hexagon:2611.01188v1

Compatible local replacements for shifted lonely runners

Chengyi Fang
ORCID 0009-0006-1237-7763

27 November 2026

Abstract

Write $1/(n+1+E_n)$ for the infimal covering radius of n shifted runners with distinct positive integer speeds. Combining three rules of Daria Poliakova and selecting compatible phases at shared speeds gives

$$E_n \geq \max\left\{\lfloor n/125\rfloor,\ \left\lfloor \frac{36071n-11671970}{4337929}\right\rfloor\right\} \quad (n\geq 1), \qquad \liminf_{n\to\infty} E_n/n \geq \frac{577511}{69406489}.$$

Sixteen exact recipes yield a recursive schedule. One gives 476 distinct speeds at radius at most $1/481$. Lean verifies that configuration and the integer counting step for the first bound. We also give a finite decision procedure for the local affine class with arbitrary positive odd added speeds and real phase parameters.

1 Covering radius and local replacements

Let $\|x\| = \min_{k\in\mathbb{Z}} |x-k|$. For distinct positive integers $v_1,\dots,v_n$ and phases $s_i \in \mathbb{R}/\mathbb{Z}$, put

$$\gamma(v,s) = \max_{t\in\mathbb{R}/\mathbb{Z}} \min_{1\leq i\leq n} \|v_i t + s_i\|, \qquad \Gamma(n) = \inf_{v,s} \gamma(v,s), \qquad E_n = \frac{1}{\Gamma(n)} - (n+1).$$

A radius d covers the time circle when the intervals on which some runner has distance at most d cover $\mathbb{R}/\mathbb{Z}$. The stationary observer is excluded from n. A single runner covers a fraction $2d$ of the circle for $0\leq d\leq 1/2$, so $\Gamma(n)\geq 1/(2n)$. Blanco, Criado and Santos proved $\Gamma(n) < 1/(n+1)$ for $5\leq n\leq 17$ [1, Theorem 1.8]. Poliakova constructed counterexamples for every $n\geq 95$ and proved $E_n \geq \lfloor n/287\rfloor$ [2, Theorems 1 and 2]. Her strongest coefficient in Table 1 is $1/224$. The construction below improves the quantitative bound by combining her rules across scales and accounting for overlaps. Zhang's question about every $n\geq 5$ and the global minimizers [3, Question 8.5] also motivates exact computations at individual sizes [4].

Set $\delta = 1/(N+1)$. The zero-phase speeds $2,4,\dots,2N$ give a δ-cover. Of the $N+1$ points $0, 2t, \dots, 2Nt$ modulo one, two coincide or two successive points in circular order have separation at most δ. Their index difference supplies a runner covering t. Call the smallest covering speed responsible for t, following Poliakova [2, Definition 5].

A rule consists of a set R of deleted even speeds and a table P of speeds v with prescribed phases $a_v + h_v\delta$. The even entries of P are rephased seed speeds and the odd entries are additions. An even entry may be prescribed the identically zero phase. The table has distinct speeds and $R\cap P = \varnothing$, where a speed in P means a speed in its domain. The affected even set is $Q = R\cup P_{\text{even}}$, the support is $Q\cup P$, and the gain is $|R| - |P_{\text{odd}}|$. At an odd scale b, multiply each speed by b and retain its phase formula. Such an application fits as soon as $N \geq b\max Q/2$.

Lemma 1 (Primitive-residue test)**.** *For the rule to cover every time for which a speed it affects was responsible in the even seed, for every N and admissible odd scale, it is necessary and sufficient that the intervals*

$$\left[\frac{q}{v}(-1-h_v),\ \frac{q}{v}(1-h_v)\right], \qquad v \in P, \quad vk/q + a_v \in \mathbb{Z}, \tag{1}$$

cover $[-1,1]$ for every $q \in Q$ and $0 \le k < q$ with $\gcd(k, q/2) = 1$.

Proof. If qb is responsible at t, write $qbt = k + \delta z$ with $k \in \mathbb{Z}$ and $|z| \le 1$. An integer divisor $r > 1$ common to k and $q/2$ would give a smaller even seed speed qb/r with distance at most δ/r. Thus the residue of k modulo q is among those tested. For each admissible v in (1), $vbt + a_v + h_v\delta$ differs from an integer by $\delta((v/q)z + h_v)$. This proves sufficiency.

