A polynomial time algorithm for computing exact equilibria in Bayesian simple security games

Shiva Kintali*

November 23, 2026

Abstract

We give two algorithms for computing an exact Bayes–Nash equilibrium of a simultaneous Bayesian simple security game with an explicitly listed rational input of L bits. Our deterministic algorithm takes $\widetilde{O}(L^4)$ bit operations. Our Las Vegas algorithm takes $\widetilde{O}(L^{10/3})$ in expectation, or $\widetilde{O}(L^3)$ when the numbers of types and targets are within constant factors.

Use of AI: The author discovered various parts of the proofs using Codex and combined the ideas and improved the running time iteratively. During this process, the author obtained several interesting algorithms with incrementally better running times: $L^{O(\sqrt{d})}$ where $d = \min(k, m)$, $poly(N, 1/\epsilon)$, $(2+d)^{O(d)}poly(L)$, $(2+L)^{O(log(2+n))}$, $O(L^{71})$ and finally a deterministic $\widetilde{O}(L^4)$ algorithm and randomized $\widetilde{O}(L^{10/3})$ algorithm. This paper presents only the best deterministic and randomized algorithms discovered so far.

1 Introduction

Security games model how a defender allocates scarce resources across targets while facing a strategic attacker. A Bayesian model allows the attacker to have a privately known type that changes the relative value of different targets. Marginal coverage probabilities give a compact representation of randomized defenses in many security games [16, 20, 27]. Conitzer asks whether this structure permits exact Bayes–Nash equilibrium computation in polynomial time when both the number of targets and the number of types can grow [8]. The complexity of general Nash search motivates the question [10]. Finding an exact Nash equilibrium of an explicitly tabulated two-player game is PPAD-complete [6].

In Conitzer's model the defender chooses a subset of exactly r of $m \geq 1$ targets, where $0 \leq r \leq m$ is integral. The attacker observes a type i drawn from a rational prior $(\pi_1, \ldots, \pi_k)$ on $k \geq 1$ types and chooses one target. Each π_i is nonnegative and the prior sums to one. The defender does not observe the type. Play is simultaneous, so neither player observes the other's action before choosing its own. Each payoff depends on the attacked target and whether it is covered. The attacker's payoff may also depend on its type. The defender's gain from covering an attacked target j is $g_j \geq 0$, independent of the attacker's type. If x_j denotes the defender's probability of covering j, the expected cost to type i of attacking j is

$$F_{ij}(x_j) = a_{ij} + b_{ij}x_j, \qquad b_{ij} \geq 0,$$

*Email: shiva.kintali@gmail.com Homepage: shivakintali.github.io

where cost is the negative of utility, a_{ij} is the cost when j is uncovered, and b_{ij} is the drop in the attacker's utility when j is covered. Let L denote the binary length of the explicitly listed rational prior and payoff table and the integer budget. The coefficients a_{ij}, b_{ij}, g_j have total bit length $O(L)$.

The expanded normal form has $\binom{m}{r}$ defender actions and m^k attacker contingent plans. Both counts can be superpolynomial in L. The best-response conditions have a compact description. Let p_{ij} be the conditional probability that type i attacks j, and write $q_{ij} = \pi_i p_{ij}$ for the joint probability of that type and target. Define the *pressure* at target j by $P_j = g_j \sum_i q_{ij}$. Relative to leaving all targets uncovered, a pure defense A changes the defender's expected payoff by $\sum_{j \in A} P_j$. Thus a best defense selects r targets with the largest pressures, with free choice at ties. A mixed defense is optimal exactly when each of its positive-probability pure defenses is optimal. The attacker can choose a best response separately for each positive-prior type. It follows that a profile is an equilibrium exactly when every positive-probability defense is optimal and

$$q_{ij} > 0 \quad \Longrightarrow \quad F_{ij}(x_j) = \min_h F_{ih}(x_h).$$

Every mixed defense has $x \in [0,1]^m$ with $\sum_j x_j = r$. Conversely, every such x is realized by the lottery construction in Appendix A. Computing a best response on either side therefore requires only row minima or a sort of the pressures. The problem is to find a single profile satisfying both conditions.

We count bit operations on an indexed store of polynomial size with binary addresses. The count includes arithmetic, memory access, and output. The notation $\widetilde{O}(\cdot)$ suppresses powers of $\log(2 + L)$. Each pure defense in the output is represented by its m-bit incidence vector.

Theorem 1.1 (Main result)**.** *For every input of the above form, the algorithms return an exact Bayes–Nash equilibrium with rational conditional attack probabilities and a rational lottery on at most m pure defenses. The conditional attack strategy for every type is supported on its cost-minimizing targets. Their indexed bit bounds are as follows.*

Algorithm and hypothesis	Bit operations
Deterministic, all inputs	$\widetilde{O}(L^4)$ worst case
Las Vegas, all inputs	$\widetilde{O}(L^{10/3})$ in expectation
Las Vegas, comparable type and target counts	$\widetilde{O}(L^3)$ in expectation

In the last row, comparability means that the ratio of the two counts is bounded above and below by fixed positive constants. The type count can be taken either before or after discarding zero-prior types.

Theorems 4.3 and 5.3 prove the two algorithmic bounds. The expectation is over internal randomness for each fixed input. Each accepted run verifies the exact best-response conditions. Each independent attempt succeeds with probability at least $1/2$, so the expected number of attempts is at most two. Appendix A proves that the strategy extraction fits within these bounds. If $r = 0$ or $r = m$, the defender has a unique pure defense, and independent scans of the attacker rows at its coverage vector give an equilibrium. A zero-prior type contributes no attack mass and can be deleted during the search. At the final coverage an original cost-minimizing target gives a conditional strategy for each deleted type.

For example, take one type, two targets, one resource, costs $F_{11}(x) = F_{12}(x) = x$, and gains $(g_1, g_2) = (1, 2)$. The defender selects each singleton with probability $1/2$. Both targets then have attack cost $1/2$, so the attacker can mix between them. Assigning attack probabilities $(2/3, 1/3)$ gives pressures $(1 \cdot 2/3,\ 2 \cdot 1/3) = (2/3, 2/3)$. Both singleton defenses are therefore best responses, as are both attacks.

For the remaining analysis k counts only positive-prior types unless otherwise specified. Set $e = km$, $n = k + m + 2$, and $d = \min(k, m)$. The input has at least one entry per type–target pair, so $e + n = O(L)$, $(d + 1)n = O(e + n)$, and $d = O(\sqrt{L})$.

1.1 Relation to prior work

Li, Conitzer, and Korzhyk's catcher–evader method covers the case of strictly positive coverage effects through a broader resource-allocation model. It increases the defender's budget while rerouting attack mass among minimizing targets to maintain equilibrium. Its proved upper bound on the number of iterations is exponential [21, Theorem 5]. Li's dissertation describes the method and its experiments [22, Chapter 3]. Conitzer's revised note cites a qualification of interchangeability for heterogeneous types. Interchangeability means that taking one player's strategy from either of two equilibria and the other player's strategy from the other equilibrium again gives an equilibrium. With strictly positive coverage effects and fixed defender gains, this property holds in the interior of the prior simplex outside a finite union of hyperplanes. Three-type, three-target examples show failure on that exceptional set [8, 5].

For complete information and strictly positive coverage effects, Korzhyk et al. relate the single-attack model to a zero-sum game by reweighting and renormalizing a single attacker distribution [19, Lemma 3.1]. With private types, each type has a separate mass constraint. Renormalizing the reweighted loads type by type can change their relative contributions to the defender's pressures. Korzhyk, Conitzer, and Parr also give a polynomial algorithm for the complete-information model with multiple attacker resources. Xu characterizes tractability via optimization over feasible defensive schedules in zero-sum and single-type variants. Graph-based and nonadditive models provide further settings where the payoff structure and the feasible schedule polytope govern the complexity [18, 30, 29, 4].

The Bayesian Stackelberg literature studies a defender who announces a mixed strategy and optimizes against the attacker's subsequent type-dependent responses. DOBSS uses a mixed-integer formulation for this objective, and the related ERASER formulations exploit coverage marginals [26, 16]. The complexity of commitment depends on the feasible schedules and type structure [17]. Li, Conitzer, and Korzhyk prove strong NP-hardness even with one attack per type [21, Theorem 2]. Conitzer and Sandholm study optimal commitment in finite games. Von Stengel and Zamir analyze leadership games with convex strategy sets [7, 28]. Li and Conitzer's work on strategic test design supplies another application of Bayesian resource allocation, including a tractable scored-test model [23].

On the market side, Baïou and Balinski introduce divisible stable allocation with fixed rankings, and Dean and Munshi give the fast explicit-instance algorithm we use for restricted markets. Cseh and Matuschke study the stable-flow formulation, while Karzanov develops rotations for a generalized allocation model. These results supply finite-market algorithms and structural tools. The search here must also select the contracts to present to such an algorithm, because the full grid can have exponentially many contracts [1, 11, 9, 15]. Buyer-specific prices also arise in matching with linear transfers, which Galichon and Jacquet connect to nonzero-sum hide-and-seek games. Balkanski, En, and Faenza give an algorithm for explicitly listed indivisible items with buyer-specific price functions. Replacing a binary-encoded rational mass by unit copies can require exponentially many items. Oki and Sato use discrete convex analysis to find valid directions in ascending auctions with heterogeneous payment functions. Our search bounds the number of restricted allocations as well as the bit cost of each one [13, 2, 24, 25].

The preliminary companion manuscript gave a conservative $O(L^{71})$ bit bound using nested bisection [12, Appendix B]. Theorem 1.1 improves that full-input bound and answers Conitzer's question for rational data, including zero coverage effects. The new analysis uses a common witness

for the pressure and price searches and a faster implementation of the allocation queries.

1.2 Proof strategy and contributions

We construct a market in which contracts specify affine attack costs and auxiliary purchases record coverage. As a common capacity parameter increases, total coverage is continuous, piecewise affine, and nonincreasing. A small dyadic shift of the resource budget puts its crossing in the interior of a strictly decreasing affine segment. The positive optional bids on that segment form a fixed witness of at most $k + m - 1$ contracts. They occupy at most two price levels at each vertex on the side chosen for posting. A trial solves a restricted market at a fixed parameter. A wrong comparison with the shifted budget, or a failed equilibrium residual test sufficiently near the crossing, implies that some witness bid blocks the allocation. A blocking bid is one whose two endpoints can both accept more of it under their rankings and capacities. This property allows the algorithm to learn from failed trials without constructing the full market.

The deterministic search offers all eligible contracts at each selected price. On a failed round it doubles weights in the scalar gap and in the price gaps marked by blockers. The small witness bounds the number of failed rounds. The randomized search samples groups of counterparties at a common price, updates their weights using blocking tests, and uses modular comparisons of forest-cut sums to accelerate allocation. The expected $10/3$ exponent comes from choosing between complete and sampled menus according to the smaller dimension. The final reconstruction enforces the original rational constraints, and exact verification checks that both players' returned strategies are best responses, including when perturbations are used in the auxiliary market.

Section 2 gives the equilibrium certificates and the auxiliary market. Section 3 proves the fixed-set property at the budget crossing. Sections 4 and 5 give the deterministic and Las Vegas algorithms and their bit bounds. Appendix A proves exact reconstruction, including the defender lottery. Appendix B gives the arithmetic implementation of restricted-market queries.

2 Equilibrium certificates and the auxiliary market

2.1 Support and coverage conditions

Assume $0 < r < m$ and discard zero-prior types as in Section 1. A *pattern* specifies a set E_0 of permitted attack pairs and a status (lower, balanced, or upper) for each target. It determines two separate linear systems. The first uses the coverages x_j and the row cost variables u_i. Its constraints are

$$0 \le x_j \le 1, \quad \sum_j x_j = r, \qquad u_i \le F_{ij}(x_j) \quad \text{for all } i, j, \qquad u_i = F_{ij}(x_j) \quad ((i,j) \in E_0). \tag{1}$$

It also fixes $x_j = 0$ at lower targets and $x_j = 1$ at upper targets. The second uses the joint masses q_{ij} and a threshold λ. Require $q \ge 0$, $q_{ij} = 0$ off E_0, $\sum_j q_{ij} = \pi_i$ for each type, and $0 \le \lambda \le \max_j g_j$. With $P_j = g_j \sum_i q_{ij}$, impose

$$P_j \le \lambda \quad \text{(lower)}, \qquad P_j = \lambda \quad \text{(balanced)}, \qquad P_j \ge \lambda \quad \text{(upper)}. \tag{2}$$

Feasible solutions of the two systems yield an equilibrium. Since each row has positive mass, at least one of its permitted pairs has positive q_{ij} and attains u_i. The row inequalities then make u_i its minimum cost and every positive attack a best response. The pressure constraints force $x_j = 1$ whenever $P_j > \lambda$ and $x_j = 0$ whenever $P_j < \lambda$. For any other feasible coverage y,

$$\sum_j P_j(y_j - x_j) = \sum_j (P_j - \lambda)(y_j - x_j) \le 0.$$

Every summand on the right is nonpositive by the endpoint conditions. Thus x is a best response. Conversely, take E_0 to be the positive attack pairs of an equilibrium, and use status lower at $x_j = 0$, upper at $x_j = 1$, and balanced elsewhere. Set u_i to the row minimum and λ to the rth largest pressure. Each optimal pure defense includes every target with pressure strictly above λ and excludes every target with pressure strictly below λ. Averaging over the equilibrium lottery gives (2). In a fixed pattern the constraints are closed, so exact reconstruction may give zero mass to a permitted pair or move a balanced target to either coverage endpoint.

