports matched-seed mean gains; error bars are plus or minus one paired Monte Carlo standard error across eight seeds. Panel b shows percentage changes in pooled realised value and pooled moving reference. The reference drops faster than value at the four interior nodes with consistent sign reversal. Uncertainty covers simulation draws at fixed settings, not model or empirical parameter uncertainty. The figure was generated with Codex-assisted Matplotlib code from inspectable simulation outputs

The reproduced runs match every inherited sweep field exactly in the tested runtime. Within each pair, the recovered fixed references are identical. The largest residual in the logarithmic identity is approximately 2.31×10^{-16} . The matched inputs and recovered totals locate the reversal in the policy paths and their moving references.

Pooled and mean-per-seed moving-ratio comparisons have the same signs at all nine nodes. Their largest difference is below 0.0003 percentage points. That agreement is a numerical result for this exercise. It does not make the two aggregation methods equivalent. A comparison with heterogeneous run sizes or references should specify which weighting answers the intended question.

5.2 What the ratio still describes

The moving ratio answers a local question: how do winners' valuations compare with the highest valuation encountered at each decision along that policy's realised path? That can be useful when describing the selection rule. It becomes misleading when interpreted as a comparison of aggregate realised value or proximity to a common feasible optimum.

A fixed upper bound avoids this particular sign problem under matched exogenous inputs. Its level gives a conservative efficiency bound: an allocation that reaches 99.5% of F must also reach at least 99.5% of the feasible optimum under the same valuation objective. The remaining gap to F is an upper bound on

avoidable loss, not an exact measure of it. Comparing V across matched paths is sufficient to establish the sign of the value change.

The audit therefore reports the realised-value change as the principal aggregate outcome. The two ratios illustrate the measurement issue. Neither ratio supplies a complete welfare assessment, and the baseline result gives no basis for an optimal tax calculation. A change in buyer composition can have consequences outside realised valuations, including housing use and distribution, that this program does not measure.

5.3 The population behind an access gain

The housing program’s full-run statistic divides unserved household exits by all arrivals during 120 periods. Some late entrants are still waiting at the horizon. It measures observed exits, not the lifetime failure probability of all entrants. The completed cohort follows entrants in periods 36 through 83, all resolved by period 119. Excluding early and late entry changes the population as well as completing follow-up. Table 3 reports gains for both populations.

Table 3. Fixed settings previously tuned toward a five-point full-run access gain; 16 held-out seeds. A gain is the reduction in the relevant unserved fraction relative to the unconstrained regime. Parentheses contain paired Monte Carlo standard errors for completed-cohort gains. All cohorts resolve within the run. Different gains describe different entry populations.

Instrument	Frozen setting	Full-run gain (pp)	Completed-cohort gain (pp)
Per-entity cap	336 acquisitions	5.42	13.53 (0.28)
Period-reset fair share	0.5000 quota share	5.78	7.91 (0.13)
Contention charge	0.2115 maximum haircut	5.03	9.34 (0.07)
Constant charge	0.0779 fixed haircut	4.97	7.15 (0.07)

The unconstrained completed-cohort unserved fraction is 60.09%. The fair-share rule exceeds the contention charge on the full-run gain, while the charge exceeds it for the completed cohort. A cumulative cap and a charge responding to earlier exits also expose different entry dates to different policies. Comparisons at equal access therefore need an entry population and follow-up rule defined before tuning.

Fresh reproduction of these five configurations required 80 runs. Every selected archived numerical field reproduced exactly, every cohort had zero remaining members, and exogenous input checks matched within each seed. The paired standard errors describe simulation randomness. They exclude uncertainty about patience, entry rates and the relevance of the cohort to a real market.

6. Portability to a service queue with deadlines

The second model schedules unit-service jobs in discrete slots. Each job has a release slot, an inclusive deadline and a fixed value. It can be served once. Highest-value-first (HV) selects the largest available value. Earliest-deadline-first (EDF) selects the earliest deadline, breaking ties by value and then identifier. Service takes place after arrivals and before that slot’s deadline. Both policies see current jobs only. The offline reference can see all arrivals and respects every release, deadline and one-job-per-slot constraint.

In a constructed three-slot example, job A has value 10 and is available in slots 0 through 2. B has value 9 and is available only in slot 0; C has value 8 and is available only in slot 1; D has value 1 and is available only in slot 2. HV serves A, C and D. EDF serves B, C and A, retaining A for later service. The moving reference again sums the largest available value before each choice. Table 4 compares the resulting paths.

Table 4. Constructed service queue example. The exact offline optimum is 27. Relative to HV, EDF raises value by 42.11% and the moving reference by 57.89%, so V/B falls by 10 percentage points. Each job uses one slot and can be served once.