For necessity take $b = 1$ and fix any tested q, k and any $z \in [-1,1]$. At k/q each smaller even speed has positive distance from the integers by the coprimality condition. At $t = (k + \delta z)/q$ they still have distance greater than δ for all sufficiently large N, whereas q covers. Each replacement with $vk/q + a_v \notin \mathbb{Z}$ also has distance greater than δ there for sufficiently large N. For an aligned replacement and sufficiently small δ, the distance is exactly $\delta|(v/q)z + h_v|$. Locality forces one aligned replacement's interval to contain z. □

A collection of rules with disjoint supports composes. At each time, an unaffected responsible speed still covers, and an affected one has a witness of its own rule in the combined configuration. There is also an overlapping composition. Take the union of all prescribed speeds when each shared speed has the same affine phase (the constant term is read modulo one). Remove only those speeds listed for deletion that are absent from the prescriptions. Every affected speed was affected by some member rule, and all witnesses of that rule survive. Hence its local coverage survives. We count the gain after the union and the removal of duplicates. A half-turn translation in time preserves the zero phases of the even seed and adds $1/2$ to the phase of each odd speed. It preserves locality for the translated rule, since responsibility for the seed is unchanged. This translation acts in the same way after multiplication by any odd scale.

2 Recipes with competing phase prescriptions

Use rule A from Poliakova's Theorem 7 and rules B, C from the $19 \to 17$ and $24 \to 21$ rows of her Table 1 [2]. For the exact data below, $[X]_{a,h}$ prescribes phase $a + h\delta$ at each speed in X. All additions are odd.

$$\begin{aligned}
R_A &= \{8, 32, 192\},\\
P_A &= [\{12, 24, 48, 96\}]_{1/2,\,1/2} \cup [\{16, 64\}]_{0,\,-1} \cup [\{3, 9\}]_{1/4,\,0},\\
R_B &= \{8, 32, 36, 40, 72, 180, 192, 240\},\\
P_B &= [\{12, 24, 48, 60, 96, 108, 120\}]_{1/2,\,1/2} \cup [\{16, 64, 80\}]_{0,\,-1} \cup [\{216\}]_{0,\,-2}\\
&\quad \cup [\{3, 9, 27\}]_{1/4,\,1/4} \cup [\{15, 45, 135\}]_{3/4,\,1/4},\\
R_C &= \{4, 16, 20, 36, 40, 64, 72, 240, 384, 432\},\\
P_C &= [\{12, 24, 48, 96, 192\}]_{1/2,\,1/2} \cup [\{60, 108, 120, 216\}]_{1/2,\,-1/2}\\
&\quad \cup [\{8, 128\}]_{0,\,-1} \cup [\{80, 144\}]_{0,\,1} \cup [\{32\}]_{0,\,-3/2}\\
&\quad \cup [\{3, 45, 81\}]_{1/4,\,-1/4} \cup [\{5, 9\}]_{1/4,\,1/4} \cup [\{15, 27\}]_{1/4,\,0}.
\end{aligned}$$

A recipe starts with several odd dilations of these tables. Reflection $t \mapsto -t$ negates both phase coefficients. A translation by half a turn changes the constant phase of an odd speed by $1/2$ and

leaves the even seed in place. These operations specify the candidate prescriptions at each speed. The recipe chooses at most one prescription at any speed. It removes a specified subset of the proposed deletions and installs a prescription at every proposed deletion it keeps. A speed can thus be deleted even if a different table proposed a phase for it. At a shared odd speed it installs only one runner.

After these choices we verify the combined recipe directly by the local condition in Lemma 1. The verification is necessary because choosing one of two competing prescriptions may remove a witness used by a constituent rule. The affected even set is the union of the actual deletions and all installed even prescriptions. Every other even speed remains at phase zero. The gain is the number of actual deletions minus the number of installed odd speeds.

All affected q for the recipes below are divisible by four. If $\gcd(k, q/2) = 1$, then k is odd and coprime to q. Every constant phase has denominator dividing four. An aligned track must therefore satisfy $q \mid 4v$. With this divisibility, alignment depends only on k mod 4, and the alignment condition is

$$(4vk + (4a_v)q) \equiv 0 \pmod{4q}.$$

One can test $k = 1, 3$ for each affected q. The accompanying independent replay enumerates all primitive numerators by this congruence, forms the rational intervals in (1), and verifies that their closed union covers $[-1, 1]$. In both checks the witnesses are drawn from the installed prescriptions. The affected sizes and their certified gains are

Gain	1	2	3	4	5	6	7	8
Seed size	96	120	216	480	600	864	1080	2400
Gain	9	10	11	12	13	14	15	16
Seed size	3000	4320	5400	5832	7776	12000	15000	21600.