Write each equality as two inequalities and clear denominators separately in each row to obtain $a^\top z \le b$, where z is the variable vector of the relevant system. For a common bound, form the catalog of all possible integer rows, including both orientations of possible equalities. Choose a power of two $H \ge 2$ that bounds the product of $\max(2, \|(a, b)\|_1)$ over the catalog and also bounds every denominator-clearing multiplier. A bound for the product is obtained by adding bounds on the binary logarithms of the individual factors. A cost row uses the rational coefficients of a single pair (i, j). A pressure row uses g_j and at most k masses and contributes $O(\mathrm{bits}(g_j) + \log(k + 1))$ to the sum. A mass row uses π_i and at most m entries and contributes $O(\mathrm{bits}(\pi_i) + \log(m + 1))$. Here bits denotes binary encoding length. The budget, status, box, threshold, and zero-mass rows together contribute $O(e + n + \log(m + 1) + \mathrm{bits}(\max_j g_j))$. The contributions sum to $O(L)$ because $m \log(k + 1) + k \log(m + 1) = O(km)$ and the rational data are explicit. The same estimate bounds the lengths of the multipliers since each uses each distinct input denominator in a row only once. If variables fixed to zero are deleted, use the same multiplier. This can only decrease the row norm and preserves the bound for subpatterns. Define

$$\delta = (8H^6)^{-1}, \qquad \xi = (64H^8)^{-1}.$$

In particular, rounding all coordinates to a common dyadic mesh of step at most ξ increases each cleared residual by at most $H\xi$. A previously unscaled violation of at most ξ contributes at most another $H\xi$ after clearing, and $2H\xi \le \delta$.

Theorem 2.1 (Exact reconstruction)**.** *Suppose a rational trial supplies coverages in $[0, 1]^m$, row minima, a nonnegative attack mass vector, and a pressure threshold, with $O(L)$ bits per coordinate. Use the positive attack pairs as E_0 and set the coverage statuses from the trial coverages. Suppose each inequality in the two original, unscaled systems has violation at most ξ, where an equality is tested in both directions. Then one can recover a feasible point of the pattern and output exact equilibrium strategies in $\widetilde{O}(L^3)$ deterministic bit operations.*

Appendix A proves the theorem and gives the explicit defender lottery. The search invokes reconstruction once, upon accepting an approximate certificate.

2.2 Stable allocation and physical coverage

A type is a *buyer* of mass π_i and a target is a *site*. A bid specifies a buyer, a site, a physical price in $[0, 1]$, and a quoted cost for the buyer. Buyers and sites rank bids by the rules below. The algorithms solve allocations on restrictions of a full market whose fine grids may have exponentially many entries.

Replace every zero slope by a common positive dyadic τ and every zero gain by a common positive dyadic τ_Q. Write $F'_{ij}(x) = a_{ij} + b'_{ij}x$ and g'_j for the perturbed costs and gains. On $[0, 1]$ the cost changes by at most τ and the gain changes by at most τ_Q. Let B_s and B_g be the least integral powers of two at least $\max(1, \max_{ij} b'_{ij})$ and $\max(1, \max_j g'_j)$, respectively. One valid choice of parameters is

$$\tau = \tau_Q = \eta = \xi/32, \qquad \rho = \eta/(4B_s), \qquad \sigma = \xi/(32B_g).$$

Each parameter is a reciprocal power of two and has $O(L)$ bits. These choices satisfy $B_s\rho \le \eta/4$, $2\eta + 2\tau < \xi/4$, and $\sigma \max_j g'_j + \tau_Q < \xi/4$. The parameter η sets the cost mesh, ρ the physical mesh, and σ the auxiliary buyer mass at each site.

Choose one side of the buyer–site graph for posting. We ordinarily use the smaller side, which has d vertices. The balanced-dimension bound uses the site side. Each vertex on the chosen side has a grid, called its *axis*. A grid level and an opposite vertex specify an optional bid as follows.

- On a site axis j, the levels are $z = 0, \rho, \dots, 1$. The bid for type i has physical price z and quoted cost $F'_{ij}(z)$.
- On a type axis i, use all multiples of η in a fixed interval with integer endpoints containing every $F'_{ij}(x)$ for $0 \le x \le 1$. The bid to site j at quoted cost w exists when $F'_{ij}(0) \le w \le F'_{ij}(1)$. Its physical price is
$$z = \rho \left\lfloor \frac{w - a_{ij}}{b'_{ij}\rho} \right\rfloor .$$
Thus $F'_{ij}(z) \le w < F'_{ij}(z) + B_s\rho$. Successive eligible levels increase the unrounded inverse by $\eta/b'_{ij} \ge 4\rho$, so their rounded prices strictly increase.

For each pair (i, j) include a cap bid with physical price 1 and quoted cost $F'_{ij}(1)$. Identify any optional copy of that bid with the cap. Type i can use an outside option of cost $W_i = \min_j F'_{ij}(1)$ for any amount of its mass. A real buyer prefers lower quoted cost. At equal cost it prefers real bids to outside, then smaller site indices, then smaller physical prices. A site prefers higher physical prices and breaks real-bid ties by increasing buyer index. At either endpoint these rules give a strict order on the distinct bids incident to it. Bids of cost greater than W_i can be deleted because the buyer always has a cheaper outside option. The caps at sites attaining W_i are always retained. Any cap that survives this deletion is permanent.

At each site insert one private bid in each elementary gap of width ρ. The site ranks it strictly between the gap's two endpoint prices. A separate private buyer of mass $\sigma\rho$ prefers that bid to its only alternative, an unlimited outside option. The private mass per site is σ.

At a parameter $t \ge 0$, site j has quota t/g'_j. A feasible allocation assigns each buyer all its mass among its bids and its outside option, and the total assigned to a site is at most its quota. A bid can receive more mass when its buyer has positive mass on a strictly worse option and its site either has spare quota or has positive mass on a strictly worse bid. A bid satisfying both tests blocks the allocation. An allocation is *stable* if none of its available bids blocks it.

A restriction keeps the permanent bids and all the private buyers and offers only a subset of the optional bids. For a restriction, partition each site's private bids into groups between consecutive retained prices, using 0 and 1 as boundaries. Within a group, the private bids have the same comparisons with all retained real bids. Replace the group by one private buyer with their combined mass and one bid ranked in that interval. In any stable allocation the occupied private bids at a site fill in decreasing rank order. If a lower private bid had positive mass and a higher one had unassigned mass at its outside option, the higher one would block. Thus the private allocation within a group is determined by its total. To expand a grouped allocation, serve the highest private bids in the group first, splitting at most one private buyer. Each real bid has the same blocking status in the expanded and grouped markets. A private buyer can block using a vacancy, a worse real allocation, or a worse private allocation in another group exactly when its grouped buyer has a block of the same kind. Expansion rules out blocks within each group. Consequently both representations give the same stable real allocations and private totals. When testing an omitted bid, use the expanded rankings, since its price may lie inside a group.

Let U_j be the total private mass purchased at site j. Define

$$x_j = 1 - U_j/\sigma. \tag{3}$$

The quantity x_j lies in $[0, 1]$. It need not be a physical grid point, since a private buyer can be partially served.

Lemma 2.2 (Original-game residuals)**.** *Take a stable allocation on the complete fine market. Redirect each type's outside balance to any site attaining W_i and aggregate its purchases by site to obtain joint attack masses q_{ij}. Then $\sum_j q_{ij} = \pi_i$, and every positive q_{ij} satisfies*

$$F_{ij}(x_j) - \min_h F_{ih}(x_h) \le 2\eta + 2\tau.$$

With $Q_j = \sum_i q_{ij}$ and $\lambda = \min(t, \max_h g_h)$,

$$x_j > 0 \implies g_j Q_j \ge \lambda - (\sigma \max_h g'_h + \tau_Q),$$
$$x_j < 1 \implies g_j Q_j \le \lambda + (\sigma \max_h g'_h + \tau_Q).$$

The redirection and pressure statements hold for any restriction that keeps the permanent caps.

Proof. At a physical grid price z, the total private mass ranked above a real bid is $\sigma(1 - z)$. If a real bid at z carries positive mass, all that higher private mass must be served by stability, so $x_j \le z$. If $z > x_j$, the site holds private mass of $U_j > \sigma(1 - z)$. Some of it ranks below z, so the site accepts an increase at z. Both conclusions hold with partially occupied private bids and for arbitrary restrictions, when comparisons use the expanded allocation.

If type i uses outside and a cap site attaining W_i has $x_j < 1$, that cap at price 1 is acceptable to the site. The buyer strictly prefers the cap to outside by the rule for tied costs, yielding a block. Thus every cap site to which mass could be redirected has $x_j = 1$. Redirection defines the original-game attack masses while the market allocation and the coverages stay fixed. It preserves the type's row mass, and the redirected amount has perturbed cost exactly W_i.

For the attack bound let c_i be the largest quoted cost with positive mass for type i, counting its outside option. Then $c_i \le W_i$. If $x_j = 1$, we have $c_i \le F'_{ij}(1)$. Otherwise, on a site axis the first grid price strictly above x_j is at most $x_j + \rho$. Its bid is acceptable to the site. If its cost is strictly below c_i, it is retained on the complete grid and blocks. Hence $c_i \le F'_{ij}(x_j) + B_s\rho$. On a type axis, suppose instead that $c_i > F'_{ij}(x_j) + 2\eta$. There is a cost grid point strictly between $F'_{ij}(x_j) + B_s\rho$ and c_i. It lies in the eligible cost interval, since $c_i \le W_i \le F'_{ij}(1)$ and $x_j \ge 0$. Its inverse exceeds $x_j + \rho$ because $b'_{ij} \le B_s$, so even after rounding its physical price exceeds x_j. This bid also blocks. In both orientations we get $c_i \le \min_j F'_{ij}(x_j) + 2\eta$. Every positive market purchase has perturbed cost at the actual coverage at most its quoted cost because $x_j \le z$ and $F'_{ij}(z)$ is at most the quote. A redirected purchase has cost $W_i = c_i$ because outside was used. Allowing τ on each of the two costs compared gives the original per-action bound.

If $x_j > 0$, some private mass at site j remains outside, and a vacancy would allow it to block. The site is full, so removing at most σ private mass leaves real load at least $t/g'_j - \sigma$. Redirection can only add to it. If $x_j < 1$, redirection adds nothing at j and its real load is at most t/g'_j. Since $Q_j \in [0, 1]$, replacing g'_j by g_j changes a pressure by at most τ_Q. Finally, if $t > \max_h g_h$, lowering the threshold to $\max_h g_h$ relaxes the lower bound and leaves the upper bound true since every pressure is at most that maximum. This proves both pressure inequalities, including the case $\max_h g_h = 0$. □

For such a full-market allocation take $u_i = \min_j F_{ij}(x_j)$ and set the pattern from (x, q). Lemma 2.2 bounds the cost-equality and pressure violations by less than $\xi/4$. The remaining certificate constraints hold exactly, apart from the budget. A rational trial of the required bit length with $|\sum_j x_j - r| \le \xi$ therefore meets Theorem 2.1. This is the connection between the budget search in Section 3 and exact equilibrium.

3 The bids that determine the budget crossing

A single set of at most $k + m - 1$ optional bids suffices to check the direction of a static budget comparison. These bids use at most two price levels per posted axis. If a restricted allocation gives a wrong comparison, a bid from this set blocks it. The same conclusion holds for a failed residual test sufficiently close to the crossing. The algorithms use these conclusions without having to compute the set.

3.1 The capacity sweep

Lemma 3.1 (Continuous coverage)**.** *On every finite restriction, including the full fine market, there is a stable sweep on* $[0, T_0]$*, affine between finitely many events. Here* T_0 *is the least integral power of two strictly greater than* $\max\{1, (1 + m\sigma) \max_j g'_j\}$*. Each private total* $U_j(t)$ *is continuous and nondecreasing, even at parameters where real flows change instantaneously. The total coverage decreases from* m *at zero to zero at* T_0*, with Lipschitz constant at most* $\sigma^{-1} \sum_j 1/g'_j$*.*

Proof. At zero, assign each buyer its entire mass outside. At any current allocation, point each site to its highest-ranked bid whose buyer strictly prefers it to that buyer's worst occupied option. This is the site's proposal. Point each buyer to its worst occupied option, called its displacement. A site with no proposal and a buyer whose displacement is outside both point to a terminal. Proposals and displacements identify individual contracts.

A movement increases quantities along proposals and decreases them along displacements. A buyer receives only an option better than the one it releases, so its worst occupied rank can only improve. Once a bid becomes unacceptable to its buyer, its site can skip it permanently. A site acquires flow only at its current proposal, and proposals move down the site's list. Thus all held bids rank at least as high as the current proposal. A full site could block only with a bid better than some held bid. Such a bid ranks above the current proposal, so its buyer has already rejected it. A site with unused quota has exhausted its list and will never have a proposal again. These invariants guarantee stability.

If the pointers form a directed cycle, augment at fixed t. Increase each proposal quantity and decrease each displacement quantity by the smallest displaced quantity on the cycle. An increasing contract differs from its buyer's decreasing contract by the strict buyer preference, including in a cycle of two parallel contracts. The movement preserves each buyer's mass and each site's load. At least one worst option is exhausted. Recompute the pointers and repeat if there is another cycle. The stability invariants continue to hold. A private buyer can only acquire its bid by releasing outside mass, so a directed cycle cannot involve it.

When the pointer graph is acyclic, send an injection of rate $1/g'_j$ into site j. Process vertices in topological order. A site adds its injection to its incoming rate and sends the total to its proposal, or to the terminal as unused quota. A buyer passes all incoming rate along its displacement, releasing outside mass if that displacement is outside. At each buyer the addition and subtraction rates agree. At each site the change in load plus the change in unused quota equals the injection. In particular a site with a proposal stays full. This gives an affine allocation up to the first zero of a decreasing worst quantity, or to T_0 if no such zero occurs before then. On the open interval all worst options are unchanged. New positive options rank strictly above the displacement at the buyer, and the proposals remain the same. At the ending event remove all exhausted options at once, then recompute and eliminate directed cycles before advancing again. Each exhausted option is permanently unavailable for acquisition by that buyer since all remaining options are strictly better. The market is finite, so only finitely many exhaustions and affine stages occur. One may take the post-augmentation allocation at every event scalar.