Policy	Service order	Realised value V	Moving reference B	V/B	V/feasible optimum
Highest value first	A, C, D	19	19	1.00	0.704
Earliest deadline first	B, C, A	27	30	0.90	1.000

The reference rises because A remains available for all three decisions under EDF. Local greedy selection gives HV a ratio of one by construction, despite its lower total value. EDF reaches the feasible optimum in this example. It is an online heuristic and need not do so in other queues.

The fixed-seed check uses six slots, two arrivals per slot, values uniform on the integers 1 through 20, and deadline extensions uniform on 0, 1 and 2 slots, truncated at slot 5. All 200 consecutive seeds from 20261006 through 20261205 are retained. EDF raises value while lowering V/B in 15 cases; 162 cases have nonzero changes with matching signs, and 23 have a tie in at least one comparison. No case has falling value and a rising moving ratio. These counts describe this deliberately small design, with HV's moving ratio constrained to one.

A dynamic program computes the feasible optimum for every case. A separate full enumeration checks it on all 5,832 three-job configurations with three slots, all feasible release-deadline intervals and values in $\{1, 2, 3\}$. Each emitted path is checked for availability, one-use capacity and reconstructed totals. The same generic routine rechecks all 72 housing pairs and 200 service pairs. Undefined zero-denominator and log comparisons return null; negative or nonfinite totals are rejected. The largest service-suite log-identity residual is below 2.1×10^{-16} . This establishes portability of the calculation; its usefulness in an external research model remains to be tested.

7. Using the audit

A policy comparison should retain V and its reference for each run. The generic compare routine takes V and B for two paths and one fixed F. It returns percentage changes, ratio gains and the log decomposition in equation (1). The input contract handles zero totals explicitly. It cannot establish that policies face the same exogenous problem; the model-specific checks must do that.

Common seeds are sufficient only when the random-number schedule remains independent of policy decisions. Full search and zero failures preserve that schedule in the audited housing program. Reduced search and failed transactions require additional care, because choices can change later draws or relistings. The deadline queue instead generates its complete job list before either policy runs. Its offline solver shares that list and the service constraints.

The reference also needs a stated interpretation. Summed current maxima describe opportunities encountered on a policy's path and can count an agent repeatedly. A global top-values reference is an upper bound that omits constraints. An exact feasible optimum solves a different comparison problem. Keeping these quantities distinct prevents a local selection score from being read as aggregate value or an exact welfare loss.

Aggregation can introduce a further difference. Mean ratios, ratios of pooled totals and averages of percentage changes weight runs differently. Equation (1) applies within each pair. Pooled comparisons require a consistently pooled numerator and denominator. The housing audit reports both calculations and checks their observed agreement in sign.

The access outcome needs its own population contract. Successful buyers' waiting times omit unserved exits. A completed cohort resolves both kinds of outcome for its members, but may differ from other entry cohorts. Instrument settings tuned against one population should be reevaluated or retuned before being described as equal-access settings for another.

The present checks remain narrow. Housing bids are fixed, institutions do not choose entry or financing, and the central audit fixes supply. Realised valuations exclude transfers and much of housing welfare. The service queue has unit-length jobs, a short horizon and no behavioural responses. Its disagreement count is conditional on those choices. Neither exercise measures how often other simulation studies have used a misleading reference.

8. Conclusion

A policy-dependent denominator can reverse a comparison of simulated allocation value. In the housing model a higher moving-reference ratio can accompany lower value. In the service queue an improved feasible allocation can receive a lower moving ratio. Reporting the numerator and denominator separately exposes both outcomes.

The supplied routine applies to totals from either program, while the model-specific checks establish timing and common inputs. The housing reference is an upper bound; the service reference is an exact feasible optimum. A completed-cohort check also shows why access comparisons depend on entry population and follow-up.

These results support an executable reporting diagnostic. Both models are illustrative, and the concern about decision-dependent evaluation is established. Testing the audit in an external simulation would provide stronger evidence of its practical use.

Use of generative AI

Anthropic's Claude assisted earlier drafting, code development and literature searches. OpenAI's Codex assisted the literature comparison, numerical reproduction, service-queue implementation, manuscript drafting and formatting. The figure was drawn by Matplotlib code developed with Codex assistance from the reported simulation outputs. These uses include substantive drafting and coding. The computational checks were AI assisted; they do not constitute independent human review. The author retains responsibility for the work.

