

HOW RECORD SQUARE PACKINGS ARE FOUND

Human oversight: [Joshua Levy](#)

Agents: **GPT-6 Astra** and **GPT-6 Sol**

v0.2.1 ([version history](#))

First published October 8, 2026 · Last revised October 10, 2026

Evidence cutoff: 10 October 2026. The selected-bound tables include the confirmed 8 October intake; newer replayed certificates awaiting adoption are identified separately.

A record square packing can begin as a geometric sketch, a random configuration, or pieces borrowed from an earlier record. The final coordinates may then pass through several optimizers before a verifier establishes that the squares fit. Understanding that chain explains both how a new arrangement is found and why an improvement in its last decimal places can be a different contribution.

Contents

1. [The Problem and the Kind of Result](#)
2. [From a Seed to a Certified Bound](#)
3. [The Geometry a Search Has to Navigate](#)
4. [Constructing and Reusing Seeds](#)
5. [Moving Between Arrangements](#)
6. [Refining a Promising Arrangement](#)
7. [Turning Coordinates into a Bound](#)
8. [Reading the Record](#)
9. [Vocabulary and Further Reading](#)
10. [Version History](#)

The Problem and the Kind of Result

Let $s(n)$ be the smallest side of a square that can contain n **unit squares**, each free to translate and rotate. The small squares must have disjoint interiors; edges and corners may touch. A construction that fits in a container of side S establishes the **upper bound** $s(n) \leq S$.

Two elementary bounds put the search in context. The pieces have total area n , so $s(n) \geq \sqrt{n}$. An axis-aligned grid supplies $s(n) \leq \lceil \sqrt{n} \rceil$. At a perfect square these meet: $s(m^2) = m$. At $n = 11$ they leave the interval $\sqrt{11} \leq s(11) \leq 4$. Tilting some pieces permits a smaller container than that grid. Walter Trump's 1979 construction uses six axis-aligned squares and five at one common angle near 40.18° , in side approximately 3.87708359. Its geometry provides an upper bound; proving that no other arrangement improves it is a separate problem ([historical survey](#); [tutorial](#)).

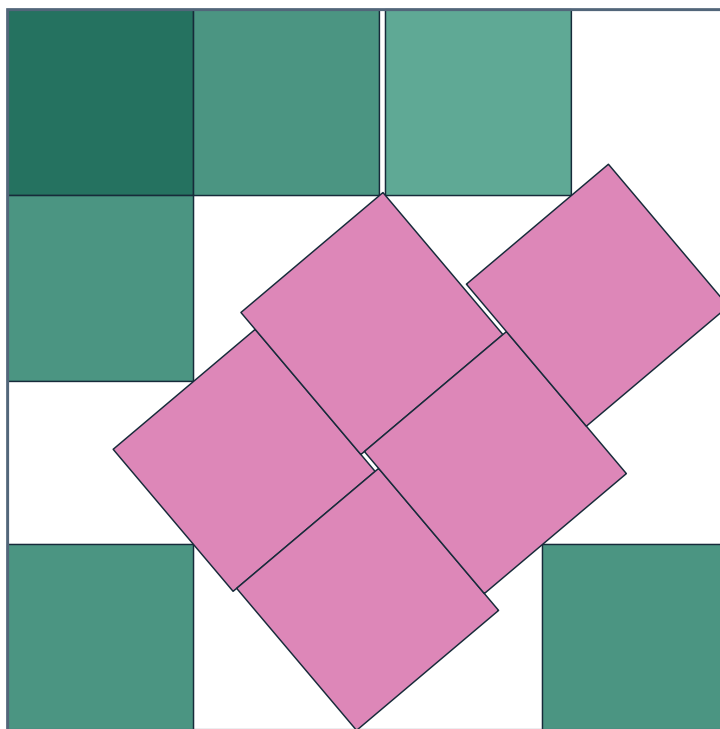


Figure 1. Walter Trump's 1979 packing of eleven squares: six axis-aligned squares surround five tilted at a common angle. This hand construction shows how a small number of geometric parameters can describe a useful arrangement. Redrawn from the retained reconstruction from David Ellsworth's geometry; its side is approximately 3.87708359.

A square's **pose** consists of its centre (x_i, y_i) and orientation θ_i . A quarter turn gives the same square, so angles can be represented modulo $\pi/2$. Together with the side S , a configuration has $3n + 1$ continuous variables. To recover its corners, write $c_i = \cos \theta_i$ and $d_i = \sin \theta_i$. The four corners are

$$(x_i, y_i) + \frac{1}{2}(u(c_i, d_i) + v(-d_i, c_i)), \quad u, v \in \{-1, 1\}.$$

Place the container at $[0, S]^2$. A candidate packing must satisfy two kinds of condition: every corner stays in the container, and every pair of squares has disjoint interiors. Reducing S leaves less room to satisfy these conditions together. A picture can suggest that the conditions hold, but tiny overlaps or an inward-rounded side can be invisible. The search produces candidate poses; certification checks the geometric conditions at the side actually claimed.

From a Seed to a Certified Bound

The workflow has four jobs. They can be separate programs, or loops inside one program.

JOB	INPUT AND OPERATION	OUTPUT
Construct a seed	design a pattern, choose a random state, or transform an existing packing	a starting configuration, possibly infeasible
Explore	move far enough to reach different arrangements or local basins	promising candidates
Refine	reduce the side near a candidate, adjusting centres, angles, or contact equations	a numerical packing or stationary candidate
Certify	check unit shape, containment, and all pairwise non-overlap rigorously	a feasible witness and an upper bound

A **seed** need not already be a packing. Adding a square to a crowded arrangement, for example, can create overlaps that the following search must resolve. A successful seed supplies useful structure: orientations, rows, a tilted block, or empty space where rearrangement can begin.

Fix a deterministic local refiner, including its settings. The starting configurations it returns to one endpoint form a **basin** for that refiner. Exploration tries to reach another basin; refinement tries to improve the configuration it has reached. If stronger refinement closes a gap to a record, it has exposed removable slack in the candidate. Repeated returns to an inferior endpoint justify trying other starts with that refiner. They do not establish that all local improvement is exhausted: the refiner may have held angles fixed, constrained squares to share an angle, or stopped at a tolerance or budget limit.

One record can therefore have several contributors. A finder discovers the arrangement; another optimizer reduces its side; a high-precision solver removes numerical slack; and an independent checker establishes the upper bound. David Ellsworth's Kingbird site is a catalogue and refinement site, continuing Erich Friedman's earlier catalogue. It is not the name of a single search algorithm.

The Geometry a Search Has to Navigate

CHOOSING A SEPARATING BRANCH

Two convex squares have disjoint interiors exactly when a line weakly separates them. The **separating-axis theorem** says that for convex polygons, it suffices to test axes perpendicular to their edges. Each square has two distinct edge normals, giving up to four axes for a pair; along each axis there are two possible orders.