Private site purchases never decrease. Along an acyclic stage, a unit of rate bought by a private buyer ends at the terminal because its displaced option is outside. Thus it is counted at most once in $\sum_j U_j$ and the slope of this sum is between zero and $\sum_j 1/g'_j$. All private totals agree before and after each fixed-t augmentation. Equation (3) proves the continuity, monotonicity and Lipschitz assertions. At T_0 each site's quota exceeds the total mass $1 + m\sigma$. Any private buyer with outside mass would then block at its site. Hence $U_j = \sigma$ for every j at T_0. □

Lemma 3.2 (Generic scalars and bit lengths)**.** *There is a finite set of exceptional scalars and a computable $B_0 = O(L)$, both valid for all restrictions. Each exceptional scalar has rational height at most 2^{B_0}. Outside that set the stable allocation is unique, and its positive bids between the k type buyers and the m sites number at most $k + m - 1$. At a rational nonexceptional scalar of $O(L)$ bits, the type-bid quantities, the site private totals and the coverages have $O(L)$ bits. Every endpoint of a sweep stage on $[0, T_0]$ and the budget at any such endpoint or exceptional scalar in $[0, T_0]$ have height at most 2^{B_0}. The height of a reduced fraction is the maximum of the absolute numerator and the positive denominator. Call a scalar outside the exceptional set* generic.

Proof. Use the unaggregated private bids. At $t = 0$ the stable allocation is unique because every buyer is entirely outside. For $t > 0$, take the usual balancing construction of Dean and Munshi [11]. With total buyer mass $P = 1 + m\sigma$ and total site quota $A = t \sum_j 1/g'_j$, introduce a dummy site of quota P for outside balances and a dummy buyer of mass A to fill unused real capacity. Each nondummy endpoint ranks its dummy contract last. Conservation forces the dummy-to-dummy amount to equal the total mass assigned on nondummy contracts. This amount is positive. Otherwise every buyer would be outside, every real site would have vacancy, and a private bid would block. The dummy buyer is last at each full real site and cannot block there. An ordinary buyer ranks outside below every retained bid. The dummy-to-dummy contract is last at both dummies. Thus blocking by nondummy bids is unaffected and no dummy bid can block, giving a bijection of stable allocations.

Use the job-proposing allocation of Dean and Munshi and its proposal and refusal graph [11, Lemmas 4, 5 and 8]. In that graph each buyer points to its best contract acceptable to the receiving site, with the dummy site always acceptable. Each nondummy site points to its worst positive contract. The pointers in the cited arguments use strict ranks of incident contracts, and every update and balance sum is over those contracts. Distinct parallel bids can therefore be treated as distinct records in the two rank lists. A proposal to an ordinary site strictly outranks its refusal and cannot use the same record. A cycle of length two then consists of two different records and has the same alternating augmentation as a longer cycle. The graph arguments and the rotation criterion apply with edges counted with multiplicity.

Give each record an upper bound larger than every feasible amount, so all positive records are strictly below their individual bounds. Every such record occurs in the pointer graph of the job-proposing allocation. A record is acquired when it is the buyer's proposal. If it stops being a proposal while still positive, it has become the site's refusal and remains that refusal until exhausted. A positive dummy vacancy record is a refusal. A buyer with positive outside mass has no acceptable real proposal by stability, so its outside record is its proposal. The dummy-to-dummy record is the dummy buyer's proposal. Every vertex has one outgoing pointer except the dummy site. Hence the component containing the dummy site is a tree and every other component contains a directed cycle. By the rotation criterion, if the graph has no cycle, the job-optimal allocation is also site-optimal, so it is the unique stable allocation.

Any nonroot component excludes both dummies, which are linked by their positive contract. No positive record crosses its boundary. Its buyer masses and site quotas must therefore balance. With I the set of its type buyers, J its nonempty set of sites and h the number of its private buyers,

this means

$$t \sum_{j \in J} 1/g'_j = \sum_{i \in I} \pi_i + h\sigma\rho, \qquad 0 \le h \le m/\rho.$$

Use as the exceptional set all ratios defined by this equation for such choices of I, J, h, together with zero. The set is finite and independent of the retained bids. The algorithm uses its height bound in choosing the scalar grid.

At any generic scalar, form the undirected graph of positive nondummy assignments. A small alternating perturbation around a cycle preserves all balances, positive records, vacancies, and outside balances. All blocking tests are unchanged, so uniqueness rules out a cycle. The same argument along a path rules out two free balances in one component, where a free balance is positive outside mass at a buyer or positive vacancy at a site. For this path perturbation adjust the two endpoint balances by the amounts needed to absorb the changes of their incident path edges. Take the step small enough that both balances and all perturbed positive edges stay positive. Consequently the graph is a forest with at most one free balance per tree. Root each tree at that balance if present, and arbitrarily otherwise. The quantity on an edge is the signed difference between the masses and quotas in the component of its deletion away from the root. This follows by summing the balance equations there, where no free balance appears. A free balance is obtained by summing the entire tree. Dropping private vertices leaves at most $k + m - 1$ positive type-to-site records.

Let D_* be the product of the prior denominators, the numerators of the original positive gains in lowest terms, and the reciprocal of the elementary private mass $\sigma\rho$. This integer clears all buyer masses and the coefficients $1/g'_j$ of the quotas. For a replaced gain the reciprocal is an integer. Let M_* be a power of two strictly greater than

$$4D_* \left(1 + m + m\sigma + \sigma^{-1} + T_0 + \sum_j 1/g'_j \right).$$

The logarithms of D_* and M_* are $O(L)$, and the sum can be computed on scale D_*. Each ratio defining an exceptional scalar has height at most M_*. Between events the positive records and positive free balances are fixed on the open stage. Between exceptional scalars their cut formulas have the form $(a + bt)/D_*$, with $|a|, |b| \le M_*$. Each coefficient uses each primitive margin or rate in its cut at most once. At a site, full private bids form a prefix in decreasing rank, with at most one partially served bid by stability. Thus U_j is a constant full-prefix mass plus at most one cut quantity. The formula for each coverage and for its sum uses scale D_* since σ^{-1} is an integer. In particular, on every open stage after subdividing at the exceptional scalars,

$$S_R(t) = \sum_j x_j(t) = \frac{A_0 + A_1 t}{D_*}, \qquad |A_0|, |A_1| \le M_*^4.$$

The magnitude bound follows by summing at most m prefix and cut formulas, and using $m, \sigma^{-1} < M_*$. These formulas also give $O(L)$-bit coordinates at a generic scalar of $O(L)$ bits, including the q obtained by aggregating and redirecting ordinary flow.

Every event ending a positive-length stage strictly before T_0 is the zero of a decreasing cut quantity on the open interval immediately preceding it, or an exceptional scalar introduced by the subdivision. The finitely many fixed-t augmentations preserve the coverages. The boundary scalars $0, T_0$ also have height at most M_*. Evaluating the displayed budget formula at a scalar of height at most M_* gives height at most $2M_*^5$. At an event use the one-sided formula by continuity. Thus $B_0 = 6 \log_2 M_* + 4$ bounds all the stated heights. □

Lemma 3.3 (Occupied price levels)**.** *In a stable allocation on the complete market, the positive optional bids on an axis use at most two grid levels. If both levels occur, they are consecutive.*

Proof. Suppose two used prices on a site axis have a grid price strictly between them. At the intermediate price, the bid of the buyer holding the higher quote costs strictly less and is retained because the higher quote costs at most its outside option. The site would prefer it to the held lower bid, so it blocks. On a type axis, use the site of the lower of two occupied costs. An intermediate cost is eligible there. It exceeds the lower cost and is below the higher one, which is at most the buyer's outside cost and hence at most its cap cost at the chosen site. Its rounded physical price is strictly higher than that of the lower purchase, and the buyer strictly prefers the intermediate cost to the higher purchase. This gives a block. The arguments use strict numerical improvement on both sides and include the case of parallel bids. □

3.2 A fixed witness for static comparisons

Use the market of Section 2, including the permanent caps, outside options, and all private demand. Write $\mathcal{H}$ for the finite set of optional bids when every fine level is available. A restriction $R \subseteq \mathcal{H}$ retains the permanent contracts. On testing stability or identifying an allocation between two restrictions, expand the private demand in its fixed decreasing-price order. Let $S_R(t) = \sum_j x_j(t)$ be the sweep coverage for R at parameter t. At a generic scalar, the sweep allocation is the unique static stable allocation of that restriction. Only this static allocation is needed at a query.

Set $r' = r + 2^{-K}$, where the integer $K > B_0 + 2$ is large enough that $2^{-K} < \xi/4$. The full market's coverage decreases continuously from m to zero. By Lemma 3.2, the denominator of r' exceeds that of every endpoint budget and every plateau value. The shifted level lies strictly between 0 and m. After subdividing at the exceptional scalars, its crossing is interior to a strictly decreasing stage and is unique by monotonicity. Denote its scalar by t_*. Write the budget there as $(A_0 + A_1 t)/D_*$ as in the lemma. Then the fraction

$$t_* = \frac{D_*(2^K r + 1) - 2^K A_0}{2^K A_1}$$

has height at most 2^{B_*} for $B_* = K + 2B_0 + 4$. If two distinct rationals have denominators at most 2^{B_*}, their distance is at least 2^{-2B_*}. Choose a reciprocal power of two γ so that $2\gamma < 2^{-2B_*}$ and $2\gamma\sigma^{-1}\sum_j 1/g'_j < \xi/4$. These inequalities require $O(L)$ bits. It follows that $[t_* - 2\gamma, t_* + 2\gamma]$ lies inside the selected open stage and that $|S_{\mathcal{H}}(t) - r| < \xi/2$ there. The other original residuals obey Lemma 2.2. Let C_* be the positive optional bids on this stage.

Lemma 3.2 gives $|C_*| \le k + m - 1$, and Lemma 3.3 gives at most two fine levels per posted axis among its bids. Both bounds apply to the individual bids on the fixed crossing stage.

Lemma 3.4 (A fixed set of bids for static queries)**.** *Suppose t is generic and no bid of C_* blocks the static stable allocation on R. Then*

$$S_R(t) > r' \Longrightarrow t < t_*, \qquad S_R(t) < r' \Longrightarrow t > t_*, \qquad S_R(t) = r' \Longrightarrow t = t_*.$$

For $|t - t_*| \le \gamma$, that allocation is the full allocation at t.

Proof. Adding C_* leaves the allocation at t feasible and stable. By uniqueness its total there is $S_{R\cup C_*}(t)$. On the crossing stage the full allocation uses only permanent bids and members of C_*. Restricting it to $R \cup C_*$ preserves its stability. Throughout the open stage the scalar is generic, so the two allocations agree there. The union's total coverage thus crosses r' strictly at t_*. Global monotonicity and strict change on both sides of the crossing imply all three signs. Within the specified interval the agreement of allocations follows directly from uniqueness. □

Use zero and T_0 as scalar endpoints, with labels high and low, respectively. Insert the odd multiples of 2^{-s} strictly between them, where $s > B_*$ and $2^{1-s} < \gamma/10$. Every interior point has reduced denominator 2^s. It is generic for all restrictions and differs from t_*. Its stable allocation has the bit length required for reconstruction by Lemma 3.2. Let $B = O(L)$, enlarged to cover the logarithmic sizes of all price grids and this scalar mesh, all separation exponents above, and $\log(2 + L)$.

3.3 Scalar queries and rounding

At an interior scalar, compute the restriction's stable allocation and recover x by (3). Redirect each outside balance to one of that type's least capped-cost sites to form q. The permanent cap argument in Lemma 2.2 applies also to a restriction, since it uses only a permanent cap and the private buyers. Set $u_i = \min_j F_{ij}(x_j)$ and $\lambda = \min(t, \max_j g_j)$. Set the permitted pairs to the positive entries of q, and determine each status from whether x_j is zero, interior, or one. Test the pattern unscaled residuals against ξ, including $|\sum_j x_j - r|$. If the test passes, reconstruct and verify as in Theorem 2.1. Otherwise assign a high label when $S_R(t) > r'$ and a low label when $S_R(t) < r'$. Either label is allowed at equality. By Lemma 3.4, an incorrect sign away from t_* or a failed residual test within γ of t_* implies a blocker in C_*. Compute blockers for the market allocation *before* redirecting its outside purchases.

Maintain one scalar menu and one price menu on each posted axis. They contain both endpoints of their grid. A candidate at a target price rounds down to the preceding or equal menu price. A candidate at a type cost rounds up to the following or equal menu cost. The rounded contract is offered if eligible, identifying a cap duplicate with its permanent bid. The candidate universe may contain ineligible pairs. Eligibility is tested at the rounded quote, so all fine levels that round to the same quote give identical offered bids for each fixed opposite vertex.

Lemma 3.5 (Rounding and blocking)**.** *Fix a static trial. At each posted axis at most two current open menu gaps can be marked so that every blocking fine bid either rounds to a blocking bid or has its fine index inside a marked gap.*

Proof. First consider a site axis. For each buyer label, the prices at which the site accepts form a suffix of its physical grid. A vacancy gives the entire grid. In the absence of a vacancy, acceptance via private mass is equivalent to $z > x_j$. Its first index is $\lfloor x_j/\rho \rfloor + 1$, with the index past the grid used for an empty suffix. For acceptance via real mass, use the worst occupied real bid. The threshold index is its price index or the next index, according to whether the candidate buyer wins the priority tie. With no real mass this suffix is empty. Taking the union of the private and real suffixes leaves at most two possible threshold indices across all buyer labels. This covers a zero-quota site, where both suffixes are empty.