The indicated sizes suffice for the specific recipes, and each recipe also works for larger seed sizes. Appendix A gives the choices of copies, deletions and phases.

At seed size 480, the recipe removes 11 even speeds, rephases 17 others and adds 7 odd speeds. For example, it keeps speed 240 at a nonzero constant phase. Speed 120 with label $(1/2, -1/2)$ meets its old zero-phase obligation. For odd k, $k/2 + 1/2$ is integral and $|z/2 - 1/2| \leq 1$ on $[-1, 1]$. The recipe gives 476 distinct positive integer speeds at radius at most $1/481$. A direct rational sweep over the time circle checks that same configuration independently of the local residue calculation. The Lean theorem `MathLab.ScaleRule.certificate476` checks the literal configuration and proves the cover for every real time.

3 Counting the savings

A dilation of any recipe by $c \geq 1$ coprime to 30 preserves the ratio in (1). At an affected scaled speed qc, the primitive numerator is still odd because q is divisible by four. The two checked residues therefore give exactly the witnesses needed under dilation. All unscaled affected speeds and odd additions are supported on the primes $2, 3, 5$. Removing these prime factors from a scaled speed recovers c. Thus recipes assigned to distinct cores c have disjoint supports and compose simultaneously.

Let H be the first thirteen seed sizes in the displayed table and let $B = (12000, 15000, 21600)$ be the last three, counted with their positions in the gain sequence. Multiplying any recipe by five frees the speeds in A_1, A_9, A_{81}, where A_s denotes rule A at dilation s. Those three copies are pairwise disjoint and prime to five, have total gain three and fit by size 7776. In fact all A_{9^i} have mutually

disjoint supports. The only speed of A divisible by nine is 9, and speed 1 is absent. Hence no speed in the support of A is nine times another speed in that support, nor can a larger power of nine give a collision.

Starting with each recipe in B, repeated multiplication by five and addition of the three copies gives three more units at each successive row of thresholds 5^jB, $j \geq 0$. Indeed the three copies fit by $5t$ for $t \in B$, and the last threshold in B comes before the first one in $5B$. Thus a single core with relative seed size x can attain the count

$$g(x) = \#\{t \in H : t \leq x\} + \sum_{j \geq 0} \#\{t \in B : 5^j t \leq x\}.$$

The recipes attaining different counts can be different. A core uses one recipe for its current count.

Write $C_m(y) = \#\{1 \leq c \leq y : \gcd(c, m) = 1\}$. The resulting saving at global seed size N is at least

$$G(N) = \sum_{t \in H} C_{30}(N/t) + \sum_{t \in B} C_6(N/t). \tag{2}$$

The second term absorbs the powers of five, since each integer coprime to six is uniquely 5^jc with c coprime to 30. If $G(n+h) \geq h$, the construction with $N = n+h$ leaves at most n runners. Padding by unused positive integer speeds preserves the cover at radius $1/(n+h+1)$. Therefore $E_n \geq h$.

Theorem 2. *For every $n \geq 1$,*

$$E_n \geq \max\left\{ \lfloor n/125 \rfloor,\ \left\lfloor \frac{36071n - 11671970}{4337929} \right\rfloor \right\}. \tag{3}$$

Moreover,

$$\liminf_{n \to \infty} \frac{E_n}{n} \ \geq \ \frac{577511}{69406489}. \tag{4}$$

Proof. For $1 \leq r < m$ coprime to m, the progression $t(r + jm)$, $j \geq 0$, contributes $\lfloor (N + t(m - r))/(mt) \rfloor$ to $C_m(N/t)$. The first 1029 integers h, starting at zero, satisfy $G(126h) \geq h$ by direct integer evaluation. Advancing h by 1029 shifts the argument by $126 \cdot 1029$. For each progression, floor superadditivity gives an increment of at least $\lfloor 126 \cdot 1029/(mt) \rfloor$. The sum is 1038, so $G(126h) \geq h$ for every h. With $h = \lfloor n/125 \rfloor$, $n + h \geq 126h$. This proves the first finite bound. The Lean theorem `MathLab.ScaleCounting.all_sizes` verifies this integer induction.