For a unit direction ν , one order requires

$$\max_{p \in Q_i} \langle \nu, p \rangle \leq \min_{q \in Q_j} \langle \nu, q \rangle,$$

where Q_i and Q_j are the two squares. Only their corners need be tested. Equality permits contact. The pair is non-overlapping if *at least one* axis and order works. That logical choice is a **disjunction**; one selected alternative is a **separating branch**.

For an exact small example, take two axis-aligned squares. The branch placing square 1 to the left of square 2 is

$$x_1 + \frac{1}{2} \leq x_2 - \frac{1}{2}, \quad \text{or equivalently} \quad x_2 - x_1 \geq 1.$$

The centres $(1/2, 1/2)$ and $(3/2, 1/2)$ satisfy this with equality in a side-2 container. Placing one square above the other uses a different branch, $y_2 - y_1 \geq 1$. Both are valid constructions. Restricting the search to horizontal separation excludes that stacked construction.

A **cell** selects one separating branch for every pair. Cells can overlap: two squares far apart diagonally may satisfy both a horizontal and a vertical branch. A cell describes separation choices for all pairs, including pairs that never touch. A **contact graph** records only touching pairs, and so does not specify a cell ([tutorial's cell decomposition](#)).

WHY FIXING ANGLES HELPS

With all angles fixed, every corner is its centre plus a constant offset. Containment then consists of linear inequalities in x_i, y_i, S . Fix a separating branch for each pair as well, and its axis is constant: the projection inequalities are linear too. Minimizing S under those restrictions is a **linear program**, or LP.

This isolates a tractable subproblem inside the larger search. An LP can optimize centres and side for one choice of angles and branches. The full problem still has to choose branches and move angles. A change in which separation condition binds can also create a kink in the optimized side as a function of angle. Smooth optimization methods have to account for those changes.

Cells and basins refer to different objects. A cell is a geometric restriction; a basin is defined by what a particular refiner does. The refiner can change angles and separating branches along its route. Its settings and tie-breaking rules affect the basins. If endpoints are grouped by pattern or by a continuous family of poses instead of identical coordinates, that grouping must be specified too. Three squares in a side-2 container, for example, admit an exact sliding family: two occupy the bottom row, and the centre of the top square moves from $(1/2, 3/2)$ to $(3/2, 3/2)$. Different returned coordinates can belong to that same family ([tutorial's basin discussion](#)).

Constructing and Reusing Seeds

DESIGNING A PARAMETRIC FAMILY

A hand construction chooses an arrangement before optimizing its dimensions. Frits Göbel placed a diagonal strip of 45° squares across an axis-aligned background. Wider strips, rational-slope tilts, and an L-shaped border supplied related counts. Trump’s eleven-square packing and Hämäläinen and Gustafson’s eighteen-square constructions likewise reduce the search to a small geometric family. The design specifies how pieces meet; algebra or one-dimensional optimization determines a remaining angle ([historical survey](#)).

Composition reuses a good packing as a block inside a larger one, or wraps an L-shaped layer around it. Such constructions can produce exact, reproducible sides. They rely on anticipating a useful pattern. The $n = 29$ arrangement found by Gensane and Ryckelynck uses six orientation classes, illustrating the complexity that an unrestricted computer search can find beyond a simple hand-designed family.

CHANGING THE COUNT OR REPLACING A COMPONENT

Packing surgery transforms an existing arrangement into a seed, often changing which pieces can touch. The retained SQUISH publications distinguish operations by how the seed is assembled, even though their later search and polishing are incompletely disclosed.

Deletion and addition connect neighboring counts. Deleting pieces preserves feasibility at the old side and creates room for rearrangement; adding pieces supplies a harder starting state. SQUISH’s original release says nine of its ten original packings began from its own neighboring certified packings this way, while $n = 126$ began with nearby search from the published record. A supplement added $n = 153$. Later reports give explicit deletion chains: $182 \rightarrow 180$ removes two squares, while $155 \rightarrow 154$ and $239 \rightarrow 238$ remove one, followed by nearby search and polishing ([original release](#); [first update](#); [second update](#)).

Grafting installs a useful subpacking in a target-count arrangement; **carving** reduces a substantially larger arrangement to a target-count seed. The five operations below distinguish what structure the author reports transferring. Nate Chaoweeraprasit, using SQUISH, reported all the resulting poses; the names in the lineage column credit the seed contributors. An arrow names the parent or component count and the resulting total count. For a graft, it does not mean that the smaller packing alone supplies all the target’s squares.

OPERATION	AUTHOR-REPORTED LINEAGE AND SOURCE	STAGE AND TRANSFERRED STRUCTURE
Neighbor-count deletion	SQUISH 110 $\rightarrow$ 108; Francisco Couzo 180 $\rightarrow$ 179 (second update)	Seed construction: remove two squares or one, respectively, preserving the rest at the old side; then search nearby and polish.
Component grafting	Couzo 102 $\rightarrow$ 123, 105 $\rightarrow$ 126, and 210 $\rightarrow$ 239 (first update)	Seed construction: transplant a useful component into a target-count catalogue packing before further optimization.
Updating an inherited core	Kingbird catalogue 41 $\rightarrow$ 88, using the Schadt/Ellsworth core refined in January 2026 (core history ; second update)	Seed construction: replace the older core in the $n = 88$ construction with its newer version, then refine the resulting packing.
Reusing a newly improved component	SQUISH 88 $\rightarrow$ 207, 236, 302 (second update)	Seed construction: reuse the newly improved $n = 88$ component in three larger target packings, propagating one gain into several counts.
Carving a larger construction	Couzo 297 $\rightarrow$ 263 (second update)	Seed construction, exploration, then refinement: reduce the larger arrangement, search nearby, and apply a final squeeze.

For the inherited core, the author explicitly says the $n = 41$ improvement post-dated the older $n = 88$ construction ([lineage explanation](#)). The final squeeze is a separate local search stage: roughly 120 random perturbations, at amplitudes 10^{-5} to 10^{-3} times the side, each followed by polishing. It improved $n = 263$ by about 4×10^{-10} ; at the other eight counts in the second update, it found no improvement above 10^{-11} ([reported protocol](#)).