Rounding down gives the buyer a strictly cheaper offer if the price moves, and leaves the bid unchanged otherwise. In particular it preserves eligibility of a blocking bid. If it destroys site acceptance, the rounded index lies strictly below that label's threshold while the original index is at or above it. The threshold lies in the same open gap as the original index, since its left endpoint is the rounded index. Mark the open gap, if any, containing each possible threshold.

Now consider a type axis. Let c be the largest quoted cost used by the buyer, including outside if used. Ignoring the per-site eligibility interval, the buyer accepts at every grid cost below c and rejects every cost above c. At equality it uses the strict rank test against the worst occupied option. Thus the last acceptable cost index either is the greatest index at cost at most c, or is one less when the cost equals c and the strict test rejects. Use the index before the grid for an empty prefix. There are at most two such boundary indices across site labels.

If rounding up moves to an eligible cost, its physical price strictly increases at the given site, so site acceptance is preserved. Failure of eligibility after increasing the cost can only mean exceeding the cap cost or the outside cost. Either makes the rounded quote strictly greater than c. Thus any loss of blocking crosses the corresponding buyer threshold. The original index is at or below that threshold and the right menu endpoint is strictly above it. Mark the open gap, if any, containing each possible last acceptable index. This proves the claim on both orientations. A threshold that is itself on the menu requires no marked gap because it cannot be crossed by this rounding. Given the worst occupied ranks and private totals, floors and strict tie tests give the one or two candidate indices on each axis. Binary search in its menu locates the open gaps containing them. □

3.4 A search round

During one round freeze the menus and all sampling weights. Starting with the opposite-labeled scalar endpoints, binary search their ordered current menu for an adjacent high–low pair. Each query replaces the endpoint with the same label as the middle index. The number of remaining index intervals is at most half the previous number rounded up. This uses only the endpoint labels and also applies when different restrictions give nonmonotone labels. Save the allocation from each retained interior endpoint. There are $O(1 + \log J)$ trials when menu sizes are $O(J)$. On a failed round double the ordinary unit-initialized weights on the scalar fine indices strictly between the bracket endpoints. On each posted axis double the union of the open gaps marked by Lemma 3.5 at those interior endpoints. These are at most four gaps per axis. Neither scalar endpoint needs a trial.

Fix one interior scalar mesh index within $\gamma/4$ of t_*. If the retained bracket contains t_* with both endpoints farther than γ from it, this fixed index gains a doubling. If it contains t_* with an endpoint in the band, that endpoint failed its residual test. If the bracket excludes t_*, at least one endpoint has the wrong sign. The two endpoints of the scalar grid have correct labels and lie outside the band. In the last two cases a member of C_* blocks at an interior endpoint. Either its price is doubled as an ordinary fine level, or its rounded bid blocks. Thus a failed round can always be charged to the fixed scalar index, one of two fixed price indices per posted axis, or a rounded blocker belonging to one of the fixed price groups in C_*. This gives at most $2d$ price indices when posting on the smaller side and at most $2m$ when posting on the site side. The search does not need to know the witness indices or bids.

4 The deterministic algorithm

Post on the smaller side and set $J = 64(d + 1)$. A *complete menu* offers every eligible bid at each of its listed quotes. The restricted market also keeps the permanent bids and all private demand of Section 2.

1. Set every scalar and price grid weight to one. Initialize each menu by the weighted quantiles in Lemma 4.1.

2. Freeze the menus for one round of Section 3.4. At every queried interior scalar, solve the restricted stable-allocation problem and test the original-game residuals. Upon passing the test, reconstruct by Theorem 2.1 and return the verified equilibrium.

3. If the round ends with an adjacent high–low pair, double the weights strictly inside the scalar bracket and in the union of the marked open price gaps on each axis. Repair the menus by Lemma 4.1 and start the next round.

Lemma 4.1 (Maintaining small menus)**.** *Consider a nonempty axis with at most $2^{O(B)}$ indices, initialized with unit weights. Suppose an update doubles the weights in at most c distinct current open menu gaps, where $c \geq 1$ is an absolute constant. For every integer $J \geq 2$, one can maintain $O_c(J)$ menu points, including both endpoints, with each open gap of weight at most W/J, where W is the current total weight. Between rebuilds the only changes are splits. Each completed interval between rebuilds has at least J/c updates, and over T updates there are $O_c(J+T)$ point insertions and removals in total, including initialization. The maintenance takes $\widetilde{O}((J+T)(B+T/J))$ bit operations when J, T are polynomial in L.*

Proof. At initialization or a rebuild, let W_0 be the current total and set $M = W_0/J$. Include the endpoints and the first index whose inclusive prefix weight is at least hM for each integer $1 \leq h < J$. Retain each selected point once. If an open gap had weight greater than M, the prefix weight at its left endpoint and the prefix weight just before its right endpoint would differ by more than M. Some threshold hM would lie strictly above the former and at or below the latter. Its first attaining index would lie in the gap, a contradiction. This argument allows several thresholds to select the same index. An axis consisting of one point has no open gaps.

Keep M fixed until $W \geq 2W_0$. In an update, double all selected gaps before changing the menu. If the total reaches $2W_0$, rebuild at the new total, retaining all grid weights. Otherwise repair each doubled gap whose new weight exceeds M. Its new weight is at most $2M$, so inserting the first point to reach half the gap's new weight leaves at most M weight strictly on either side. An insertion is charged to the increase of weight in its gap, which exceeds $M/2$ because the old weight exceeded $M/2$. These increases are disjoint within an update. Each update adds at most cM, and the total increase up to and including the update that triggers a rebuild is at most $W_0 + cM$. Thus at most $O_c(J)$ points are inserted by splits in any interval. The menu starts each interval with at most $J+1$ points, so the final rebuild also changes at most $O_c(J)$ points. An increase of at least W_0 takes at least $W_0/(cM) = J/c$ updates. Adding the initial and any unfinished interval proves the lifetime bound. Throughout an interval the open gaps have weight at most $M \leq W/J$.

For the implementation, store the fine indices as contiguous runs of constant weight in a balanced sequence tree. A leaf stores its run length and weight. Store subtree lengths and total weights, with a lazy doubling tag for a subtree. Splitting at the ends of an updated interval creates at most two new runs. Splitting and rejoining the tree allows one lazy doubling on the selected interval. The number of runs is $O_c(1+T)$. Prefix weights and weighted ranks are found by descending the tree using the subtree sums. If a rank falls inside a run, an integer ceiling division gives the first index attaining it. Gap and rank tests use integer comparisons after clearing J or 2, as appropriate. Keep the ordered menu separately so that a gap containing a given index can be found by predecessor and successor searches.

The total before each update bounds the gap weights by W/J, so over T updates

$$W \leq W_{\text{initial}}(1 + c/J)^T.$$

Since W_{initial} is the number of fine indices, the run weights and subtree sums have $O_c(B + T/J)$ bits. Run lengths and grid indices have $O(B)$ bits. All tree operations take logarithmically many arithmetic and comparison operations on these integers, with $O(\log J)$ additional bits for the quantile comparisons. Inspecting the doubled gaps uses $O_c(T)$ operations on the trees. Quantile selections at initialization and at rebuilds, together with the repairs, use $O_c(J+T)$ such operations. This proves the maintenance bound. □

Lemma 4.2 (Complete menus)**.** *With complete menus on the smaller side and $J = 64(d+1)$, there are at most $O((d+1)B)$ failed rounds. Each static query has $O((e+n)(d+1))$ contracts after aggregating private demand.*

Proof. Section 3.4 gives a fixed set of at most $2d+1$ witness indices. It includes one scalar index near the budget crossing and at most two price indices per posted axis. The bracket argument there applies to both labels at equality and to brackets that retain a grid endpoint. Every eligible rounded bid is offered, with cap duplicates identified with their permanent bids. Therefore a rounded bid cannot block the restricted stable allocation. A failed round must double at least one of the witness weights.

Let β be the maximum of one and the base-two logarithms of the initial total weights of the axes, so $\beta = O(B)$. The sum of the base-two logarithms of the witness weights is initially zero and is at least T after T failures. On each axis at most four disjoint open gaps are doubled per failure, counting the union of the marked gaps at the two bracket endpoints. The scalar axis uses at most one gap. Applying the total weight bound in Lemma 4.1 with $c = 4$ gives

$$T \le (2d+1)\beta + (2d+1)T \log_2(1 + 4/J).$$

The coefficient of T on the right is at most $4(2d+1)/(J \ln 2) \le 1/(8 \ln 2) < 1/2$. Thus $T \le 2(2d+1)\beta$. Every round either passes the residual test and returns an equilibrium or counts as a failure, so this also bounds the total number of rounds up to one additional successful round.

Each of the e type–site pairs gives at most $O(J)$ optional records and one cap record. The number of private groups at a site is at most one plus the number of retained real bids there. Each group contributes one private bid and one outside option. Including the outside contracts of ordinary buyers and the vacancy arcs of the sites gives $O(eJ + e + n) = O((e+n)(d+1))$ contracts. □

Theorem 4.3 (Deterministic bit bound)**.** *With complete menus an exact equilibrium is computable deterministically in*

$$\widetilde{O}((e+n)(d+1)^2L^2 + L^3)$$

indexed bit operations. In particular the worst-case bound is $\widetilde{O}(L^4)$.

Proof. Use the direct solutions of Section 1 for the extreme budgets and discard the zero-prior types during the search. For an interior budget, the one-time setup of the grids and height bounds, the permanent bids, and the common denominator in Lemma 3.2 costs $\widetilde{O}(L^2)$. Indeed the formulas in Sections 2 and 3 use $O(e+n)$ operations on $O(L)$-bit integers and rationals. Products and sums for the common scale can be formed with $O(L)$ bits throughout.

Let N denote the number of records of a query, including the balancing arcs. Enumerate the $O(eJ + e + n)$ candidate records from the current menus, test eligibility, and compute the retained physical prices and costs exactly. Sort the strict ranks, then form the private groups by subtracting consecutive physical prices at each site and multiplying by σ. This costs $\widetilde{O}((e+n)JL)$. At a queried scalar with reduced denominator 2^s, the common scale D_*2^s from Lemma 3.2 makes the quotas and all buyer masses integral. In particular each private group comprises an integer number of elementary masses $\sigma\rho$. The scaled sum of the margins, including the dummy margins, has $O(L)$ bits.

Run the bipartite algorithm of Dean and Munshi [11, Section 3] with the balancing construction and the strict ranks on individual contracts described in Lemma 3.2. Its worst-case $O(N \log(2+N))$ operation bound counts rank advances and dynamic-tree path and cycle augmentations. Rank advances count parallel bids individually. On a two-record cycle, the same augmentation rule applies to the two distinct residual amounts. Thus the operation bound counts records with multiplicity. On the integral scale, each transferred quantity is a minimum of residual capacities and, on a path,

the available mass of the starting buyer. Give each record a redundant upper bound of one plus the sum of the scaled margins. Every flow, residual, and individual path update is bounded by this integer. The dynamic trees use minima and uniform additions on paths. Their stored weight differences and accumulated updates also have $O(L)$ bits, since the number of bounded updates is polynomial in L. The solve takes $\widetilde{O}(NL)$ bit operations and returns the unique restricted allocation at the generic scalar.

Scan its records to recover the worst occupied ranks, private totals and vacancy flags. Compute x by (3) and form q by redirecting outside mass as in Section 3. All the load sums use the common flow scale, and $1/\sigma$ is integral, so the coverages and their sum also have a common scale of $O(L)$ bits. Computing row minima uses pairwise exact cost comparisons. The pressure, status, and budget tests can then be evaluated with $O(L)$-bit operands in $\widetilde{O}((e+n)L)$ work after the record scan. At a retained interior bracket endpoint, use the saved allocation data before redirection to find the rounding gaps by Lemma 3.5. Its floor, rank and menu searches fit the same bound.

By Lemma 4.2, writing T for the number of failed rounds, we have $T = O((d+1)L)$ and $N = O((e+n)(d+1))$. There are $O(1+\log J)$ queries per round. On the $d+1$ axes together, the maintenance in Lemma 4.1 costs $\widetilde{O}((d+1)(J+T)L)$, since $B = O(L)$ and $T/J = O(L)$. This is covered by the query bound. Adding preprocessing and the single exact reconstruction, including verification and strategy extraction, gives

$$\widetilde{O}((T+1)(e+n)(d+1)L + L^3) = \widetilde{O}((e+n)(d+1)^2L^2 + L^3).$$

The final worst-case bound follows from $e+n = O(L)$ and $(d+1)^2 = O(L)$. □

5 The Las Vegas algorithm

For $d+1 \le L^{1/3}$, use the complete menus of Section 4 with the faster static queries in Lemma 5.2. Otherwise use the grouped sampler below on the smaller side. For the bound with comparable dimensions, use the grouped sampler on the site side.

1. Start an attempt with unit ordinary grid weights and fresh independent random bits for the sampler and the modular comparisons. Initialize the menus as in Section 4, with the size parameter of the chosen branch. In the sampled branch initialize the pools of Section 5.1.

2. Perform a round of Section 3.4. At each interior scalar use either all eligible bids at the current quotes or the bids from a fresh sample of tuples, according to the branch. In both cases retain the permanent bids and private demand. Solve the restricted allocation and test its residuals.

3. At the first passing screen, reconstruct the profile and verify it against the original game. Return the profile if it passes verification. Restart if verification fails, an intermediate state is invalid, or the attempt reaches its work or round cap.

