# THE ENUMERATION TABLE OF RECTANGULAR WEIGHING DESIGNS $W(m, z, k)$: THE STARRED CELLS SETTLED EXACTLY, AND FOUR PRINTED VALUES CORRECTED

KOREA SUPERINTELLIGENCE LABS

Abstract. A weighing design $W(m, z, k)$ in the sense of Lejeune Herman and Goos (arXiv:2608.04814) is an $m \times z$ matrix over $\{0, \pm 1\}$ with exactly $k$ non-zero entries in every column, at most $m - k$ zeros in every row, and $W^{\mathsf{T}} W = kI_z$. Their Table 1 counts the isomorphism classes (row and column permutations, row and column sign changes) for $4 \le m \le 24$ with two zeros per column when $m$ is even and three when $m$ is odd; the entries marked $\star$ were obtained by a partial enumeration and are lower bounds. We re-enumerate the whole table by canonical augmentation with a canonical form in place of the pairwise isomorphism tests of the source, at a total cost of 2.8 processor-hours against the 9.6 days the source reports for one extension step. Of the 111 completely enumerated cells, 106 reproduce exactly. The other five are corrections: the cell $(5, 3)$ holds one class, not three; the $m = 23$ row is $5, 22, 119$ at $z = 3, 4, 5$, not $2, 4, 59$, and it does not terminate at $z = 5$ but continues with 459 and 1834 classes at $z = 6, 7$; and $(24, 4)$ holds 77 classes, not 72. All 25 starred cells of the rows $m = 16$ and $m = 18$ and of the cells $z \le 7$ $(m = 20)$, $z \le 6$ $(m = 22)$, $z = 5$ $(m = 24)$ are settled exactly; every exact value exceeds the printed bound, by a factor of up to 1806, and the $m = 18$ row runs four columns further than the table prints. Two of the exact values — at the square cells $(16, 16)$ and, past the printed row, $(18, 18)$, which are the classical $W(16, 14)$ and $W(18, 16)$ — agree with counts announced by Lampio and Ganzhinov in 2025 and are replications; for the rest we found no prior enumeration. For 26 cells a refuting statement is verified in Lean 4: an explicit list of designs, each checked against the definition, pairwise separated by a proved isomorphism invariant, and longer than the printed entry — at $(23, 6)$ and $(23, 7)$, where the table prints nothing, the list refutes the termination mark instead; at $(5, 3)$ the exhaustive half is verified as well.

## 1. Introduction

Lejeune Herman and Goos [1] construct large orthogonal minimally aliased response surface designs by folding over and concatenating *weighing designs*, the rectangular generalisation of weighing matrices in which the number of columns may be smaller than the number of rows. The combinatorial core of their paper is an enumeration: for every pair $(m, z)$ with $4 \le m \le 24$, and with $k = m - 2$ for even $m$ and $k = m - 3$ for odd $m$, they list the isomorphism classes of weighing designs $W(m, z, k)$ column by column and print the class counts in their Table 1. (Table and section numbers of [1] are those of the arXiv PDF of version 1, which we also rebuilt from the e-print with tectonic; the two agree on every section, table and appendix number and on the pages of Tables 1 and 2, the rebuild running one page later than the arXiv PDF from Appendix A on. The table carries the label `tab:rc-single` in the source and sits on page 16 of both.) The legend of that table, read off the same page, is the following. Grey rows are even $m$, "each column contains exactly two zeros"; white rows are odd $m$, "each column contains three zeros". A superscript † means "no further orthogonal designs exist that meet the specified constraints", a superscript ‡ "there are no more valid zero-pattern extensions allowing a larger design", and

> "A superscript $\star$ indicates that the corresponding number of designs was obtained through partial rather than complete enumeration."

The subscripts # and $\diamond$ are quality flags of the resulting response-surface designs and carry no counting content. The partial enumeration (Section 4.1.2 and Appendix D of [1]) keeps, at each extension step, only a top fraction of the classes found so far, ranked by a design-quality criterion, and extends only those; the retention percentages are printed in the source's Table 4 and go down to 0.01%. A starred entry is therefore a lower bound on the number of classes, not a count, and the source says so. Section 4.1.1 of [1] explains why the enumeration stops where it does:

> "In cases where the computation became infeasible, the procedure did not complete all iterations even after several days."