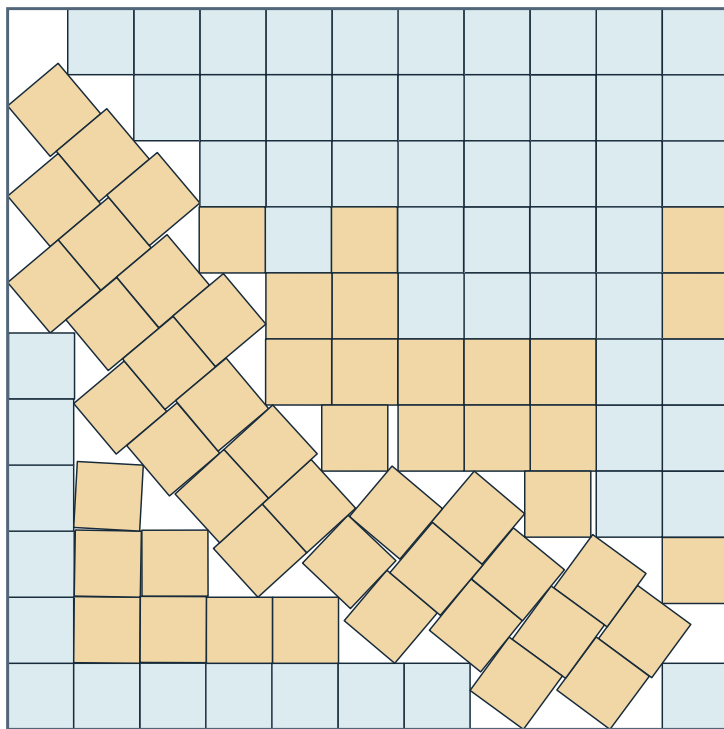


Figure 2. A historical SQUISH packing of 108 squares, reported by Nate Chaoweeraprasit (itsnaka) on 7 October 2026. The reported seed was an $n = 110$ packing with two squares removed, followed by nearby search and polishing. The drawing shows the resulting certified pose; the source does not identify the two removed squares, and the search has not been reproduced here. Its safe side ceiling 10.9099400734448775 has since been superseded by Xu’s $n = 108$ witness. Redrawn from retained exact coordinates (source certificate).

These are author-reported lineages. Independent exact replay confirms feasibility of the resulting poses; it does not reconstruct the surgeries or reproduce the searches. The packets do not disclose which squares were removed, graft interfaces, carving decisions, or the optimizer behind nearby search and squeeze. Francisco Couzo supplied many of the seeds: his September release contains 49 improved packings, and his October 3 update improves seven of them. He credited Claude assistance for the first $n = 102$ and 103 results, but the retained September and October 3 publications specify no discovery algorithm. Those publications and subsequent reuse establish available seeds; they do not reveal how the original seeds were found ([September packet](#); [October update](#)).

Moving Between Arrangements

Exploration needs a rule for proposing moves and deciding which to keep. Methods called physical, billiard, or annealing can use quite different rules.

HARD-PARTICLE MOTION AND GREEDY BILLIARDS

In physical billiards, bodies travel between collision events. A simulation predicts the next collision and updates velocities when it occurs. Lubachevsky–Stillinger dynamics also grows the particles, making collisions progressively more frequent as the packing densifies. This has been effective for disks. Rotating polygons make collision prediction much harder, because the equations involve changing orientations as well as centre motion. Two-dimensional compression can also favor ordered arrangements; for congruent squares the axis-aligned grid is an obvious ordered state. No freely rotating square record in this catalogue is attributed to Lubachevsky–Stillinger dynamics ([retained simulation survey and primary references](#)).

Gensane and Ryckelynck’s 2005 square **billiard** instead tests proposed end-points. It grows congruent squares inside a fixed box. If the common square side is a in a unit box, rescaling gives unit squares in side $1/a$; maximizing a minimizes the final container side.

At fixed size, the algorithm chooses one square and proposes a random translation and rotation of amplitude ε . It accepts the new pose only if it overlaps neither another square’s interior nor the outside of the box. It does not simulate the path between the old and new poses. If this walk permits growth, it inflates the squares and doubles ε ; otherwise it halves ε . A second phase perturbs *every* square, recomputes an admissible common size, and runs the billiard again. It keeps the refined result only if it improves the previous one.

Thousands of random starts used $\varepsilon = 0.1$ initially and 10^{-8} as the first pass’s stopping threshold. Collective perturbations then used a stopping threshold of 10^{-12} . This is a greedy stochastic walk with adaptive step size and collective shakes. It produced the 2004 packings reported at 5.934342 for $n = 29$ and 6.603236 for $n = 37$ ([published pseudocode](#)). The $n = 29$ arrangement remained best known until Schadt’s December 2025 search ([catalogue history](#)).

FORCES FROM AN OPTIMIZATION PENALTY

A physically inspired optimizer can instead allow temporary overlap, assign a cost to it, and move squares in a direction that reduces that cost. For a *schematic* example, let δ_{ij} be a chosen nonnegative overlap measure and w_i a wall-violation measure. With positive penalty weights λ and μ , an objective could take the form

$$E = S + \lambda \sum_{i < j} \delta_{ij}^2 + \mu \sum_i w_i^2.$$

The first term favors compression; the other terms oppose invalid poses. Where the chosen functions are differentiable, their negative gradients with respect to centres and angles determine translations and torques for a descent step. A program might add damping, noise, or a schedule for the penalty weights. Those are choices in an optimizer; calling its derivatives forces does not imply that it simulates elastic collisions. The formula here explains the mechanism and is not an attribution to any record producer.

The surrogate objective needs care. A finite penalty can trade overlap against a smaller side, so reducing E does not by itself produce a packing. The gradient of a squared penalty also tends to zero as a smooth violation closes. For rotating squares, changing the best separating branch introduces further nonsmooth behavior. Exact geometric checking must follow any such search ([simulation survey](#)).

ANNEALING AND BASIN HOPPING

Simulated annealing evaluates a proposal using an objective or energy. It accepts favorable moves and sometimes accepts unfavorable ones. A common schematic rule accepts a worsening $\Delta E > 0$ with probability $\exp(-\Delta E/T)$, where T is the **temperature**. Cooling reduces that probability. Temperature controls acceptance of worse objective values; the procedure need not integrate physical trajectories. The objective, proposal distribution, and cooling schedule all affect which arrangements it reaches.

Thomas Schadt's program found a new $n = 29$ arrangement from randomness in December 2025 and later found the $n = 51$ and $n = 55$ structures. His public $n = 29$ packet says discovery used C++ `float`, followed by Boost high-precision relaxation. It does not publish the energy, proposal distribution, or cooling schedule, so the schematic acceptance rule above should not be read as a reconstruction of that program ([Schadt packet](#)).

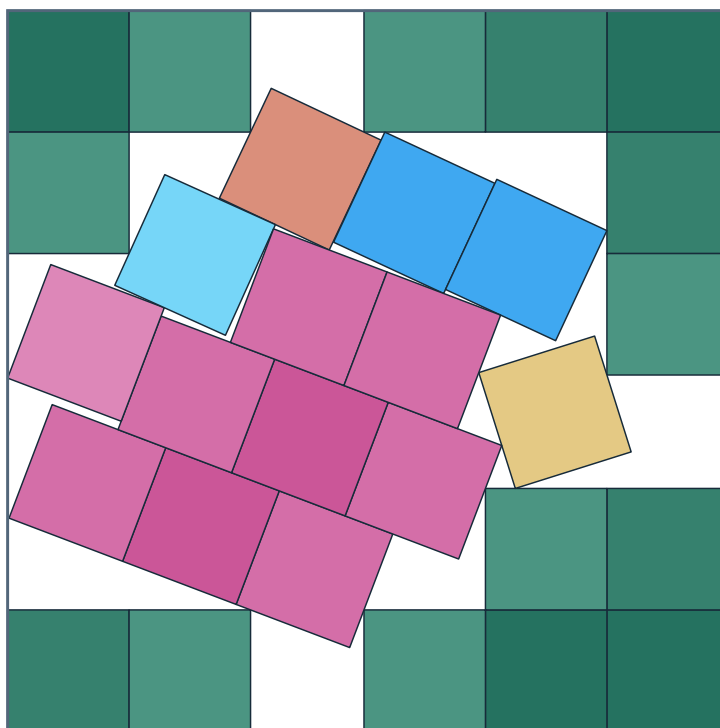


Figure 3. The 29-square arrangement found by Thomas Schadt’s simulated annealing in December 2025 and subsequently improved analytically by David Ellsworth. The mixture of tilted and axis-aligned pieces illustrates an arrangement reached through stochastic search. This drawing uses the retained numerical reconstruction, rather than the original floating-point search output. The separate interval certificate supplies the safe side ceiling 5.9338334626769292.