For the second finite bound, the periods mt of the progression indicators divide their least common multiple $P = 87480000$. Each period has gain 721420, hence slope $p/q = 36071/4374000$. The maximal discrepancy

$$K = \max_{N \geq 0} \{pN - qG(N)\} = 16045969$$

occurs at $N = 4239239$ where $G(N) = 34956$. It is enough to check the integers immediately before the hits in the progressions in one period, and both period endpoints. On each intervening interval the discrepancy increases, and the discrepancy is periodic since $pP = q \cdot 721420$. The exact replay merges the sorted progressions and obtains the displayed maximum. For $h = \lfloor (pn + q - 1 - K)/(q - p) \rfloor > 0$, one has

$$qG(n + h) \geq p(n + h) - K \geq q(h - 1) + 1.$$

Integrality gives $G(n + h) \geq h$. The zero-phase seed proves the bound when $h \leq 0$.

To obtain (4), improve the per-core construction. For $x < 38880$ use the recipe giving $g(x)$. For $x \geq 38880$, use the improved construction at $x/5$, dilate it by five, and add every A_{9^i} that fits at x. The supports are disjoint by the preceding argument. For $x \geq 38880 = 5 \cdot 7776$ the head H

fits at both x and $x/5$, and all of the first triple B fits at x. Thus $g(x) = 3 + g(x/5)$ in this range. The first three A_{9^i} always fit there, and the next first fits at $96 \cdot 9^3 = 69984 > 38880$. It follows by induction along the recursion that the improvement is one additional unit at each threshold $96 \cdot 9^i 5^j$, $i \geq 3$, $j \geq 0$. For every fixed finite subset of these thresholds the density of counted cores at each one is $4/15$. Taking increasingly many of them gives a lower limit of at least

$$\alpha = \frac{36071}{4374000} + \frac{4}{15} \sum_{i \geq 3,\ j \geq 0} \frac{1}{96 \cdot 9^i 5^j} = \frac{577511}{69984000}$$

for the saving divided by N. For $0 < a < \alpha$, all sufficiently large seed sizes save at least aN. Taking $h = \lfloor an/(1-a) \rfloor$ and $N = n + h$ then gives $h \leq aN$. Letting a increase to α proves (4). □

For example, at $n = 100000$ the two terms of (3) give 800 and 828, respectively. The second yields a covering radius at most $1/100829$. The comparison is with earlier constructive bounds for the same quantity. Poliakova's Theorem 1 gives $E_n \geq \lfloor n/287 \rfloor$, the largest coefficient for a single rule in her Table 1 is $1/224$, and the first MathLab release obtained $\lfloor n/150 \rfloor$ and the coefficient $513079/69470921$ [2, 6]. Poliakova's separate construction already gives $E_n \geq 1$ for $n \geq 95$. The new bounds concern the same all-size extremal quantity, with no new size in the open interval $58 \leq n \leq 94$.

4 A finite decision problem for affine local rules

Fix an even integer $L \geq 2$ and a target gain $r \geq 1$. Consider exactly the class of Lemma 1, in which the affected even speeds do not exceed L, the added speeds are positive odd integers, and each phase is affine in δ with coefficients independent of N. `aadrianvv` used SAT to test finite speed and phase menus for this kind of local condition, and carried out fixed-size patch searches on the even seed [5].

Proposition 3. *The existence of such a rule of gain at least r is decidable. With L in unary and r given as an integer, the decision problem belongs to NP.*

Proof. Fix the affected set Q of any viable rule and let $M = \operatorname{lcm}(Q)$. The number of odd additions is at most $L/2 - r$. Within each residue class modulo M compress the added speeds, in order, to the smallest distinct positive members of that class. Since M is even, these speeds are odd. Replacing v by $v' \leq v$, keep a_v and replace the slope by $h_v v'/v$. Alignment for every tested q, k is unchanged, while the absolute value of the error expression in (1) is multiplied by v'/v. Thus all obligations remain covered and all added speeds are at most $(L/2 - r)M$.

Discard unused additions and restore unused even prescriptions to phase zero. The latter can only remove affected rows. Each remaining constant phase appears in some alignment, so it has denominator dividing the original M when normalized modulo one. Write $u_v = -h_v/v$. On each tested row with affected speed q the eligible intervals in the coordinate z/q have centers u_v and radii $1/v$, and must cover $[-1/q, 1/q]$. Choose a simple intersecting chain of eligible intervals whose union covers the target. Such a chain exists by restricting the interval graph to intervals meeting the target and connecting one that contains each endpoint. Every chain constraint is a difference inequality on centers or on one center and the reference $u_0 = 0$, with bound a sum of signed terms from $1/q, 1/v$. We can discard or restore any prescriptions unused by these chains.