4. After a failed round, update the tuple weights using the retained interior trials in the sampled branch. Then double the ordinary weights and repair the menus as in Section 4. Start the next round within the attempt limits.

5.1 Tuple distributions and updates

Write h for the number of posted axes and a for the number of opposite labels per axis. Thus $ha = e$, and $h = d$ when posting on the smaller side. Group the fixed witness bids C_* of Section 3 by axis and fine price. Represent a group of size s_0 by an ordered tuple of its opposite labels,

padded to the next power of two by repeating one label. There are $\ell = 1 + \lceil \log_2 a \rceil$ possible lengths $b = 1, 2, 4, \ldots, 2^{\ell-1}$. If H_b denotes the number of witness tuples of length b, Lemmas 3.2 and 3.3 give

$$\sum_b H_b \le 2h, \qquad \sum_b bH_b \le 2n, \qquad H_b \le K_b := \min\{2h, \lfloor 2n/b \rfloor\}.$$

Each K_b is positive. The algorithm uses K_b and does not require the witness groups.

A pool contains items $(j, s, i_1, \ldots, i_b)$, with axis j, fine index s, and b opposite labels in order. Give an item weight

$$M_{jb}(s) \prod_{v=1}^{b} p_{i_v}, \qquad p_i = u_i / \sum_{i'} u_{i'}.$$

The u_i belong to the current rounding cell of s on axis j in this pool. A cell consists of all indices rounding to the same menu quote. On a site axis the cell extends from its quote up to but excluding the next quote, with the maximum index a singleton. On a type axis it extends from just after the previous quote through its own quote, with the minimum index a singleton. At initialization set all marginals $M_{jb}(s)$ and label weights u_i to one. To sample an item, choose an axis and cell in proportion to their summed marginals, then draw b independent labels from its p_i. Only the cell and labels are needed to offer the eligible rounded bids.

For each trial draw independently with replacement in each pool. Fix its menus for the trial. For any allocation, a rounded bid blocks precisely when its buyer and site both accept it. Each acceptance test is given by a cutoff in a strict order, with site vacancy allowing the entire order. Eligibility and identification with permanent bids are fixed. Hence on q tested bid positions the family of possible blocker sets has at most $(q+2)^{k+m}$ traces. This counts ties by their strict ranks. On u tuple positions, take the disjunction of the b entry tests, giving at most $(bu+2)^{k+m}$ traces. Repeated bid or tuple positions do not increase these bounds.

Set $\eta_b = (C_1 \ell K_b)^{-1}$ with $C_1 = 256$. We use the following finite-sample form of the ε-net argument [14]. If u independent draws miss a range of mass greater than η_b, select one such range by a fixed order. A second independent sample of size u hits it in at least $u\eta_b/2$ positions with probability at least $1/2$ when $u\eta_b \ge 8$, by the binomial lower-tail bound. Equivalently generate $2u$ draws and uniformly split their positions into the two samples. Conditional on these draws, the probability that a specified trace of size v lies entirely in the second sample is at most 2^{-v}. A union bound over traces shows that the probability of the first sample missing any range of mass greater than η_b is at most

$$2(2bu+2)^{k+m} 2^{-u\eta_b/2}.$$

With $u = C(n+1)\eta_b^{-1}\lceil \log_2(2+L) \rceil$ for a sufficiently large positive integer C, this bound is an arbitrarily small inverse power of $2+L$. Indeed b, K_b, ℓ, n are polynomial in $2+L$, so the logarithm of the trace bound is $O(n \log(2+L))$. The constant can be chosen for a prescribed polynomial cap on the number of queries.

This event applies simultaneously to all cutoffs, including those of the stable allocation computed from the sample itself. Every sampled eligible bid is present in that restriction. Stability makes the resulting blocker range disjoint from the sampled items, so its mass in pool b is at most η_b on the net event. The same estimate holds conditional on the previous transcript at each fresh draw. The number of sampled bid entries over all pools is $\widetilde{O}(n^2)$, since the number of sampled tuples in pool b is $\widetilde{O}(nK_b)$ and $bK_b \le 2n$. Permanent contracts and aggregated private groups fit the same bound since each retained bid introduces at most one additional private group and $e = O(n^2)$.

On a failed round, take the union of the rounded blocker sets of the retained interior trials. They use the same menus and pool weights, so the union has mass at most $2\eta_b$ in pool b when both trials satisfy the net event. For one cell let D be its labels in the union and $\beta = \sum_{i \in D} p_i$. A tuple in the cell meets D with probability $p = 1 - (1 - \beta)^b$. The update is as follows.

- If $b\beta \geq 1$, double $M_{jb}(s)$ throughout the cell.
- If $b\beta < 1$, double u_i for $i \in D$ and multiply $M_{jb}(s)$ throughout the cell by $(1 + \beta)^b$, using the old β.

In the second rule the increase of the label normalizer cancels the marginal multiplier, so a tuple with v entries in D gains a factor 2^v. With either rule all tuples meeting D at least double, and no tuple loses weight. The first rule is allowed for $b\beta \geq 1/2$, and the second for $b\beta \leq 2$. In the former case $p \geq 1 - \exp(-1/2)$. In the latter case

$$(1 + \beta)^b - 1 \leq \exp(b\beta) - 1 \leq \exp(2)(1 - \exp(-b\beta)) \leq \exp(2)p.$$

Thus the increase factor of a marginal is at most $1 + 8p$ under either permitted rule. When D is empty the update leaves the weights as they are. Summing over cells bounds the pool's growth factor by $1 + 16\eta_b$.

Apply the pool updates before changing any menu. Inserting a quote splits a cell, and both parts inherit its label weights. A rebuild resets the label weights on that axis to one in each pool and preserves $M_{jb}(s)$ at every fine index. Both operations leave the pool totals unchanged.

5.2 Number of rounds

Theorem 5.1 (Grouped sampling)**.** *Take $J = 256n\lceil \log_2(2a) \rceil$ on every axis, including the scalar axis. With exact market computations, a search capped as below succeeds with probability at least $3/4$ over the sampler bits. It uses $O((h+1)B+n\log(2a))$ rounds, $\widetilde{O}((h+1)B+n)$ static queries, and $\widetilde{O}(n^2)$ contracts per query. Over T rounds the exact grid and interval arithmetic takes $\widetilde{O}((h+1)(J+T)B)$ bit operations. The remaining pool work takes $\widetilde{O}(n^2 + eJ)$ operations on $O(\log(2+L))$-bit words per query amortized.*

Proof. First consider rounds in which the trial samples are nets for the distributions used in the weight analysis. Use the potential Φ equal to the sum of the base-two log weights of the ordinary witnesses and the witness tuples. It starts at least at $-2n\log_2 a$. The bracket argument in Section 3.4 gives an ordinary witness doubling or a rounded blocker from a witness group. In the second case the tuple for that group meets D and at least doubles. Thus each failed round adds at least one to Φ before menu repairs. These claims also hold when only one of the bracket endpoints is an interior trial.

A split preserves each witness weight. At a rebuild, a witness tuple of length b on the rebuilt axis has its conditional probability reset to a^{-b} from a value at most one. Its log weight therefore loses at most $b\log_2 a$. By Lemma 4.1, each axis has at most $4T/J$ rebuilds after T failures. The total loss is at most $8n\log_2(a)T/J$.

Let R be the maximum of one and the ceilings of the base-two logarithms of all grid sizes, plus $\lceil \log_2(2h) \rceil$. Then $R = O(B)$ and each pool initially has total weight at most 2^R. The sum of ordinary log witness weights is at most $(2h+1)(R + T\log_2(1 + 4/J))$. Each witness tuple weighs at most its pool total. Since $\sum_b H_b \leq 2h$ and $\sum_b H_b\eta_b \leq 1/C_1$,

$$-2n\log_2 a + T - 8n\log_2(a)T/J \leq \Phi \leq (4h+1)R + (2h+1)T\log_2(1 + 4/J) + \frac{16T}{C_1 \ln 2}.$$

Using $\log_2(1+x) \le x/\ln 2$ and $2h+1 \le 2n$, the sum of the reset and growth coefficients of T is at most $8/256+(8+16)/(256\ln 2) < 1/2$. Hence an attempt cannot have $1+2((4h+1)R+2n\lceil\log_2(2a)\rceil)$ failed rounds while every queried sample is a net. Use this integer as the failed-round cap. Each round uses $O(1+\log J)$ queries. It uses at least one because the scalar axis has an interior fine point and its two endpoint weights are always one. In every quantile initialization or rebuild with $J \ge 3$, the rank at fraction $\lceil J/2\rceil/J$ lies above the first point's weight and at most the total weight minus one, so this quantile selects an interior point. Indeed the rank is between one half and two thirds of a total of at least three. Splits preserve the selected point until a rebuild. Thus query counts can pay for work done once per round.

The exact ordinary weights use $O(B)$ bits by Lemma 4.1 and the round bound, which gives $T/J = O(B)$. To maintain the marginals on a price axis, keep all historical cell boundaries. They partition the grid into intervals with constant per-point marginal in each pool. There are $O(J+T)$ such boundaries per axis, and each new interval length is computed once in $\widetilde{O}(B)$ bit operations. Maintain sums and lazy multiplicative tags in a balanced interval tree. Splitting a leaf copies its per-point weight and uses the new lengths. Range sums use disjoint subtrees so that all contributions are positive. A rebuild may merge current cells but retains their historical subdivisions. Resetting the label weights requires no change to these marginal trees.

There are $O(eJ)$ live label entries per pool. Scan them using the ordered contract cutoffs, and build sum trees for choosing an axis and cell and then for choosing labels within that cell. These trees can be reused while the menus and weights are frozen. Each draw then takes polylogarithmic work per emitted entry. Computing the union weights and applying one range multiplier per cell also takes $\widetilde{O}(eJ)$ short operations per pool per round. Copying or resetting the label vectors costs $\widetilde{O}(a)$ per new or reset cell. Lemma 4.1 bounds this work by $\widetilde{O}(ha(J+T))$ across the attempt, which fits the $\widetilde{O}(eJ)$ amortized query allowance. The selected entry count gives the $\widetilde{O}(n^2)$ sampling term.

It remains to implement the pools with short words. Store each u_i as an exact power-of-two exponent. For positive masses use binary floating point with $C_4\log_2(2+L)$ mantissa bits and a signed binary exponent. Compute β from the two positive sums $\sum_{i\in D} u_i$ and $\sum_i u_i$, handling an empty D directly. The relative error can be made inverse polynomial of any prescribed degree by increasing C_4. Choosing the first rule when the computed $b\beta$ is at least one and the second otherwise therefore selects a rule in the permitted overlap. In the second rule compute the multiplier by repeated squaring of $(\sum_i u_i + \sum_{i\in D} u_i)/\sum_i u_i$. Its relative error is bounded by a polynomial in L times the unit roundoff, since the sums have at most a terms and the power b is polynomial in L.

For a fixed transcript of choices, define an exact weight history using the same branches and label updates. In the marginal trees each per-index weight is the product of the successive multipliers for that index. Pushing a tag to disjoint children and splitting a uniform interval preserve this representation. Each multiplier occurs once per index. The stored interval sums are positive sums of these products times interval lengths. An error bound for these operations follows by multiplying the relative error factors of the individual updates and additions along a summand's computation. Each such product contains at most polynomially many factors in a capped attempt, including the relative errors in the multipliers and length conversions. This gives inverse-polynomial relative error of any prescribed degree for every positive sum used to sample. Exponent magnitudes are polynomial in L: the lengths are at most $2^{O(B)}$, label exponents are bounded by the round cap, and every marginal multiplier is at most $\exp(2)$. All these operations therefore use $O(\log(2+L))$-bit words.

Each branching probability in a sum tree is a ratio of two positive sums and is approximated to inverse-polynomial absolute error of any prescribed degree. Compare it to a uniform grid draw with $O(\log(2+L))$ unbiased bits. Conditional on the current transcript, the probability of differing from the ideal continuous uniform comparison is bounded by the absolute approximation error plus the

grid spacing. Couple all choices of a trial to independent draws from its exact weight history, using fresh random bits conditional on the transcript. There are polynomially many branch decisions per trial, so the chance of a discrepancy is as small an inverse power of $2 + L$ as required. Combine this estimate with the conditional net bound and take a union bound over the capped calls. With probability at least $3/4$ every trial is a net for the exact weight history. The potential argument then forces success within the cap. This proves the theorem. □

5.3 Bit complexity and exact verification

Lemma 5.2 (Static query costs)**.** *Take $B = \Theta(L)$ with $B \geq L$ to bound the precisions in Section 3. On the exact-comparison transcript, the complete-menu search with $J = O(d+1)$ and up to $T = O((d+1)B)$ rounds costs $\widetilde{O}((J+T)L^2 + L^2)$ bit operations before reconstruction. For sampled menus of size $J = \widetilde{O}(n)$ and $\widetilde{O}(n^2)$ contracts per restricted market, the amortized costs per query are*

$$\begin{aligned} &\widetilde{O}(n^3 + nB + e\sqrt{L} + \sqrt{L}B), && \text{site prices,} \\ &\widetilde{O}(n^3(b_0+g_0) + nB + LB/b_0 + LB/g_0 + e\sqrt{L} + \sqrt{L}B), && \text{type costs,} \quad 1 \leq b_0, g_0 \leq B. \end{aligned}$$

The integers b_0, g_0 are arithmetic parameters. Over Q sampled queries multiply the applicable cost by $J+Q$ and add $\widetilde{O}(L^2)$ for preprocessing. This includes initial and later quotes, ranking, residual tests, and the menu and pool work. Conditional on the sampler bits, the probability of a first erroneous modular comparison on the capped exact-comparison transcript can be made an arbitrarily small inverse power of $2 + L$.