Ellsworth modified Schadt’s annealer, ran parallel GPU searches, and then performed analytic minimization. The $n = 51$ record was refound from randomness. The $n = 55$ breakthrough began from a cherry-picked state produced by Schadt’s reimplement of the Gensane–Ryckelynck search; Ellsworth later refound its basin from randomness. These histories separate the seed’s contribution from the annealer and refiner ([Kingbird comparison](#)).

Griffin Casson’s September 2026 release gives a reproduction command with 8,192 single-precision GPU chains, a twelve-minute search setting, and twelve CPU polishers. The command seeds the search from published records; the reported GPU was an RTX 3070. Annealing supplied candidates; sequential linear programming refined them. It found new arrangements at $n = 106$ and 123. Direct local refinement of published records supplied another 37 reported numerical improvements ([first-party README](#)).

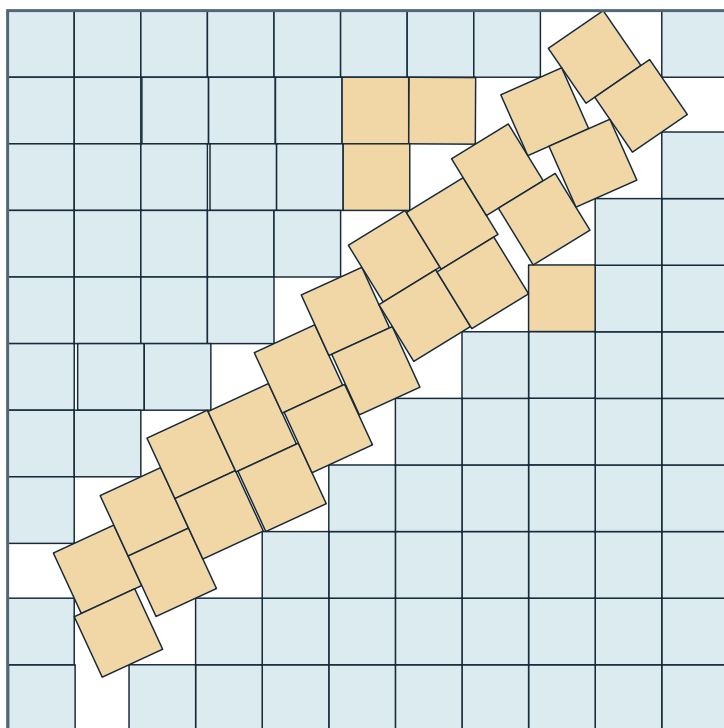


Figure 4. Griffin Casson’s September 2026 packing of 106 squares, one of the two arrangements reported from GPU simulated annealing followed by sequential linear programming. Redrawn from the dated source coordinates (pinned original, CC BY 4.0). This source pose is an illustration of the reported workflow; it has not undergone an independent local feasibility replay here.

Basin hopping inserts local refinement into the exploration loop: perturb a locally optimized pose, refine the perturbed pose, and apply an acceptance rule to the refined outcome. Thus even a large raw perturbation can count as no escape if refinement returns to the same endpoint. Casson’s README names a CPU basin-hopping component, but its detailed document and parameters are not retained. SQUISH, short for “Square-packing Using Iterative Shrink-Hopping,” explicitly reports basin hopping after seed transformations, followed by polishing. Its published evidence does not disclose the objective, local optimizer, square-selection rule, or hop schedule ([Casson README](#); [original SQUISH submission](#)).

Couzo’s 8 October follow-up discloses a more specific workflow. He starts from Ryan Xu’s packings at $n = 84, 86, 105, 175$ and Daniel’s at $n = 270$. Couzo reports basin hopping at four counts; at $n = 175$, Ellsworth’s `refine_packing` followed by Daniel’s `fq` supplied the refinement. Daniel’s exact contact solver wrote the rational certificates. These disclosures identify tools and stages, while leaving the perturbation distribution, acceptance rule, and stopping criteria unspecified. The complete certificates have passing native feasibility replays here; their adoption as standing bounds remains pending at this survey’s cutoff ([eight refinements; five follow-up refinements](#)).

Daniel’s 9 October record hunt supplies a reported example of moving uphill between minima. He reports starting from Couzo’s $n = 132$ packing, taking an uphill excursion through intermediate minima, and applying contact/KKT refinement and rational certification. The retained certificate permits independent feasibility checking; the search path and numerical local-minimum checks have not been reproduced here ([record-hunt packet](#)).

Joost de Winter describes the 16 September 2026 $n = 211$ result as full-packing adaptive search, grouped-angle local refinement, and interval-verified decimal export. This identifies the stages but not their detailed mechanics. Sharing an angle among selected squares can reduce the nonlinear degrees of freedom; the source does not specify how its groups were selected or constrained. No code, adaptive schedule, or interval boxes were published ([retained packet](#)).

Refining a Promising Arrangement

FIXED-ANGLE LP AND ANGLE REFINEMENT

A **fixed-angle LP** holds all angles and one separating branch per pair fixed, then minimizes S over centres and side. The formulation is exactly linear. A floating-point solver still has tolerances: its returned coordinates can slightly violate an inequality. Even an exact optimum solves only that chosen subproblem; other branches or other angles may admit smaller sides.

This project’s **quench**, its local refiner, alternates two loops. The inner loop reads separating branches from the pose, solves their fixed-angle LP, then reads branches again from the result. It repeats until the selected cell stabilizes, or reports that it stopped unsettled. The outer loop brackets and minimizes angles shared by selected groups of squares. An optional pass moves individual angles as well. Angle refinement can improve a pose whose fixed-angle LP no longer improves ([quench description](#)).

Sequential linear programming (SLP) can instead let angles move inside each local LP. Near the current pose, it replaces nonlinear angle-dependent constraints with their first-order approximations. It solves that linear subproblem, updates the pose, and builds another approximation. Casson’s implementation uses HiGHS and moves each square’s centre and angle. It polished both annealer outputs and slightly perturbed copies of catalogue records. Even an exactly solved linearized subproblem can produce a pose that violates the original nonlinear constraints, so the resulting geometry needs a feasibility check ([implementation description](#)).

The distinction is the variables in the LP. Fixed-angle LP has exact linear separation constraints and moves centres and side. Rotational SLP also moves angles, using a linear approximation to their nonlinear effect. Both are local ingredients; neither searches all branch choices.