Let D be the lcm of the original Q together with the remaining prescribed speeds. Scale all centers by D. The right-hand sides are integers, so flooring every coordinate preserves every selected inequality, since $x_i \leq x_j + c$ implies $\lfloor x_i \rfloor \leq \lfloor x_j \rfloor + c$ for integral c. The reference stays zero. Each

originally used center satisfies $|u_v| \le 1/2 + 1/v$ because its chain interval meets a target segment. The resulting certificate has $u_v \in D^{-1}\mathbb{Z}$, with $|u_v| \le 1/2 + 1/v + 1/D$.

One can encode the certificate using the universal M equal to $\operatorname{lcm}(2, 4, \ldots, L)$ and the corresponding larger grid. By a product bound its bit length is $O(L \log L)$. There are at most $L/2$ prescribed speeds, so $\log D = O(L^2 \log L)$. The checker tests the gain and distinctness and verifies at most $\sum_{q \le L} q$ interval unions with polynomial bit complexity. No chain choices need be stored. □

A complete search without scanning a phase grid. Include every even $q \le L$ in the rows, including retained zero-phase speeds. A row is indexed by $r_0 = k/q$ from Lemma 1 and has target $[-1/q, 1/q]$. Identical fractions can be merged by keeping the largest target interval. A retained even speed with $a_v = h_v = 0$ covers each of its own rows. Conversely any cover of these rows by at most $L/2 - r$ runners gives a rule as in Lemma 1, with each unused even speed deleted. Declare all kept even speeds through L to be prescribed, including the ones with zero phase. The assigned witnesses all lie in the finite cover.

Start each row at its left endpoint. Suppose some rows are unfinished, and pick a row with the smallest covered frontier f. In any completion some unused eligible runner covers an interval immediately to the right of f on this row, because the existing covered segments outside the frontier are empty in this starting and updated representation. A finite completion has a member covering an entire right neighborhood of f, with left endpoint at most f. Place this runner's left endpoint at f, preserving its width $2/v$ and its alignment on the chosen row. The position requires $a_v = -vr_0 \bmod 1$ and $h_v = -1 - vf$. It loses only points strictly to the left of f, already covered on every unfinished row where the runner was eligible. Eligibility on another row s_0 is equivalent to $\operatorname{den}(s_0 - r_0) \mid v$. On eligible rows we advance the frontier to $f + 2/v$ if this is an increase, capped at the right endpoint. These advances maintain connected coverage from the original left endpoint since the leftmost frontier was selected. On every unfinished row, all previously placed intervals met its original or then-current covered segment whenever they extended it, so they leave no covered points beyond its current frontier. Branching on the unused eligible speeds gives soundness and completeness by induction on the remaining runner budget.

Only finitely many speeds need be branched on. Put $M = \operatorname{lcm}(2, 4, \ldots, L)$. Two odd speeds with the same $\gcd(v, M)$ allow precisely the same row incidences after alignment at a common row, since each relevant denominator divides M. Within any class choose at most the first $L/2 - r$ positive speeds. To see completeness, replace the speeds of a completion by the smallest distinct speeds in that class, sorted in matching order. Align each replacement at an original served row and retain its interval's left endpoint. The interval grows and all incidences are preserved. Even speeds are left unchanged.

The exact search `local_pivot.py` finds no cover of these rows with $L/2 - 1$ runners for $L = 2, 4, 6, 8, 10, 12$. At $L = 12$ it uses 24 rows and 26 candidate speeds. Its pruning is also exact. At any step a set of pairwise incompatible active rows requires as many remaining runners as there are rows in the set. When just one runner remains, it must align with every active row and its right endpoint must reach the largest active target endpoint. This excludes positive gain through $L = 12$ in the stated class. The cutoff is a limit on local even speeds, independent of the global number of runners.

Verification and attribution

The recipe choices are finite certificates, not claims of optimality over every possible mixture. The intervals in this proof are checked by rational arithmetic. The discrepancy and the low-cutoff

exclusions use exact finite enumeration. The Lean development verifies a generic local transfer result, the 476-runner example and the integer all-size counting inequality under the displayed progression data. The identification of those progressions with a physically realizable schedule, the larger discrepancy bound and the limiting-layer argument use the written proof and exact arithmetic checkers.