Appendix B proves the static query estimates, using b, g for the arithmetic parameters b_0, g_0. For the site orientation the number of updated axes is m, and their maintenance allowance per round is included in nB. Exact reconstruction and the deterministic output check cost $\widetilde{O}(L^3)$ by Theorem 2.1.

Theorem 5.3 (Las Vegas bit bound)**.** *The complete-menu branch has expected total cost $\widetilde{O}((d+1)L^3)$. With sparse restrictions on the smaller side, the expected total cost is*

$$\widetilde{O}\Big((d+1)L\,[n^3 + Ln^{3/2} + nL + L^{3/2}] + L^3\Big).$$

The algorithm returns an exact equilibrium in expected $\widetilde{O}(L^{10/3})$ indexed bit operations for all input dimensions. With comparable k, m, the site-oriented sampled branch takes expected $\widetilde{O}(L^3)$.

Proof. Use the direct solutions in Section 1 at the extreme budgets. For an interior budget, first bound the work of a correct attempt, allowing one reconstruction. The complete branch uses $\widetilde{O}((d+1)L)$ queries by Lemma 4.2, giving $\widetilde{O}((d+1)L^3)$ work by Lemma 5.2.

The sampled branch uses $J + Q = \widetilde{O}((h+1)L)$ by Theorem 5.1, since $B = \Theta(L)$ and $n = O(L)$. For type costs choose $b_0 = g_0 = \max(1, \lceil\sqrt{LB/n^3}\rceil)$. These integers are at most B because $B \geq L$. If the square root is at most one, $LB \leq n^3$ and the two tuning terms are $O(n^3)$. Otherwise the terms $n^3(b_0+g_0) + LB/b_0 + LB/g_0$ are $O(\sqrt{LB}\,n^{3/2})$. With $e = O(L)$ the cost per amortized query on either orientation is at most $\widetilde{O}(n^3 + Ln^{3/2} + nL + L^{3/2})$. Taking $h = d$ gives the stated sparse bound.

For $d + 1 \leq L^{1/3}$ the complete-branch bound is at most $\widetilde{O}(L^{10/3})$. For $d + 1 > L^{1/3}$ substitute $n = O(L/(d+1))$ into the sparse bound. The four terms before reconstruction are bounded, up to logarithmic factors, by

$$\frac{L^4}{(d+1)^2}, \qquad \frac{L^{7/2}}{\sqrt{d+1}}, \qquad L^3, \qquad (d+1)L^{5/2}.$$

They are all $O(L^{10/3})$ because $L^{1/3} < d + 1 = O(\sqrt{L})$. For comparable dimensions, use site prices with $h = m$. Comparability before deletion gives $m, n = O(\sqrt{L})$ by the full input table, and comparability after deletion gives the same bounds by the retained subtable. In either case $J + Q = \widetilde{O}(L^{3/2})$ and the site cost per amortized query is $\widetilde{O}(L^{3/2})$. Adding reconstruction gives $\widetilde{O}(L^3)$.

Cap each attempt's bit work by the applicable bound with sufficiently large constants and logarithmic factors. For uniform discrete choices made by rejection, cap the retry count at $O(\log(2 + L))$ per choice. The total truncation probability is inverse polynomial of any prescribed degree. With the sampler bits fixed, follow the capped transcript obtained with exact market comparisons. This transcript is independent of the random primes. The modular error bound in Lemma 5.2 applies up to its first differing answer. On the exact-comparison transcript the complete branch succeeds deterministically and the sampled branch succeeds with probability at least 3/4, including finite precision in its sampling. The probability of a modular error or truncation can be bounded by 1/4 over an attempt. Thus each attempt succeeds with probability at least 1/2. The independent restarts give at most two attempts in expectation.

Each attempt invokes reconstruction at most once. The output check uses the original rational payoffs and prior. It verifies feasibility, each supported attack against its type's minimum cost, and every positive-weight defense against the pressure order statistic, as detailed in Appendix A. Hence accepting a profile certifies exact best responses regardless of the earlier random computations. □

References

[1] M. Baïou and M. Balinski. The stable allocation (or ordinal transportation) problem. *Math. Oper. Res.* 27(4) (2002), 662–680. Corrected printing.

[2] E. Balkanski, C. En, and Y. Faenza. An algorithm for the assignment game beyond additive valuations. arXiv:2406.13620v3, 2026.

[3] R. P. Brent and P. Zimmermann. *Modern Computer Arithmetic*. Cambridge University Press, 2010. Author's reference.

[4] J. Černý, C. K. Ling, C. Kroer, and G. Iyengar. Layered graph security games. In *Proceedings of IJCAI*, 2695–2703, 2024. Proceedings version.

[5] ChatGPT-6 Astra. Interchangeability in Bayesian security games: Exact balance and generic uniqueness. Research note, 2026. Note cited in Conitzer's revision.

[6] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *J. ACM* 56(3) (2009), Article 14. Published version.

[7] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of ACM EC*, 82–90, 2006. Paper.

[8] V. Conitzer. Open problem: Computing Bayes–Nash equilibrium in simple security games. arXiv:2610.08796v2, 2026.

[9] A. Cseh and J. Matuschke. New and simple algorithms for stable flow problems. arXiv:1309.3701, 2013.

[10] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39(1) (2009), 195–259. Published version.

[11] B. C. Dean and S. Munshi. Faster algorithms for stable allocation problems. *Algorithmica* 58 (2010), 59–81. Author manuscript.

[12] *Exact Bayes–Nash equilibrium in compact Bayesian simple security games*. Preliminary companion manuscript, November 2026.

[13] A. Galichon and A. Jacquet. The matching problem with linear transfers is equivalent to a hide-and-seek game. *Games Econom. Behav.* 152 (2025), 333–344. Published version.

[14] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.* 2 (1987), 127–151. doi:10.1007/BF02187876.

[15] A. V. Karzanov. On one generalization of stable allocations in a two-sided market. arXiv:2504.19978, 2025.

[16] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of AAMAS*, 689–696, 2009. Proceedings version.

[17] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Proceedings of AAAI*, 805–810, 2010. Paper.

[18] D. Korzhyk, V. Conitzer, and R. Parr. Security games with multiple attacker resources. In *Proceedings of IJCAI*, 273–279, 2011. Proceedings version.

[19] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J. Artificial Intelligence Res.* 41 (2011), 297–327. Preprint.

[20] J. Letchford and V. Conitzer. Solving security games on graphs via marginal probabilities. In *Proceedings of AAAI*, 591–597, 2013. Paper.

[21] Y. Li, V. Conitzer and D. Korzhyk. Catcher–evader games. In *Proceedings of IJCAI*, 2016. Extended version.

[22] Y. Li. *Game-theoretically allocating resources to catch evaders and payments to stabilize teams.* PhD thesis, Duke University, 2016. Dissertation.

[23] Y. Li and V. Conitzer. Game-theoretic question selection for tests. *J. Artificial Intelligence Res.* 59 (2017), 437–462. Paper.

[24] T. Oki and R. Sato. Ascending auctions for combinatorial markets with frictions: A unified framework via discrete convex analysis. arXiv:2604.10563v3, 2026.

[25] T. Oki and R. Sato. Finding competitive equilibria in online marketplaces with platform fees, taxes, and subsidies. arXiv:2610.40626v1, 2026.

[26] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordóñez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *Proceedings of AAMAS*, 895–902, 2008. Paper.

[27] A. Sinha, F. Fang, B. An, C. Kiekintveld, and M. Tambe. Stackelberg security games: Looking beyond a decade of success. In *Proceedings of IJCAI*, 5494–5501, 2018. Published version.

[28] B. von Stengel and S. Zamir. Leadership games with convex strategy sets. *Games Econom. Behav.* 69(2) (2010), 446–457. Accepted manuscript.

[29] S. Wang and N. Shroff. Security game with non-additive utilities and multiple attacker resources. *Proc. ACM Meas. Anal. Comput. Syst.* 1(1), 2017. Preprint.

[30] H. Xu. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proceedings of ACM EC*, 497–514, 2016. Preprint.

A Proof of exact reconstruction

The two systems in Section 2.1 share a pattern of permitted attack pairs and coverage statuses. We recover exact prices and flows separately within that pattern. Throughout this appendix, H, δ, and ξ are the row bound and tolerances of Section 2.1.

Lemma A.1 (Choosing tight rows)**.** *Consider either cleared integer system, allowing coordinates fixed to zero to be deleted. If every row violation at z_0 is at most δ, a single sorted pass selects a set of independent rows whose affine space has the following property. The orthogonal projection of z_0 onto that space is feasible and lies within Euclidean distance $H^3\delta$ of z_0.*

Proof. For each row $a_i^\top z \le b_i$ set $s_i = b_i - a_i^\top z_0$ and $h_i = \max(1, \|a_i\|_1)$. Sort by nondecreasing s_i/h_i. Start with no selected rows and radius $R = \delta$. At the next row, stop if $s_i/h_i > R$. Otherwise let α_i be the length of the projection of a_i onto the nullspace of the selected row matrix A. If $\alpha_i > 0$, select the row as an equality and set $R \leftarrow 3h_iR/\alpha_i$. If $\alpha_i = 0$, keep the current space and radius. The implementation uses R^2 and α_i^2. A nonpositive slack always passes the radius test, and a positive slack is tested by squaring both sides.

Let z_A be the projection onto the current affine space. Inductively $\|z_A - z_0\|_2 \le R$. For a row that passes the radius test, $-\delta \le s_i \le h_iR$, so $|a_i^\top z_A - b_i| \le 2h_iR$. When $\alpha_i > 0$, the new equality meets the old affine space at distance $|a_i^\top z_A - b_i|/\alpha_i$ from z_A. Since $\alpha_i \le h_i$, the distance of the new projection from z_0 is at most the updated radius. The radius never decreases. For the rows selected so far, Gram–Schmidt gives $\prod_{i\in A} \alpha_i = \sqrt{\det(AA^\top)} \ge 1$, where the product is in selection order and the empty determinant is one. The radius is therefore at most $3^{|A|}\delta \prod_{i\in A} h_i \le H^3\delta$, using the row product bound and $2^{|A|} \le H$.

If $\alpha_i = 0$, the row residual is constant on the current affine space. Evaluate it at the rational point $A^\top(AA^\top)^{-1}b_A$, or at zero for an empty matrix. Its denominator divides $\det(AA^\top) \le H^2$ by Hadamard's inequality. Its absolute value is at most $2h_iR \le 2H^4\delta < H^{-2}$, so it is zero. The equality persists as the space shrinks. At termination every unprocessed row has slack greater than h_iR and is satisfied at z_A by the distance bound. The selected and skipped rows also hold there. □

Proof of Theorem 2.1. Round the trial coordinates to a common dyadic mesh of step ξ, preserving zeros and nonnegativity of the masses. Retain the original pattern even when a positive coordinate rounds to zero. Clearing a row multiplies its initial violation by at most H, and perturbing its coordinates adds at most $H\xi$. Thus the cleared systems start with violations at most $2H\xi \le \delta$.

Price reconstruction. Apply Lemma A.1 to the price system. For the selected rows other than the budget, join the two variables of each binary row by an edge, and treat a unary row as a pin. The edges on a spanning tree determine each homogeneous coordinate as a scalar multiple of its root coordinate. A pin or a non-tree row that imposes a nonzero constraint on the root sets that coordinate to zero. Every other component C, including an isolated variable when unpinned, contributes one direction w_C, normalized to have root entry one. The directions have disjoint supports. Write B_x for the row of the unsigned budget equality and set

$$Z_C = w_C^\top w_C, \qquad G_C = B_x w_C, \qquad S = \sum_C G_C^2/Z_C,$$

where the sum is over the free components. A candidate row a with $U_C = a^\top w_C$ has squared projected length

$$\sum_C U_C^2/Z_C - \frac{(\sum_C U_C G_C/Z_C)^2}{S}$$

when a budget orientation is active, and the first sum alone otherwise. These are the projections onto the component directions and then onto the nullspace of the budget. An active budget row was selected independently and therefore has $S > 0$. A budget candidate has squared projected length S before a budget orientation is selected and zero afterwards. A zero row falls under the dependent case of Lemma A.1. Rebuilding these data from the at most $k + m$ selected rows costs $O(n)$ rational operations per candidate.

For the final projection, solve the tree equalities in a free component as $z_v = \beta_v + w_v t_C$ for $v \in C$. In a pinned component, a fixing row supplies the root value. Independence ensures that the selected affine equations are consistent. In each free component the unconstrained minimizer of the squared distance from z_0 has $t_C = \sum_{v \in C} w_v(z_{0v} - \beta_v)/Z_C$. If there is an active budget, replace each t_C by $t_C - \mu G_C/Z_C$, where substituting in the budget determines the common μ since $S > 0$. The resulting point is the projection specified in Lemma A.1 and satisfies every price constraint.

All reduced operands have $O(L)$ bits. For a component, charge the bit lengths of the spanning-tree rows and any fixing row to that component. The product of the nonzero tree coefficients clears the path denominators. A propagated affine constant is a sum of at most $|C|$ products along a path, with each tree row used at most once per product. A fixing equation introduces one additional division by a rational of the same bit bound. Squared norms and inner products use a fixed power of the resulting component denominator. Allow $O(\log(2 + |C|))$ bits for the summation overhead. These bounds sum to $O(L)$ over the disjoint components by the row product bound. Consequently the denominators of fractions summed over components, including S, have a product of $O(L)$ bits. The coordinates of the rounded z_0 share one dyadic denominator, used only once in the affine projection. Also

$$R^2 = \delta^2\, 9^{|A|} (\textstyle\prod_{i \in A} h_i)^2 / \det(AA^\top)$$

has $O(L)$ bits. Evaluating the initial slacks and sorting them costs $\widetilde{O}((e + n)L)$ bit operations. The projection tests and reconstruction use $\widetilde{O}((e + n)nL) \subseteq \widetilde{O}(L^3)$ bit operations with fast rational arithmetic [3].