References

- Arora, R., Dekel, O., and Tewari, A. (2012). Online bandit learning against an adaptive adversary: From regret to policy regret. *Proceedings of the 29th International Conference on Machine Learning*. Author copy: <https://oferdekel.github.io/pdf/2012AroraDeTe.pdf>.
- Bertsimas, D., Farias, V. F., and Trichakis, N. (2011). The price of fairness. *Operations Research*, 59(1), 17-31. <https://doi.org/10.1287/opre.1100.0865>.
- Furtado, B. A. (2022). PolicySpace2: Modeling markets and endogenous public policies. *Journal of Artificial Societies and Social Simulation*, 25(1), 8. <https://doi.org/10.18564/jasss.4742>.
- Grimm, V., et al. (2020). The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 7. <https://doi.org/10.18564/jasss.4259>.
- Kesselman, A., Lotker, Z., Mansour, Y., Patt-Shamir, B., Schieber, B., and Sviridenko, M. (2004). Buffer overflow management in QoS switches. *SIAM Journal on Computing*, 33(3), 563-583. <https://doi.org/10.1137/S0097539701399666>.
- Perdomo, J. C., Zrnic, T., Mendler-Dünner, C., and Hardt, M. (2020). Performative prediction. *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119, 7599-7609. <https://proceedings.mlr.press/v119/perdomo20a.html>.
- Windrum, P., Fagiolo, G., and Moneta, A. (2007). Empirical validation of agent-based models: Alternatives and prospects. *Journal of Artificial Societies and Social Simulation*, 10(2), 8. <https://jasss.soc.surrey.ac.uk/10/2/8.html>.

Appendix A Model description and reproducibility

A.1 Purpose, entities and state

The model is intended to expose how repeated allocation changes the queue used to evaluate a policy. It is not fitted to the distribution of household search histories in a city. Following the ODD reporting structure, its entities are buyer-instances, institutional entity identifiers and district-period listings. Civilian instances store buyer class, valuation and entry time. Institutional instances also store an entity identifier. District state comprises waiting buyers, acquisitions by entity and a rolling exit history. The evaluation routines store realised value, the current-pool reference and all generated valuations.

There are no spatial links between districts in the core program. All current civilian buyers observe all local listings under full search. Valuations and patience are fixed at entry. The score does not vary with elapsed waiting time. The model's central interaction is competition among current offers and the removal or expiry of bidders. Institutional entities generate capacity repeatedly but have no endogenous balance sheet or intertemporal objective.

A.2 Scheduling and stochastic inputs

The schedule is arrivals, civilian expiry, institutional instances, listings, lagged signal calculation, sequential allocations and exit-history update. An instance can win at most once. Institutional instances disappear at the period boundary. Civilian expiry occurs when current period minus entry period reaches the specified patience, before the period's listings are processed. These scheduling choices are part of the feasible problem, not implementation details that can be ignored by a dynamic optimum.

Poisson arrival and listing draws and lognormal valuation draws use the main random generator. At full search the number of these draws is independent of winners. A separate generator is reserved for transaction failures. The full-search, zero-failure audit does not activate it. The inherited reduced-search diagnostic separates search draws from exogenous inputs; those extensions are outside the reported audit. Model parameter and source hashes accompany the new outputs.

The quota checks institutional wins against quota share times current listings before choosing each winner. Because wins are integers, a period-reset rule can permit a purchase before the quota threshold is crossed. It is not a smooth fractional allocation mechanism. The cap accumulates acquisitions over all periods. These rules help explain why entry cohort and accounting period matter to comparisons.

A.3 Commands and provenance

Run `python replication/benchmark_toy.py` for Table 1. Run `python replication/benchmark_audit.py` for Table 2 and the seed-pair file. Run `python replication/cohort_audit.py` for Table 3. Run `python build_benchmark_figures.py` to draw Fig. 1 from the saved outputs. Online Resource 1 contains the code, run-level results and reproduction instructions.

The housing sweep contains 144 runs; the cohort audit contains 80. Every inherited field selected for those checks reproduced exactly in Python 3.14.7 with NumPy 2.5.3. The cohort checks additionally verified valuation hashes, fresh listings and fixed-reference equality. This is a targeted fresh reproduction, not a rerun of every historical sensitivity or proof of empirical accuracy.

A.4 Service queue and portable comparison

Run `python replication/deadline_queue.py` to regenerate Table 4, all 200 service cases and the independent enumeration check. Run `python replication/portable_audit.py` to apply `compare_benchmarks.py` to the saved housing and service totals. The service queue and portable comparison use Python's standard library; no external data or network access is required. The service example is separately implemented and imports no housing code.

The offline dynamic program maximises value over (slot, served-job set) states. It permits an idle slot or any unused job whose inclusive release-deadline interval contains the current slot. Each transition advances one slot. This recurrence enumerates every feasible next action. Its exponential served-set state space is suitable for the small verification design, not a proposed large-scale scheduler. The independent verifier instead enumerates complete slot assignments, including idling, then rejects repeated jobs and interval violations.

The constructed four-job example is reported separately from the random suite. The random generation design and full consecutive seed range were fixed before inspecting outcomes; all cases appear in `audit/deadline_runs.json` and `deadline_pairs.csv`. Mean value-to-optimum ratios in that suite are 0.9869 for HV and 0.8940 for EDF. EDF's success in Table 4 therefore establishes no general superiority over HV. Its policy is included to change queue composition and test the diagnostic.