Poliakova devised the three rules and their original application at disjoint odd dilations. The first MathLab release combined copies whose shared phases were already compatible [6]. This revision checks recipes that choose from competing phases, and uses the resulting sixteen recipes to improve the finite and asymptotic bounds. The finite decision theorem is retained from the first release. The prior SAT experiments on bounded phase menus are credited to `aadrianvv` [5].

Chengyi Fang initiated and directed the work. OpenAI Codex led proof development, integration and verification. GPT-6.2 Sol subagents helped with mathematical audits and publication. GPT-6.2 Astra supplied derivations and independent review through the Responses API. AI systems wrote the proofs, code and manuscript. Chengyi Fang has not independently checked the proofs.

The article is licensed under CC BY 4.0. The original accompanying software uses the MIT License.

A Recipe certificates

Each copy is specified by its rule letter, odd dilation, reflection bit and half-turn bit. The deletion list gives the speeds actually removed. At a surviving speed with multiple distinct labels, the list records the selected $(4a, 4h)$. Otherwise the unique surviving label is used. All other evens in the seed retain phase zero. These data suffice to reconstruct every interval union in Section 2. The primitive-center check is to require $4v$ divisible by q, retain arcs satisfying $4vk + (4a)q \equiv 0 \bmod 4q$, and sweep the resulting closed intervals in increasing order of left endpoint.

Gain 1, seed size 96.

```
Copies: A(1,0,0)
Deleted: 8, 32, 192
Ties: none
```

Gain 2, seed size 120.

```
Copies: B(1,0,0)
Deleted: 8, 32, 36, 40, 72, 180, 192, 240
Ties: none
```

Gain 3, seed size 216.

```
Copies: C(1,0,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 240, 384, 432
Ties: none
```

Gain 4, seed size 480.

```
Copies: A(1,0,0), C(1,0,1), A(5,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 160, 384, 432, 960
Ties: 3:(3,-1), 9:(3,1), 45:(3,-1)
```

Gain 5, seed size 600.

```
Copies: A(1,0,0), C(1,0,1), B(5,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 160, 180, 200, 360, 384, 432, 900, 960, 1200
Ties: 3:(3,-1), 9:(3,1), 15:(3,-1)
```

Gain 6, seed size 864.

```
Copies: A(1,0,0), C(1,0,1), A(3,1,0), B(5,1,0), A(9,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 160, 180, 200, 288, 360, 384, 900, 960, 1200, 1728
Ties: 3:(3,-1), 9:(3,0), 15:(3,-1), 48:(2,2), 81:(3,-1), 144:(0,4), 192:(2,2)
```

Gain 7, seed size 1080.

```
Copies: C(1,0,0), A(3,0,0), B(5,1,1), B(9,1,1), C(5,1,0)
Deleted: 4, 16, 20, 36, 64, 72, 80, 100, 180, 200, 288, 320, 324, 360, 384, 648, 900, 1200,
    1620, 1728, 1920, 2160
Ties: 9:(1,0), 15:(3,1), 27:(1,-1), 45:(3,-1), 48:(2,2), 75:(3,-1), 135:(3,-1), 144:(0,4),
    192:(2,2), 225:(3,-1), 300:(2,-2), 400:(0,4), 405:(3,-1), 540:(2,-2), 600:(2,-2),
    720:(0,4), 1080:(2,-2)
```

Gain 8, seed size 2400.

```
Copies: A(1,0,0), C(1,0,1), A(3,1,0), B(5,1,0), A(9,1,0), C(5,0,0), B(15,0,1), C(9,0,1),
    A(25,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 200, 288, 324, 384, 540, 648, 800, 1920, 2160,
    2700, 2880, 3456, 3600, 3888, 4800
Ties: 3:(3,-1), 9:(3,0), 15:(3,-1), 27:(3,-1), 45:(3,-1), 48:(2,2), 60:(2,-2), 75:(1,-1),
    81:(3,-1), 108:(2,-2), 120:(2,-2), 135:(3,-1), 144:(0,4), 192:(2,2), 216:(2,-2),
    225:(1,-1), 240:(2,-2), 405:(3,-1), 432:(2,-2), 480:(2,-2), 675:(1,-1), 720:(2,2),
    864:(2,-2), 960:(2,2), 1080:(2,-2), 1200:(2,-2)
```