Flow reconstruction. Consider the positive rounded masses. Insert their edges into an initially empty graph on the types and targets, maintaining a forest of positive edges. When an insertion creates a cycle, alternately increase and decrease its edge masses, decreasing the new edge, by the smallest mass on a decreasing edge. Remove all zero edges. The cycle move preserves the incident mass sum at every vertex and removes at least one edge of the unique cycle. Thus the final forest preserves all rounded row and site totals. Nonnegativity and the lower and upper pressure residuals are preserved, as are the row mass residuals and threshold bounds. The new zero masses also respect the original permitted pairs. The cycle moves keep the common dyadic scale, and a search of the current forest costs $O(n)$ operations per edge inserted.

Restrict the flow system to the resulting forest and denote the rounded starting threshold by λ_0. By Lemma A.1 this system has a feasible point within $H^3\delta$ of the thinned starting vector. Its set of feasible thresholds is a nonempty closed interval, since the exact row sums bound all its nonnegative masses and the threshold itself is bounded. Each endpoint is attained at a vertex of the bounded polytope. Cramer's rule and the product of the row norms give reduced denominator at most H at a vertex, also when the polytope has lower dimension, by using a full-rank selection of its tight rows.

First test λ_0 for feasibility. If it fails, the nearest point of the interval is an endpoint within $H^3\delta < 1/(4H^2)$ of λ_0. It is the only fraction with denominator at most H in that radius, since two such fractions are separated by at least $1/H^2$. Legendre's continued-fraction criterion gives this endpoint among the convergents of λ_0, since the error is less than $1/(2q^2)$ for its reduced denominator q. Use the floor convention and include the integer and terminal convergents. Enumerate the convergents, test the radius and denominator bounds, and test the recovered endpoint for feasibility.

This also covers an interval consisting of one point. The Euclidean algorithm takes $O(L)$ divisions of operands of $O(L)$ bits, and the convergent recurrence and tests fit $\widetilde{O}(L^2)$ bit operations.

To test a threshold, first check $0 \le \lambda \le \max_j g_j$. For $g_j > 0$, divide the applicable pressure constraint in (2) by g_j and intersect with the site load range $[0, 1]$. For $g_j = 0$, check the same constraint with $P_j = 0$ and allow all of $[0, 1]$ if it holds. Orient each edge from its type to its site and root each component. At a vertex the divergence is required to lie in an interval $[v^-, v^+]$, a singleton $\{\pi_i\}$ at a type and the negative of the load interval at a site. If the sum of the signed child contributions at this vertex ranges over $[c^-, c^+]$, its parent edge can contribute any value in $[v^- - c^+, v^+ - c^-]$ intersected with its signed range from a mass in $[0, 1]$. Reverse the sign to obtain the edge's interval as seen at its parent. At a root require zero in the computed parent-contribution interval. Reject empty intervals. Conversely, for an accepted parent contribution choose a child sum in the intersection of $[c^-, c^+]$ with the required divergence interval shifted by that contribution. Starting each child at its lower bound and distributing the remaining increase realizes the sum. A downward pass therefore constructs the flows. This argument applies to isolated vertices with child sum zero. The restriction $q_{ij} \le 1$ is implied by the exact row sums. All interval endpoints and restored flows use a common scale from the prior denominators, reciprocal positive gains, and the single tested threshold, with $O(L)$ bits. The passes and cycle deletions fit $\widetilde{O}(L^3)$ bit operations.

Strategy extraction and verification. Set the conditional attack probabilities to q_{ij}/π_i for the retained types. For the defense, put $s_0 = 0$ and $s_j = \sum_{h \le j} x_h$, so $s_m = r$. For $v \in [0, 1)$, include target j when $[s_{j-1}, s_j)$ contains a point of $v + \mathbb{Z}$. The r points $v, \dots, v + r - 1$ in $[0, r)$ each belong to exactly one interval, and each interval has length at most one. Hence the selected set has size r. The set of shifts selecting target j has measure x_j by reduction of that interval modulo one. In particular an empty interval never selects its target and an interval of length one always selects it. Between successive distinct fractional parts of $s_0, \dots, s_{m-1}$ and the right endpoint 1, the selected set is constant in the interior. Use an interior representative of each interval and its length as the probability. This lists at most m defenses. Every target with $P_j > \lambda$ is always included and every target with $P_j < \lambda$ is omitted, so all listed subsets are best responses by Section 2.1.

The formula for the affine projection gives a common price denominator dividing the product of the dyadic denominator of z_0 and $\det(AA^\top)$. The lottery weights share that denominator, and the forest pass supplies a common flow scale. All these denominators have $O(L)$ bits. Verify the output by checking the attack probabilities and their row sums, their joint masses, and the cost minimum at every positively used target. Restore each zero-prior type at a minimizing target under the recovered coverage. Check the lottery weights for nonnegativity and normalization, recompute its marginals, and check that each positive-weight subset has size r, includes all pressures strictly above the rth largest, and includes none strictly below it. Ties at that order statistic are allowed. The common scales bound all sums in these checks. Even an explicit scan of m incidence vectors of length m uses $\widetilde{O}(m^2 L) \subseteq \widetilde{O}(L^3)$ bit operations. This completes exact reconstruction and verification. □

B Bit implementation of static queries

Take $B = \Theta(L)$ large enough to cover all heights and precisions in Section 3. A short word, including an index into a polynomial-size store, has $O(\log(2 + L))$ bits. We count its access cost. Arithmetic and division on s-bit integers take $\widetilde{O}(s)$ bit operations [3].

B.1 Comparing sums of long numbers

Lemma B.1 (Modular sign comparisons)**.** *Consider a family of forms $S = \sum_a c_a v_a$, with integer coefficients satisfying $\sum_a |c_a| \le U$ for a common $U \ge 1$. The primitives have been scaled by a common power of two so that $|v_a| < 1$. Every nonzero form has magnitude at least 2^{-h_0}, with $h_0 + \log(2U) = O(B)$. Assume an oracle at depth t supplies the residue of an integer $A_{a,t}$ with $|A_{a,t} - 2^t v_a| \le C$ for an absolute integer $C \ge 1$. The represented integers are independent of the modulus.*

There is a random modulus of length $O(\log(2L) + \log(2U))$ for which a sign evaluation uses $O(\log(2B))$ residue queries per term and $\widetilde{O}(\log(2L) + \log(2U))$ additional bit operations per query. For a prescribed polynomial number of evaluations, the probability of any erroneous answer can be made at most $(2 + L)^{-c}$ for any fixed c. This also covers zero tests of $O(B)$-bit integers given their residues. For an explicit rational primitive of $O(B)$ bits, preprocessing takes $\widetilde{O}(B + \log(2U))$ bits of work and each prefix query takes $\widetilde{O}(\log(2L) + \log(2U))$ bit operations.

Proof. Put $V = CU$ and $W_t = \sum_a c_a A_{a,t}$, using $A_{a,0} = 0$. The error bounds give

$$|W_t - 2^t S| \le V, \qquad |W_{t+1} - 2W_t| \le 3V.$$

Once W_t exits $[-3V, 3V]$, it stays outside on the same side. At its first exit, its magnitude is at most $9V$, and its sign is the sign of S. A zero form never exits. Every nonzero form has exited by depth $h_0 + \lceil \log_2(5V) \rceil + 2$. Thus one test at the final depth and a binary search for the first exit suffice.

With a modulus $M > 20V$, use the centered residue to test the window and to read the first exit. An error requires M to divide some nonzero integer $W_t - j$ with $|j| \le 9V$. For zero tests of exact integer forms, the bad integer is the tested value itself when nonzero. All these integers have $O(B)$ bits.

For sufficiently large L, sieve the primes in $[L^2, 2L^2]$ in $\widetilde{O}(L^2)$ bit operations. There are $\Omega(L^2/\log L)$ such primes by the prime number theorem. A nonzero $O(B)$-bit integer has at most $O(B/\log L)$ distinct divisors in this prime list. Use the product of $\lceil C_0(1 + \log(2U)/\log(2L)) \rceil$ independent uniform primes from the list, allowing repetitions. For any fixed bad integer, divisibility by the product requires each drawn prime to divide it, so the probability is at most $(C'/L)^{\lceil C_0(1+\log(2U)/\log(2L)) \rceil}$. The product has the claimed length and exceeds $20V$ for sufficiently large C_0. The candidate bad integers are indexed by polynomially many intended queries and depths and $O(U)$ offsets per depth. A union bound gives the stated failure probability by increasing C_0. For an adaptive computation, fix the other random choices and use the sequence of queries with exact answers. A first error must occur on this sequence. The argument requires only a constant number of modulus families here, and each uses its own bound U. Small values of L may use exact arithmetic.

For an explicit rational, compute its signed binary expansion through the maximum depth by one scaled division. Use a Euclidean floor for the integer part. Store the residues at block boundaries spaced by the bit length of M, and append the remaining digits from the preceding boundary on demand. The total preprocessing and per-query costs follow from short modular arithmetic and the $O(B)$-bit expansion. □

B.2 Static allocation on restricted markets

Let N be the number of records after grouping adjacent private gaps and adding the permanent and balancing contracts. A group's mass is a difference of cumulative amounts σz with z a physical grid endpoint. At a fixed trial scalar t, the priors, quotas t/g'_j, and cumulative private amounts all have the integral scale $D_* 2^s$ of Section 3, where 2^s is the reduced denominator of t. The total

scaled mass and quotas have $O(B)$ bits. These facts give the separation required by Lemma B.1 for every nonzero signed sum of the primitives, including after a common power-of-two normalization.

Lemma B.2 (Static allocation from short comparisons)**.** *Suppose every retained record has its ranks at its two endpoints. Each cumulative private endpoint supplies the bounded-error prefixes of Lemma B.1 and its exact residue on the integral scale in $\widetilde{O}(u)$ bit operations, with $u \geq 1$. An endpoint can be materialized in $\widetilde{O}(B)$ bit operations. Assume the ordinary margins have been preprocessed. If all modular comparisons are correct, a stable allocation with its positive ordinary flows and all site private totals can be computed in $\widetilde{O}((N+n)nu + nB)$ bit operations.*

Proof. Use the proposal and refusal rules of Dean and Munshi [11, Section 3]. Represent a vacancy by an assignment from a dummy buyer. A dummy site receives the permanent outside assignments. Initially the dummy buyer fills each site's quota. Each other buyer has a separate temporary edge to the dummy site, currently carrying that buyer's full mass. This edge represents unprocessed mass and has no market rank. The dummy site's total load is then the sum of the nondummy buyer masses and all vertices are balanced. Individual record bounds are nonbinding.

Process one buyer at a time until its temporary edge is empty. At a real site the refusal points to its least preferred positive record, including the vacancy record if present. A buyer's proposal points to its most preferred contract that strictly beats the refusal of its receiving site, or to the outside option if there is no such real bid. The dummy buyer proposes only to the dummy site. Omit sites of zero quota and their incident real bids from this calculation. They have neither private holdings nor vacancy. All sites have positive quotas at the interior query scalars.

Follow the pointers from the current buyer. If the route ends at the dummy site, close it using that buyer's temporary edge. Increase all proposal flows and decrease all refusal flows and the temporary flow by the same amount. If the route enters a directed cycle, augment on the cycle alone, with proposals increasing and refusals decreasing. In each case the amount is the least flow on a decreasing record. This preserves every vertex balance and exhausts a refusal or the current temporary edge, possibly both. The temporary edge and the permanent outside edge may close a two-record cycle. A directed cycle between a buyer and a real site also uses distinct records, because a proposal strictly beats that site's refusal.

A site's refusal can only improve. A proposal can move down a buyer's list only when the former record becomes unacceptable to the site. If that record still carries flow, it must be the site's refusal: site acceptance fails at equality and no positive record ranks below the refusal. It stays the refusal until exhausted. Thus every positive ranked record belongs to the current proposal/refusal graph. A refusal exhausted by an augmentation never receives flow again. There are $O(N+n)$ augmentations, and cached proposal pointers can each skip a record permanently when it becomes unacceptable. At termination all permanent assignments are feasible. If a buyer holds mass on a record, any strictly better record was unacceptable when that mass was assigned, and remains so by monotonicity of refusals. This proves stability after removing the balancing records.

For the numerical implementation, maintain the forest of positive records, including the temporary records. Initially it consists of two stars and isolated vertices. In the proposal/refusal graph every vertex has one outgoing pointer except the dummy site. Its component is a tree, and every other component has a single cycle, counting parallel records as separate edges. Other unprocessed buyers have just their temporary edge in the positive support. They are leaves there and cannot be reached by a refusal.

Let Z be the zero proposal records added in a single augmentation. For a route reaching the dummy site, the affected proposal/refusal tree together with the current buyer's temporary edge has one cycle. For a route entering a cycle, the temporary edge connects its unicyclic component to the dummy-site component. In either case the union of Z and the old positive forest contains exactly

one cycle in the affected component: the augmented cycle. The other temporary edges are at leaves and create no cycles in that union. The augmentation deletes a decreasing edge of this cycle, so the new positive support is again a forest. In particular Z is nonempty, since the old positive support is acyclic. Summing balances over either side of a forest cut now expresses a positive flow as a signed sum of vertex balances.