CONTACT EQUATIONS AND ANALYTIC RECONSTRUCTION

Near a settled arrangement, likely **contacts** identify equalities: a particular corner touches a particular edge, or an edge or corner meets a wall. Knowing only that two squares touch does not specify the equation. A contact graph therefore needs geometric feature information before a solver can use it.

Ellsworth writes contact equations, eliminates coordinates where possible, and adds stationarity equations if the contacts leave freedom to change the side. For example, suppose a reduced contact equation is $F(S, a) = 0$, where a is one angle and $F_S = \partial F / \partial S \neq 0$. Locally it defines S as a function of a , with

$$\frac{dS}{da} = -\frac{F_a}{F_S}.$$

A stationary side along this smooth family therefore satisfies $F_a = 0$. The contact equation and derivative equation can be solved together. With more variables, a **Jacobian** collects first derivatives; appending the side’s gradient to the contact gradients gives rank conditions that Ellsworth expresses through determinants ([analytic minimization notes](#)).

On a regular smooth contact family, **stationarity** means the side's derivative vanishes in every tangent direction. A zero determinant alone can also arise from redundant or singular equations. Stationary solutions can be maxima or saddles, and motions that open contacts or change separating branches require separate analysis. High-precision root finding supplies numerical candidates. Integer-relation methods then seek small integer coefficients for a polynomial nearly zero at the numerical side. A rigorous algebraic reconstruction must identify the intended root and verify the equations and every packing inequality ([tutorial's reconstruction](#)).

Evan Daniel's October 2026 solver uses **Karush–Kuhn–Tucker (KKT) equations**. For a smooth branch, write the pair and wall clearances as $g_k(q) \geq 0$, where q collects the side, centres, and angles. At a feasible pose the KKT conditions require multipliers λ_k satisfying

$$\nabla S = \sum_k \lambda_k \nabla g_k, \quad \lambda_k \geq 0, \quad \lambda_k g_k(q) = 0.$$

Here ∇ is the vector of first derivatives with respect to q . The last equation allows a nonzero multiplier only at a closed contact; a closed contact may still have zero multiplier. The first equation balances the pressure to reduce S against contact forces and torques. Under suitable regularity conditions, KKT is necessary for a branch minimum. Singular branches can require more general first-order conditions, and satisfying KKT alone does not prove a minimum ([tutorial's stationarity conditions](#)).

Daniel's equilibrium LP identifies the **load-bearing contacts**: contacts that can carry positive force in at least one nonnegative equilibrium. Squares with no such contact are force-free; they include **rattlers**, squares with room for local motion. His solver projects the pose onto contact equations and takes high-precision Newton steps in the KKT system: each step uses a linear approximation to correct the current solution. When contact equations are redundant, he selects an independent subset and checks the omitted equations afterwards. Flat motions, such as sliding between parallel sides, can make the Newton system singular. He freezes the corresponding variables during the solve, then checks their omitted stationarity equations. Force-free squares are moved to gain clearance where possible; those that cannot gain positive clearance are brought into the contact system ([solver's active-set treatment](#)).

Corner-to-corner touches require additional care: the squares can separate along different branches. Daniel sets these alternatives apart from the smooth contact equations and tests first-order motions with a **mixed-integer model**, combining continuous motions with discrete separating-branch choices. His numerical checks also examine contact multipliers and second-order behavior. Of 321 configurations with certificates, 315 passed the producer’s numerical local-minimum checks. These are numerical tests, not interval proofs of a KKT root or of local optimality. Even a rigorous result for one smooth contact branch would leave other corner-contact branches and better arrangements elsewhere to be addressed ([producer method and limitations](#)).

Contact selection can change during the numerical solve. Daniel’s updated implementation reselects its independent contact subset when Newton’s Jacobian becomes nearly singular near convergence, as reported for $n = 130$. For $n = 105$, it selects one corner-contact separating branch for the KKT and second-order solve. Those conclusions concern the selected branch; other admissible branches need separate checks ([updated solver limitations](#)).

Turning Coordinates into a Bound

A **witness** supplies the container side and every square’s pose. A certificate adds the data and interpretation needed for a rigorous checker. Discovery and high-precision refinement help make that witness, but feasibility can be established without proving that its coordinates are the exact numerical optimum.

SQUISH stores a rational container side, rational centres, and a rational half-angle parameter $t = \tan(\theta/2)$ for each square. Define

$$c = \frac{1 - t^2}{1 + t^2}, \quad d = \frac{2t}{1 + t^2}.$$

The identity $c^2 + d^2 = 1$ holds exactly. Thus (c, d) and $(-d, c)$ are perpendicular unit directions, and the corner formula gives an exact unit square. A verifier can check every corner’s containment and every pair’s separation with rational arithmetic, allowing boundary contact ([certificate explanation](#)).

For a small exact example of the representation, take $t = 1/3$, so $c = 4/5$ and $d = 3/5$. Centre the square at $(7/10, 7/10)$. Its corners, in cyclic order, are $(4/5, 7/5)$, $(7/5, 3/5)$, $(3/5, 0)$, and $(0, 4/5)$. They lie in the side- $7/5$ box, and consecutive corner differences have length one. All of these statements are rational checks. This illustrates certification of a pose; a single axis-aligned square already fits in side 1.

For a full packing, the verifier checks the square count and unit shape, every wall inequality, and non-overlap for every pair, whether by explicit separating-axis tests or another rigorous geometric route. Rigorous interval methods can instead enclose uncertain quantities and prove the required inequalities over the enclosures. A gap interval that straddles zero leaves that separation test undecided; a tighter enclosure, another separating axis, or an exact contact identity may resolve it. Every pair still needs a rigorous non-overlap decision.

Daniel first expands centres and container slightly while keeping the pieces unit size, then rounds centres and half-angle parameters to rationals. The added clearance permits exact verification without identifying the original optimum as an algebraic number. The upper bound is the side of the resulting certified packing, including enlargement and rounding. A shorter decimal claimed as an upper bound must round the certified side upward ([producer certificate construction](#)).

This matters even when the source publishes many digits. SQUISH's second $n = 263$ update prints 16.7404196795387747, below its exact certificate side. The safe sixteen-place ceiling is 16.7404196795387766. The exact certificate remains valid; the inward-rounded display does not itself follow as an upper bound ([retained exact side and displays](#)).

Rational rounding is one route to a certificate. Ryan Xu's separate **undilated algebraic witness** at $n = 51$ uses axis-aligned and 45° squares in side

$$S = \frac{16 + 5\sqrt{2}}{3}, \quad s(51) \leq S.$$

The exact 45° directions have coefficients $\sqrt{2}/2$. They belong to the number field $\mathbb{Q}(\sqrt{2})$, whose elements have the form $a + b\sqrt{2}$ with rational a, b . A checker can represent an element by the pair (a, b) and decide its sign with exact rational comparisons. This preserves the exact square shapes and contacts while checking the same containment and separating-axis conditions. The native number-field route and a separate coefficient-pair and corner implementation checked all 51 squares; both accepted the witness and rejected duplicate-square and outside-container controls. This undilated witness is checked separately from the dilated rational certificate of the same arrangement. It proves feasibility at S , without establishing that S is the minimum ([complete geometry and replay scope](#)).