Gain 9, seed size 3000.

```
Copies: C(1,0,0), A(3,0,0), B(5,1,1), B(9,1,0), C(5,0,1), C(9,0,0), B(25,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 200, 288, 324, 360, 384, 648, 800, 900,
    1000, 1620, 1800, 1920, 2160, 3456, 3888, 4500, 4800, 6000
Ties: 9:(1,0), 15:(1,-1), 27:(1,-1), 45:(1,-1), 48:(2,2), 60:(2,-2), 75:(3,-1), 81:(1,-1),
    108:(2,-2), 120:(2,-2), 135:(1,-1), 144:(0,4), 192:(2,2), 216:(2,-2), 240:(2,-2),
    243:(1,0), 405:(1,-1), 432:(2,-2), 480:(2,-2), 864:(2,-2), 1080:(2,-2), 1944:(2,-2)
```

Gain 10, seed size 4320.

```
Copies: C(1,0,0), A(3,0,0), B(5,1,1), B(9,1,0), C(5,0,1), C(9,0,0), B(25,1,0), A(45,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 200, 288, 324, 360, 384, 648, 800, 900,
    1000, 1440, 1620, 1800, 1920, 3456, 3888, 4500, 4800, 6000, 8640
Ties: 9:(1,0), 15:(1,-1), 27:(1,-1), 45:(1,-1), 48:(2,2), 60:(2,-2), 75:(3,-1), 81:(1,-1),
    108:(2,-2), 120:(2,-2), 135:(1,-1), 144:(0,4), 192:(2,2), 216:(2,-2), 240:(2,-2),
    243:(1,0), 405:(1,-1), 432:(2,-2), 480:(2,-2), 864:(2,-2), 1080:(2,-2), 1944:(2,-2)
```

Gain 11, seed size 5400.

```
Copies: C(1,0,0), A(3,0,0), B(5,1,1), B(9,1,0), C(5,0,1), C(9,0,0), B(25,1,0), B(45,1,1)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 200, 288, 324, 360, 384, 648, 800, 900,
    1000, 1440, 1620, 1800, 1920, 3240, 3456, 3888, 4500, 4800, 6000, 8100, 8640, 10800
Ties: 9:(1,0), 15:(1,-1), 27:(1,-1), 45:(1,-1), 48:(2,2), 60:(2,-2), 75:(3,-1), 81:(1,-1),
    108:(2,-2), 120:(2,-2), 135:(1,-1), 144:(0,4), 192:(2,2), 216:(2,-2), 240:(2,-2),
    243:(1,0), 405:(1,-1), 432:(2,-2), 480:(2,-2), 864:(2,-2), 1080:(2,-2), 1944:(2,-2),
    5400:(2,-2)
```

Gain 12, seed size 5832.

```
Copies: C(1,0,0), B(5,1,1), B(9,0,0), B(25,1,1), B(45,1,1), C(25,1,0), C(27,1,1)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 288, 360, 384, 400, 500, 900, 960, 972,
    1000, 1440, 1600, 1620, 1728, 1800, 1944, 4500, 6000, 6480, 8100, 8640, 9600, 10368,
    10800, 11664
Ties: 15:(1,-1), 27:(1,0), 75:(3,-1), 81:(1,-1), 108:(2,2), 135:(1,-1), 144:(0,-4), 216:(2,2),
    225:(3,-1), 243:(1,-1), 375:(3,-1), 405:(1,-1), 540:(2,2), 675:(3,-1), 720:(0,-4),
    864:(2,2), 1080:(2,2), 1125:(3,-1), 1215:(1,-1), 1500:(2,-2), 2000:(0,4), 2025:(3,-1),
    2160:(2,-2), 2700:(2,-2), 3000:(2,-2), 3600:(0,4), 5400:(2,-2)
```

Gain 13, seed size 7776.

```
Copies: C(1,0,0), B(5,1,1), B(9,1,0), B(25,1,1), B(45,1,1), C(25,1,0), A(81,1,1)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 288, 324, 360, 384, 400, 500, 648, 900,
    960, 1000, 1440, 1600, 1620, 1728, 1800, 2592, 3240, 4500, 6000, 8100, 8640, 9600, 10800,
    15552
Ties: 15:(1,-1), 27:(1,0), 75:(3,-1), 81:(1,-1), 225:(3,-1), 243:(1,0), 375:(3,-1),
    675:(3,-1), 1080:(2,-2), 1125:(3,-1), 1500:(2,-2), 1944:(2,-2), 2000:(0,4), 2025:(3,-1),
    2700:(2,-2), 3000:(2,-2), 3600:(0,4), 5400:(2,-2)
```