Process the private buyers first, in decreasing priority at each site. Their only alternatives are their private bid and outside. This leaves at each site a fully served prefix of its private groups, at most one partial group, and a fully displaced suffix. This initial state can be found directly: binary search the cumulative masses of the private groups for the site's quota, compare with the total private mass, and fill any remaining vacancy by the dummy buyer. This is exactly the private-first outcome, including when the quota equals a cumulative mass. The initialization takes $\widetilde{O}(nu)$ work after the rank lists are available. Subsequent real proposals can displace only the least preferred private group still present. A private buyer reached by a refusal immediately proposes to the dummy site.

A fully held group is a leaf at its site and a fully displaced group is a leaf at the dummy site. Suppress these leaves and add their positive divergences to the signed balances of their parents. Each full prefix and suffix total is a difference of two cumulative endpoints. Keep the possible partial group's vertex and its two endpoints. The dummy site's adjusted balance uses a sum of $O(m)$ endpoint terms and the original total mass. The dummy buyer's original balance is the total site quota. The resulting forest has $O(n)$ vertices and $O(n)$ primitive occurrences in its signed balances. During an augmentation restore a suppressed group only if it is the current refusal on the route. There is at most one such leaf because its next pointer reaches the dummy site. Its endpoints and the prefix and suffix boundaries give the adjustments in the same representation.

To find the augmentation amount, cut the augmented cycle at its members of Z. The remaining positive segments lie in distinct trees of the old forest. Otherwise an old forest path connecting two segments, in addition to the augmented cycle, would give another cycle in the union with Z. List the decreasing edges in order within each segment. For two of them, delete both from that tree and take the component between them, including its branches off the segment. The sum of its signed balances equals the difference of their flows, up to the chosen order. Indeed the two decreasing records are traversed against their buyer-to-site orientations, so one is directed into the component and the other out of it.

Pair neighboring candidates in each ordered list, retain a minimum of each pair, and repeat with the survivors in order. At one level the between-edge portions are disjoint, so they use $O(n)$ primitive occurrences in total. One can enumerate these occurrences by assigning each off-path vertex to its attachment to the segment and scanning the paired intervals. A second tournament compares one winning cut expression per tree. These expressions have disjoint trees at each level and again use $O(n)$ occurrences. The coefficients are signed integers of total absolute value $O(n)$ in every comparison. Lemma B.1 and the endpoint access bound give $\widetilde{O}(nu)$ work for both tournaments.

Compute the cut flows once by subtree sums modulo the same small modulus on the integral scale. Comparing each decreasing record's residue with that of the bottleneck identifies all exhausted records, including a simultaneous exhaustion of the temporary edge. The protected zero tests of Lemma B.1 apply to those differences. Update incidences and the private prefix boundaries using the exhausted records. Maintain positive real ranks in an ordered set at each site. The worst rank among these records, the last occupied private group and any vacancy gives the current refusal. This permits constant-many short rank queries, up to logarithmic factors, at each step of a route or rank scan. A route uses $O(n)$ vertices after the private phase. Refresh cached proposals against the current refusals on demand. There are $O(N+n)$ total rank skips and $O(n)$ incidence inspections or changes per augmentation.

Finally materialize the $O(n)$ retained balance terms and compute cut sums on the integral scale to recover the positive ordinary flows, private holdings, and vacancies. This final work takes $\widetilde{O}(nB)$ bit operations. Combining it with the rank scans and $O(N+n)$ augmentations proves the bound. □

B.3 Physical prices from type quotes

Let the physical grid spacing be 2^{-D}, where $D = O(B)$ is at least the fractional binary length of every type quote. Use arithmetic parameters $b, g \in [1, B]$ as in Lemma 5.2. A tariff is light if its two original coefficients have total bit length at most b. At most $O(L/b)$ tariffs are heavy. For a light tariff, the inverse evaluated at a dyadic quote y has the form

$$v = 2^h(py+q)/s, \qquad X = \lfloor 2^D v \rfloor, \qquad z = 2^{-D}X,$$

with positive integers p, s and signed integer q of $O(b)$ bits. For a positive original slope take $h = 0$ and clear the original denominators. For an originally flat tariff whose perturbed slope is 2^{-h}, take the rational form of $y - a_{ij}$ inside the shift. First test eligibility against the two tariff endpoints and the outside cost, using two-term sign comparisons with stored explicit prefix tables. This takes polylogarithmic work per label and quote. The following calculations use only eligible prices, so $0 \le v \le 1$.

Fix the modulus M for the small-coefficient comparisons. For $0 \le t < D$ put $u = t + h$ and $Y_u = \lfloor 2^u y \rfloor$. Let f be the truncation of the nonnegative fractional part of $2^u y$ to $\lceil \log_2(p+1) \rceil + 1$ binary places. Compute the residue of the following integer:

$$A_t = \left\lfloor \frac{pY_u + 2^u q + \lfloor pf \rfloor}{s} \right\rfloor.$$

The inner floor can be omitted when taking the outer floor, since the other terms are integers and the divisor s is integral. Replacing the true fractional part by f changes the unfloored quotient by less than one. Taking the outer floor costs less than one more, and $0 \le 2^t(v - z) < 1$. Hence $|A_t - 2^t z| < 3$. For any signed integer numerator w,

$$\lfloor w/s \rfloor \bmod M = \lfloor (w \bmod (sM))/s \rfloor,$$

where the remainder on the right is nonnegative. Thus one computes the numerator only modulo sM. At depth D the fractional part of $2^{D+h}y$ vanishes. The same formula gives $X \bmod M$ exactly. For $t > D$ use $2^{t-D}X \bmod M$, and for negative depths use zero.

The cumulative mass σz and the fixed family normalization both use power-of-two shifts of the requested depth. Its exact scaled residue follows by multiplying $X \bmod M$ by the residue of the integral scale divided by the dyadic denominator of $\sigma 2^{-D}$. Perform that last division on the known common scale before reduction. This works even if a denominator and M share a factor.

For a newly inserted quote on a type axis, build a product tree of the light-label moduli sM, including repeated factors. Its root length is at most the sum of their bit lengths. Read the binary digits of the quote in blocks of g. At each boundary update the signed prefix modulo the root, and use a remainder tree to obtain its residues at the leaves. The signed integer part requires one initial division. Also prepare the explicit prefix table for y itself in $\widetilde{O}(B)$ work. Pad with zeros beyond the fractional digits. There are $O(B/g + 1)$ boundaries up to the maximum needed depth, including the common shift. Each step costs $\widetilde{O}(g + \sum \mathrm{bits}(sM))$ bits of work in that row. The sum of the label and modulus lengths over the table is $\widetilde{O}(L)$, so inserting one quote per type costs $\widetilde{O}(kB + LB/g)$ for these tables. For a requested prefix, append at most g bits to the preceding stored residue, read

$O(b)$ fractional bits for f, and compute 2^u modulo sM by repeated squaring. This gives $\widetilde{O}(b+g)$ work per prefix or terminal residue. The cost of forming a heavy label's rounded price and explicit prefix table is $\widetilde{O}(B)$ per new quote, for $\widetilde{O}(LB/b)$ when inserting once on every type axis. The permanent endpoints and their tables cost $\widetilde{O}((e+n)B)$ once.

B.4 Rank lists and residual tests

On type axes, list the $O(eJ)$ eligible offers at the active quotes. A buyer's quoted cost is either a shared dyadic quote, an outside cost, or a permanent endpoint. Use its stored order or explicit prefix table. For site rankings compare physical prices with two unit coefficients and the preceding oracle. Ties use the fixed priorities of Section 2, and an optional copy of a permanent bid has the same record identifier. Sorting the ranks and grouping equal physical prices costs $\widetilde{O}(eJ(b+g))$. Form the restricted lists from the selected offers and the permanent records. Consecutive retained physical prices at each site specify the endpoints and relative ranks of its private groups.

On site axes a quote is an explicit dyadic shared by all bidders at that site. At insertion, store its cumulative private-mass prefix table and exact residue on the integral scale in $\widetilde{O}(B)$ work. Put $c = \lceil \sqrt{L} \rceil$. For tariffs with more than c coefficient bits, prepare their exact quoted costs and prefix tables. There are $O(L/c)$ such tariffs. Prepare the explicit permanent endpoint and outside costs once. For a comparison between two quoted costs, clear the denominators of any light tariffs among the operands. The resulting expression has $O(1)$ terms and coefficient bit lengths $O(c)$. Its primitives are constants, shared dyadic quotes, power-of-two shifts of them for perturbed flat tariffs, and any explicit heavy or permanent costs. Their heights are $O(B)$ in bits. The product of this constant number of primitive denominators supplies the separation in Lemma B.1. The comparison modulus has length $O(c + \log(2+L))$. Its prefix tables are prepared at cost $\widetilde{O}(B)$ per explicit primitive. Inserting one quote on each site thus costs $\widetilde{O}(ec + (m + L/c)B)$, including comparisons for inserting the new keys in each affected buyer's ordered dictionary. Store both the numerical order with equality classes and the refined strict order. Old keys may remain as historical entries. Using their rank handles to assemble restricted lists costs $\widetilde{O}(eJ)$ per trial. Menu maintenance allows $O(J+T)$ insertions on each axis over T rounds, so the stated insertion costs are multiplied by $O(J+T)$ even when the axes insert at different times.

The restricted solve returns the positive ordinary records, private totals, and vacancies. To test omitted rounded bids, the buyer cutoff is its worst positive option, including outside. At a site, acceptance uses vacancy, the worst positive real rank, or the private cutoff. The last holds for a physical price z exactly when $z > x_j = 1 - U_j/\sigma$, by the descending occupation of the expanded private gaps and the grid alignment from Section 2. This strict test also handles an exact grid boundary and a partially occupied group. Combined with the strict real rank test it gives a single cutoff in the physical ordering. Locate it by binary search. On type axes the required physical prices can be materialized for those comparisons in $\widetilde{O}(B)$ work apiece. A vacancy accepts every offer, and a zero private total contributes no private acceptance. The price comparisons to locate the cutoffs take $\widetilde{O}(nB)$ bit operations overall. Using the two endpoints' cutoff ranks to mark eligible blocking labels uses $\widetilde{O}(eJ)$ short operations. The same predicate tests a bid already present for an increase because its record has no binding individual cap. For the ordinary gap updates, compute the fine boundary indices in Lemma 3.5 from these exact cutoffs and find the corresponding open menu gaps in $\widetilde{O}(nB)$.

For the original-game residual screen, prepare the exact x_j, the tolerance, and the original costs on the $O(L/c)$ heavy labels, together with their prefix tables. Each comparison of original costs, or of a used cost with the row minimum plus the tolerance, is an $O(1)$-term form with $O(c)$-bit coefficients after clearing the light denominators. Exact ties are zero forms and the tolerance test is

inclusive. There are $O(e+n)$ comparisons. Their bit cost is

$$\widetilde{O}(ec+(n+L/c)B).$$

After redirecting outside mass the positive attack pairs still number $O(n)$, since each type redirects to one cap site and the ordinary flows lie on a forest. Aggregate and test the masses and site pressures on the common flow scale, with one multiplication by g_j per pressure. Compute the coverages on the common scale from the private totals and $1/\sigma$, and test the budget, the two pressure inequalities as applicable to the statuses, and the budget's comparison with r' for a failed screen. Preparing quotas and ordinary primitives for the trial and these exact tests all fit $\widetilde{O}(nB)$ work. Reconstruction is needed only after the screen passes.

B.5 Proof of the static query bounds

Proof of Lemma 5.2. For complete menus on the smaller side, Lemma 4.2 gives $J=O(d+1)$ and $T=O((d+1)B)$ failed rounds. For each newly introduced label–quote pair, compute the exact cost, physical price, order keys, and explicit prefix data in $\widetilde{O}(B)$ bit operations. Maintain ordered dictionaries with exact comparisons at insertion, charged to that insertion. By Lemma 4.1 the lifetime cost is $\widetilde{O}((J+T)(e+n)B)$. There are $N=O((e+n)(d+1))$ contracts per query. In Lemma B.2 use $u=1$ in the soft bound, since the explicit tables give polylogarithmic access. The solve costs $\widetilde{O}(nN+nB)=\widetilde{O}(L^2)$ because $(d+1)n=O(e+n)$. Forming the restricted lists and testing the original residuals with full precision fit the same query bound. Including setup and menu maintenance, this is $\widetilde{O}((J+T)L^2+L^2)$ as claimed.

For sampled menus there are $J=\widetilde{O}(n)$ active quotes per axis and $N=\widetilde{O}(n^2)$ retained contracts. Apply Theorem 5.1 with $h=m$ for site prices and $h=k$ for type costs. If there are T failed rounds and Q interior queries, then $T\le Q$. The total exact grid and historical interval work is $\widetilde{O}((h+1)(J+T)B)$, covered by the nB allowance multiplied by $J+Q$. The remaining pool operations cost $\widetilde{O}(eJ+n^2)$ per amortized query and are covered by $\widetilde{O}(n^3)$.

For site prices the forest solve uses $u=1$ in the soft bound. The ranking and screening costs above, including the lifetime charge for new quotes, give total

$$\widetilde{O}((J+Q)(n^3+nB+e\sqrt{L}+\sqrt{L}B)+L^2).$$

For type costs use $u=b+g$. The cost $\widetilde{O}(eJ(b+g))$ of ranking fits the $\widetilde{O}(n^3(b+g))$ solve allowance. The lifetime prefix-table charges for new quotes give $LB/b+LB/g$ inside the amortized sum, while kB is included in nB. The result is

$$\widetilde{O}((J+Q)(n^3(b+g)+nB+LB/b+LB/g+e\sqrt{L}+\sqrt{L}B)+L^2).$$

The prime sieve, integral scales and permanent keys are included in the one-time $\widetilde{O}(L^2)$ term. For the error bound fix the sampler bits, and apply Lemma B.1 to each comparison family on the exact-comparison transcript up to the round cap. This transcript is independent of the moduli. Any first difference must occur there. This gives the claimed bound for the entire search. □