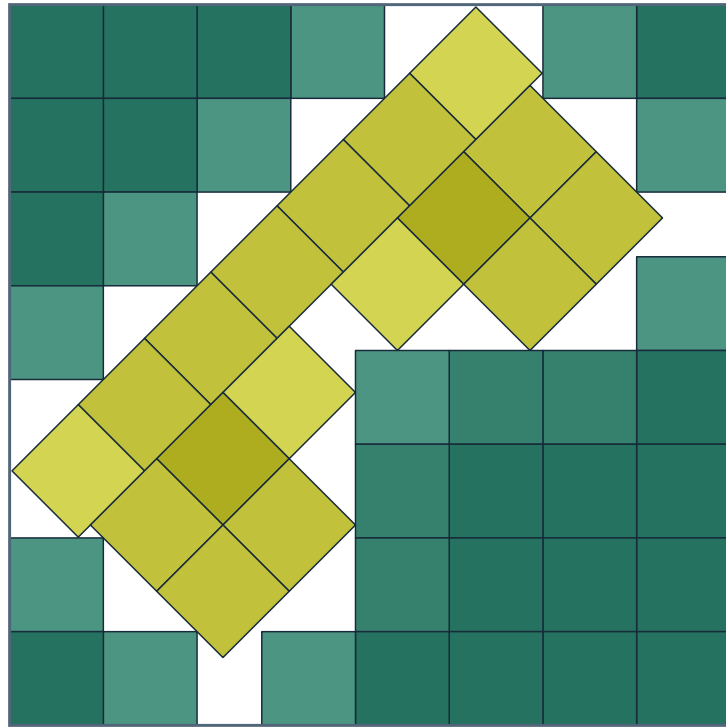


Figure 5. Ryan Xu’s October 2026 packing of 51 squares, using only axis-aligned and 45° pieces. The arrangement’s structure permits an exact certificate over $\mathbb{Q}(\sqrt{2})$ at side $(16 + 5\sqrt{2})/3$. Redrawn from the retained algebraic witness drawing. The certificate verifies feasibility; the reported LLM-assisted workflow leaves the exploration algorithm unspecified, and the picture supplies no optimality proof.

Three conclusions require different evidence:

CONCLUSION	WHAT MUST BE ESTABLISHED
Feasibility, hence an upper bound	the listed unit squares fit in the claimed side
Local optimality	no smaller-side packing exists in a specified neighborhood, including applicable contact branches
Global optimality	no smaller-side packing exists anywhere, usually by a matching lower bound

A valid witness establishes the first. A long decimal, equilibrium equations, or unsuccessful perturbation tests do not supply the other two.

Local rigidity asks whether a specified packing admits nontrivial feasible motion at fixed container side, after declared symmetries are removed; the two optimality statements above concern whether the side can be reduced ([tutorial](#)). The atlas's *R* badge can reflect a proved local-rigidity result or a reported catalogue assertion. When comparing methods, record which configuration was assessed and whether its rigidity evidence is proved or reported. A numerical single-square translation screen with no hit establishes neither rigidity nor local optimality: it omits rotations and coordinated motions. Rigidity evidence belongs to the assessed configuration. A different packing at the same side may have different motions, as the retained alternatives at $n = 52, 149, 296$ illustrate ([source index](#)).

Reading the Record

METHODS AND RECORD EXAMPLES

The categories below place each method at its stage in the workflow. Examples retain their finder and source credit, including historical records that have since been improved. A source's description of discovery or refinement and an independent feasibility check answer different questions.

METHOD FAMILY	STAGE AND MECHANISM	SOURCE, FINDER, AND REPRESENTATIVE RESULT
Geometric construction and composition	Seed construction or a complete construction: design diagonal strips, tilted cores, boundary extensions, or combinations of smaller packings; optimize the remaining parameters.	Frits Göbel's strip families and Walter Trump's 1979 $n = 11$ construction are historical examples (Friedman survey).
Billiard / inflation search	Exploration and refinement: translate and rotate squares by random admissible endpoint moves, increase their common size, adapt move amplitudes, and shake the whole configuration when jammed.	Thierry Gensane and Philippe Ryckelynck's 2004 $n = 29$ packing set a historical record. Their $n = 37$ result was a reported improvement but did not beat Cantrell's earlier packing (published algorithm ; catalogue history).
GPU simulated annealing	Exploration: run many stochastic chains in parallel from random or selected seeds; refine promising outputs separately.	Thomas Schadt found the $n = 29, 51$, and 55 structures with his annealer. David Ellsworth used its modified GPU version to refine and refine the latter two, then analytically optimized the packings. The energy, proposal distribution, and full cooling schedule remain unpublished (record histories).
Sequential linear programming (SLP)	Local refinement: repeatedly linearize angle-dependent separation constraints and optimize centres, angles, and side; restart from perturbed poses when useful.	Griffin Casson's September 2026 release reports two new arrangements, at $n = 106, 123$, from GPU annealing followed by SLP, and 37 numerical improvements from SLP directly on catalogue packings (first-party method and results).

METHOD FAMILY	STAGE AND MECHANISM	SOURCE, FINDER, AND REPRESENTATIVE RESULT
Surgery followed by basin hopping	Seed construction, exploration, then refinement: delete, add, graft, or carve; perturb and locally optimize the transformed seed, then polish.	Nate Chaoweerasaprasit's SQUISH releases report neighboring-count seeds and explicit component lineages. The resulting rational packings have independent exact checks here; detailed hop mechanics remain undisclosed (original packet ; update lineages).
Contact-equation and stationarity refinement	Local refinement: identify geometric contacts and solve their equations together with stationarity conditions, removing numerical slack near an existing arrangement.	Ellsworth's analytic optimization is an antecedent of Evan Daniel's contact/KKT solver. Daniel's 5 October 2026 batch supplied 48 smaller certified sides, principally for existing arrangements. Gupta's later SQUISH refinements use Daniel's optimizer; exact rational certification follows the numerical solve (analytic method ; batch report ; Gupta refinements).
Adaptive search with grouped-angle refinement	Exploration and local refinement, as reported: search the full packing, then refine groups sharing an angle. The adaptive algorithm and grouping rule are unspecified.	Joost de Winter's $n = 211$ release names those stages and interval-verified export. Its checker and interval boxes are unpublished; separate exact and interval checks here establish feasibility of the supplied pose (disclosure and independent checks).
Discovery method undisclosed	Discovery remains unknown; exported coordinates supply a witness for separate feasibility checks. They do not identify the search algorithm.	Francisco Couzo's September 2026 release supplies 49 improved packings, with subsequent revisions. That release names no search method or checker; exact and interval checks here establish safe bounds independently (provenance and certification).