Gain 14, seed size 12000.

```
Copies: C(1,0,0), B(5,1,1), B(9,1,1), B(25,1,1), B(45,1,1), C(25,1,0), A(81,1,1), C(45,0,0),
    A(125,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 288, 324, 360, 384, 400, 500, 648, 900,
    960, 1000, 1440, 1600, 1620, 1728, 1800, 2592, 3240, 4000, 4500, 8100, 9600, 10800, 15552,
    17280, 19440, 24000
Ties: 15:(1,-1), 27:(1,-1), 75:(3,-1), 135:(1,-1), 225:(3,-1), 243:(1,-1), 375:(3,-1),
    405:(1,-1), 540:(2,-2), 675:(3,-1), 1080:(2,-2), 1125:(3,-1), 1215:(1,-1), 1500:(2,-2),
    1944:(2,-2), 2000:(0,4), 2025:(3,-1), 2160:(2,-2), 2700:(2,-2), 3000:(2,-2), 3600:(0,4),
    4320:(2,-2), 5400:(2,-2), 9720:(2,-2)
```

Gain 15, seed size 15000.

```
Copies: C(1,0,0), B(5,1,1), B(9,1,1), B(25,1,0), B(45,1,1), C(25,1,0), A(81,1,1), C(45,0,0),
    B(125,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 288, 324, 360, 384, 400, 500, 648, 900,
    960, 1000, 1440, 1600, 1620, 1728, 1800, 2592, 3240, 4000, 4500, 5000, 8100, 9000, 9600,
    10800, 15552, 17280, 19440, 22500, 24000, 30000
Ties: 15:(1,-1), 27:(1,-1), 75:(3,-1), 135:(1,-1), 225:(3,-1), 243:(1,-1), 375:(3,-1),
    405:(1,-1), 540:(2,-2), 675:(3,-1), 1080:(2,-2), 1125:(3,-1), 1215:(1,-1), 1500:(2,-2),
    1944:(2,-2), 2000:(0,4), 2025:(3,-1), 2160:(2,-2), 2700:(2,-2), 3000:(2,-2), 3375:(3,-1),
    3600:(0,4), 4320:(2,-2), 5400:(2,-2), 9720:(2,-2)
```

Gain 16, seed size 21600.

```
Copies: C(1,0,0), B(5,1,1), B(9,1,1), B(25,1,0), B(45,1,1), C(25,1,0), A(81,1,1), C(45,0,0),
    B(125,1,0), A(225,1,0)
Deleted: 4, 16, 20, 36, 40, 64, 72, 100, 160, 180, 288, 324, 360, 384, 400, 500, 648, 900,
    960, 1000, 1440, 1600, 1620, 1728, 1800, 2592, 3240, 4000, 4500, 5000, 7200, 8100, 9000,
    9600, 15552, 17280, 19440, 22500, 24000, 30000, 43200
Ties: 15:(1,-1), 27:(1,-1), 75:(3,-1), 135:(1,-1), 225:(3,-1), 243:(1,-1), 375:(3,-1),
    405:(1,-1), 540:(2,-2), 675:(3,-1), 1080:(2,-2), 1125:(3,-1), 1215:(1,-1), 1500:(2,-2),
    1944:(2,-2), 2000:(0,4), 2025:(3,-1), 2160:(2,-2), 2700:(2,-2), 3000:(2,-2), 3375:(3,-1),
    3600:(0,4), 4320:(2,-2), 5400:(2,-2), 9720:(2,-2)
```

References

[1] M. Blanco, F. Criado and F. Santos, *Coloopless zonotopes and counterexamples to the Shifted Lonely Runner Conjecture*, arXiv:2603.24784v4, 2026.

[2] D. Poliakova, *More (shifted) runners, less loneliness*, arXiv:2609.23952v1, 2026.

[3] Y. Zhang, *Tight instances of the Lonely Runner Conjecture: one-entry modifications and the growth of speeds*, arXiv:2608.13599v3, 2026.

[4] `aadrianvv` and `tomdif`, *Shifted Lonely Runner counterexamples for* $18 \leq n \leq 57$, solved-conjectures, issue 110, 2026.

[5] `aadrianvv`, Finite-menu uniform rule search and fixed-size structured patch search, *solved-conjectures*, issue 110, 21 November 2026.

[6] C. Fang, *Compatible local replacements for shifted lonely runners*, MathLab v1.0.0, 26 November 2026.