METHOD FAMILY	STAGE AND MECHANISM	SOURCE, FINDER, AND REPRESENTATIVE RESULT
Basin hopping with named refiners	Exploration and local refinement: perturb and optimize an inherited packing, then solve contacts and export a rational certificate. The detailed hopping rules remain unpublished.	Couzo's 8 October follow-up names basin hopping, Ellsworth's <code>refine_packing</code> , and Daniel's <code>fq</code> and contact solver. Passing feasibility replay and standing-bound adoption are separate stages (workflow and status).
LLM-assisted workflow, search algorithm unspecified	Assistance describes how work was conducted; it does not identify an energy, proposal rule, or optimizer. Complete poses and certificates permit separate feasibility checks.	Ryan Xu's $n = 51$ radical construction and 25 rational certificates have independent finite feasibility checks. The retained disclosure does not justify assigning a finer exploration algorithm (source and verification).

DISCOVERY GAINS AND NUMERICAL GAINS

The releases below combine the four jobs in different ways. Counts describe the release's scope at publication, rather than the number of current records.

PUBLICATION	PRODUCER AND DISCLOSED METHOD	SCOPE AND REPRESENTATIVE RESULT
16 September 2026	Joost de Winter : author-reported adaptive full-packing search and grouped-angle refinement	a certified $n = 211$ packing at side $14.99796070496771500150 < 15$
23 September 2026	Griffin Casson : GPU annealing and SLP, plus direct SLP on records	39 reported improvements: two new arrangements, at $n = 106, 123$, and 37 local refinements
23–27 September and 3 October 2026	Francisco Couzo : published configurations and revisions; discovery algorithm unspecified	49 counts, then seven updated; October $n = 208$ lowers the verified ceiling from 14.937018796984568 to 14.926534459703512
5 October 2026	Evan Daniel : contact/KKT refinement and rational certification; some inputs received an unpublished SLP squeeze first	48 certified improvements of about 3.5×10^{-13} to 5.0×10^{-11} , principally numerical slack removed from existing arrangements
7 October 2026	Nate Chaoweeraprasit , SQUISH : author-reported neighbor-count seeds, basin hopping, grafting, carving, and polishing; exact certificates	23 distinct counts across the original release, supplement, and two updates; second-update $n = 108$ has verified ceiling 10.9099400734448775
8 October 2026	Ryan Xu : LLM-assisted workflow, with exploration algorithm unspecified	25 rational certificates and a separate undilated radical $n = 51$ witness; 17 rational witnesses and the radical witness supply selected bounds
8 October 2026	Siddharth Gupta : precision refinement of Chaoweeraprasit’s SQUISH constructions with Daniel’s optimizer	17 verified rational certificates; 14 selected improvements and three withdrawn offers retained as historical inputs

PUBLICATION	PRODUCER AND DISCLOSED METHOD	SCOPE AND REPRESENTATIVE RESULT
9 October 2026	Evan Daniel : reported uphill excursions between minima, contact/KKT refinement, and rational certification	two replayed certificates: $n = 132$ below the selected bound and $n = 155$ at the same exact side as Couzo’s pending T-128 certificate; review and adoption remain pending

Casson’s 37 direct-refinement improvements have all since been superseded in this repository’s frontier ([source coverage](#)). That does not change their role as evidence that local optimization can expose slack in a published arrangement.

SQUISH’s original $n = 108$ ceiling was 10.9206589394033085. Its second-update gain of about 0.01072 followed deletion of two squares from its $n = 110$ packing and nearby search. This exceeds the 10^{-10} -scale final squeeze reported for $n = 263$ by many orders of magnitude. Both $n = 108$ configurations have confirmed feasibility; the producer supplies the account of how the new seed reached the improved result ([original certificate](#); [second update](#)).

SELECTED BOUNDS AFTER THE OCTOBER 8 INTAKE

The integrated register contains 32 smaller selected upper bounds than at this survey’s first publication: 18 from Ryan Xu and 14 precision refinements by Siddharth Gupta. The table gives safe sixteen-place decimal ceilings for the previous and selected witnesses. Each count links to its case and exact form; the finite feasibility results are recorded as T-125, T-126, and T-127 ([results register](#)). For $n = 51$, the selected exact side is $(16 + 5\sqrt{2})/3$. These are construction upper bounds, with no new local- or global-optimality conclusion.

N	PREVIOUS CEILING	SELECTED CEILING	CREDIT
51	7.7007992354170200	7.6903559372884918	Xu
70	8.8816667570090100	8.8809603717088810	Xu
84	9.7071067811865476	9.6980520605096981	Xu
86	9.8228756555322953	9.8205657300098206	Xu
88	9.8824510304821347	9.8824510304812469	Gupta
102	10.6071746801760512	10.6058286965106059	Xu
103	10.7035167555725421	10.6792320475106793	Xu

<i>N</i>	PREVIOUS CEILING	SELECTED CEILING	CREDIT
105	10.8060778655197047	10.7906765754107907	Xu
108	10.9099400734448775	10.9048247851109049	Xu
123	11.6009077785163406	11.5913781457115914	Xu
126	11.7733036066072403	11.7426406872117427	Xu
127	11.8228756555323000	11.8109366475118110	Xu
129	11.8793752067111287	11.8720298492118721	Xu
130	11.9044830325168772	11.9044830325157865	Gupta
131	11.9549168302161245	11.9511500449119512	Xu
146	12.6009077785130200	12.5837822775125838	Xu
153	12.8796793733329640	12.8796793733293146	Gupta
154	12.9265622458535390	12.9265622458523470	Gupta
175	13.7781745930520300	13.7688992766137689	Xu
179	13.8837954905121866	13.8837954905108985	Gupta
180	13.9176534174514757	13.9176534174501843	Gupta
199	14.6175721735980400	14.6175721735928069	Gupta
207	14.8879922583077482	14.8879922583026574	Gupta
208	14.9245187720328190	14.9245187720293559	Gupta
209	14.9496179522017921	14.9496179522003981	Gupta
236	15.8678008394255397	15.8678008394199166	Gupta
237	15.9036762351906287	15.9036762351891381	Gupta
238	15.9261468570124710	15.9261468570109784	Gupta
239	15.9493131697291908	15.9493131697276962	Gupta
261	16.6829268292683000	16.6787798754166788	Xu
267	16.8466671928434900	16.8388319611168389	Xu
295	17.7071067811865500	17.7042327915177043	Xu

Xu identifies the contributor of the complete configurations. The source reports LLM assistance but leaves the exploration algorithm unspecified. All 25 rational certificates have complete passing feasibility replays; 17 supply selected bounds, and the separate radical $n = 51$ witness supplies the eighteenth. The other eight rational inputs remain verified historical witnesses ([Xu packet](#)).

Gupta identifies the precision refiner. Construction credit remains with Nate Chaoweeraprasit’s SQUISH, and the source credits Evan Daniel’s optimizer. All 17 certificates, including the three withdrawn offers at $n = 108, 123, 129$, have complete passing replays. Fourteen supply selected bounds; withdrawal affects selection, not their established feasibility ([Gupta packet](#)). The gains in Gupta’s displayed ceilings are roughly 10^{-12} to 10^{-11} , whereas Xu’s changes in the table range from about 10^{-3} to 3×10^{-2} . This contrast separates precision refinement from the larger gains in the submitted configurations; it does not independently establish novelty or reconstruct discovery.

The rational packets use two deciding geometry implementations with shared certificate parsing, half-angle conversion, rational arithmetic, and separating-axis methodology. Their agreement and full-roster duplicate-square and outside-container controls support the stated finite feasibility checks. They are not two independent mathematical methods, nor a replay of the authors’ optimizers.

REPLAYED CERTIFICATES AND PENDING ADOPTION

A successful geometry replay and a change to the standing register are separate events. Couzo’s eight exact refinements and five follow-up refinements have complete passing positive and negative-control replays, while T-128 and T-130 remain reported entries pending confirming integration and preservation of displaced source houses. The follow-up refines Xu’s $n = 84, 86, 105, 175$ witnesses and Daniel’s $n = 270$ witness; its safe $n = 105$ ceiling is 10.7893037837481589, below the selected value in the table. That smaller certificate is useful evidence about refinement even before standing adoption ([eight-case packet](#); [five-case packet](#)).

Daniel’s 9 October $n = 132, 155$ certificates have complete feasibility replays. Both positive jobs passed both rational geometry implementations, which also rejected all four duplicate-square and outside-container controls. The $n = 132$ certificate has safe sixteen-place ceiling 11.9870993322450633, about 4.23×10^{-3} below the selected bound. It is recorded as T-131, with independent review and standing-bound adoption pending ([record-hunt packet](#); [results register](#)).

The $n = 155$ certificate has exactly the side of Couzo’s pending T-128 certificate, with safe ceiling 12.9524989440140074. The two share 152 of 155 exact poses. The producer labels the three differing squares as free or having flat motion; a connecting motion and equivalence of their local minima have not been verified. Daniel’s certificate supplies equal-bound evidence to T-128, whose adoption remains pending ([exact comparison](#)).

Pending source claims need their own scope. Daniel's retained $n = 105, 130, 292$ supplement, T-129, is an unreplayed historical source report. Exact-form, KKT, or local-minimum claims attached to a previous configuration do not transfer to a newly adopted packing at the same count ([intake requests](#); [results register](#)).

RARE HITS AND SCOPED FAILURES

A basin can be rare under a particular search. Ellsworth's retained $n = 51$ statistics describe an RTX 3080 Ti running 65,536 GPU search threads. Of 3,004 categorized annealer outcomes, only four refined to the record pattern. The source estimates 4.917 GPU hours per hit on its tuned RTX 3080 Ti setup. This measures that implementation's sampling and refinement, rather than the geometric volume of a basin or a portable cost for another search ([run statistics](#)).

Negative experiments help distinguish missing exploration from weak refinement. This repository's fixed-side shrink-and-re-anneal attempts, using single-square moves, failed to escape the grid. Simultaneous all-square perturbations substantially improved $n = 10, 11, 17$, and 26 , but returned the grid unchanged at $n = 29, 37, 50$, and 52 . At $n = 17$, the median side across five seeds per arm fell from 5.000000 under the single-square control to 4.707376 with collective moves, at the same declared pair-test budget. Both remained above the 4.675530 incumbent; neither arm beat a record ([collective-move experiment](#)). An isotropic spread penalty favored a round cloud rather than the square container and regressed proved controls. Basin hopping over the LP quench improved ordinary test cells but reached no record; it also exposed the need to repair floating-point LP outputs monotonically before calling them packings ([retained experiments and discussion](#)). The workbench's force-and-shake physics experiment likewise reached no record in the reviewed runs; its closest repaired result, at $n = 5$, remained about 0.15% above the record after 39,871 seeds. These runs used the previous-count record, the target side, and rigid blocks matched from both records. The quoted result followed harness repair and checking; the report does not retain the final poses as an independent certificate ([measured physics results](#)).

A useful control is to ask whether the local refiner recovers an unperturbed record before interpreting kicked trials. Daniel's reported $n = 292$ calibration found that loosening the container by 2% before a soft squeeze lost the endpoint even with no kick; the subsequent SLP hit its time limit above the starting side. Removing the loosening step recovered that endpoint, after which perturbed trials could be compared with it. These are author-reported calibration runs, not independently reproduced experiments or a local-optimality proof ([calibration table](#)).

At the $n = 11$ contact kink, the tested Powell and Nelder–Mead angle searches performed worse than finite-difference descent. That is a result about those tested angle optimizers, not an experiment on differentiable simulation ([exp-006](#)). A failed search at declared settings and budget constrains a method claim; it does not prove that a family of methods cannot work or that the incumbent packing is locally optimal.

Vocabulary and Further Reading

TERM	MEANING HERE
Pose	a square’s centre and orientation
Seed	the starting configuration supplied to a search
Separating branch	one axis and order that can establish a pair’s non-overlap
Cell	one separating branch selected for every pair
Contact graph	which pairs touch; feature-level equations require more information
Basin	starting configurations grouped by their endpoint under a specified refiner
Quench or polish	local refinement toward an endpoint
Surgery	a seed transformation such as deletion, addition, grafting, or carving
Witness	a container side and complete square poses
Certified upper bound	a side at which feasibility has been rigorously established

A useful record names the finder, seed source and parent count, seed transformation, exploration algorithm, local refiner, certificate method and verifier, certified side, publication date, and source revision. Each method or lineage field should distinguish author-reported mechanics from independently reproduced steps; independently checked feasibility describes the output. Missing fields remain unknown. The register’s single `construction_method` field has coarse labels such as `simulated-annealing`, `inflation-billiard`, and `unknown` ([field vocabulary](#)). Recording seed, exploration, refinement, and certification separately would retain the distinctions in these hybrid workflows.

The [tutorial](#) develops the mathematical model, cell and basin distinctions, exact reconstruction, and optimality proofs. The [strategy catalogue](#) indexes search families and their record histories; the [frontier register](#) records bounds, evidence, and attribution by count. The retained [annealing survey](#) and [simulation survey](#) provide deeper mechanism comparisons and their source limitations. For the other half of the problem, [the lower-bound explainer](#) shows how a certificate can exclude every packing below a specified side.

Version History

- **v0.2.1 — October 10, 2026.** Use typographic punctuation in version histories and consistent Chicago title case for paper titles and links.
- **v0.2.0 — October 10, 2026.** Add the confirmed October 8 intake: 32 selected upper-bound improvements, Couzo’s disclosed refinement workflow, Xu’s radical certificate, and refiner calibration and branch-selection examples, with five attributed packing illustrations. Include Daniel’s replayed October 9 record-hunt certificates, pending independent review and standing-bound adoption. Distinguish local rigidity, source assertions, and translation screens from optimality evidence.
- **v0.1.0 — October 8, 2026.** First publication: a systematic tutorial on modeling, searching, refining, reconstructing and verifying record square packings, with sourced method and seed lineage tables.

The Squares Project · github.com/jlevy/squares
Formatted and typeset with [Flowmark](#) and [KPress](#)