Its Table 2 gives the incremental runtimes of the complete enumeration, among them 9.6 days for the single step from $z = 5$ to $z = 6$ at $m = 16$, 11.6 hours for $z = 3 \to 4$ at $m = 24$, and 3 hours for $z = 2 \to 3$ at $m = 23$.

The reason for these runtimes is the isomorph rejection. The source extends every representative of a $(z-1)$-class by every admissible column and then, in the words of its Section 3.2, adds a candidate to the collection "if they satisfy the orthogonality condition $M'M = kI_z$ … and if they are not isomorphic to any matrix already in" the collection, isomorphism being decided by a graph-isomorphism call on a coloured $\pm$-bipartite encoding (its Appendix C). That is one isomorphism test per (candidate, stored class) pair, quadratic in the size of the answer. Replacing the pairwise test by a *canonical form* — one canonical-labelling call per candidate and a hash lookup — is the standard remedy (McKay's isomorph-free generation [12], in the form described in Kaski–Östergård [14]), and it makes the step linear in the number of candidates. With it, the whole of Table 1 at the levels reached below costs 2.76 processor-hours on one core, and the whole $m = 16$ row, out to $z = 16$, takes 24 seconds.

This paper reports what that re-enumeration finds, and what part of it is verified by a proof assistant.

- **Replication** (Section 3). Of the 111 cells of Table 1 that are printed without a star, 106 reproduce exactly; 18 of the 21 rows reproduce cell for cell as far as they are printed complete, and every †/‡ termination claim holds except one.
- **Four corrections and one false termination** (Section 4). The cell $(5, 3)$ prints 3; it holds exactly one class (Theorem 4.1, verified in both directions). The $m = 23$ row prints $2, 4, 59^{\ddagger}$ at $z = 3, 4, 5$; the counts are $5, 22, 119$, and designs $W(23, 6, 20)$ and $W(23, 7, 20)$ exist, 459 and 1834 classes of them, so the ‡ is false (Theorems 4.2 and 4.3). The cell $(24, 4)$ prints 72; it holds 77 classes (Theorem 4.6).
- **The starred cells** (Section 5). The ten starred cells of the $m = 16$ row, the nine of the $m = 18$ row, and the six cells $(20, 5..7)$, $(22, 5..6)$, $(24, 5)$ are settled exactly; the $m = 18$ row continues to $z = 18$ where the table prints nothing after $z = 14$. Every exact value exceeds the printed bound, as the legend requires. For 19 of these cells (all but $(16, 11..16)$), and for the five corrected cells above, an explicit list of pairwise non-isomorphic designs larger than the printed value is verified in Lean 4.
- **Cost** (Section 6). The measured per-level costs, the comparison with the source's Table 2, and what remains out of reach at the per-cell budget used here.

**What is new and what is not.** The objects are the source's own, including the row condition "each row contains at most $m - k$ zeros", which is part of its definition (Section 2 of [1]) and is enforced in its algorithm as "Exclusion criterion 2"; without it the counts are those of a different object. The column-by-column recursion is the source's (its Appendix B proves that it reaches every class), and the coloured $\pm$-graph encoding of a signed matrix is its Appendix C. Canonical augmentation, canonical labelling and colour refinement are textbook [12, 13, 14]. What is claimed here is the content of the cells: the corrections and the exact values at the starred cells, and the machine-checked certificates that a printed value is too small (or, at $(5, 3)$, too large).

Seven of the levels reached below are *square* ($z = m$; $m = 4, 6, 8, 10, 12, 16, 18$), and there the object is a classical weighing matrix $W(m, m-2)$ and the classical literature applies. The first four hold one class each and lie inside the range of the classical work listed below. At $(12, 12)$ the value 5 is the published count of $W(12, 10)$ (Harada–Munemasa [8], Table 4). At $(16, 16)$ and $(18, 18)$ the values 17 and 4 are the counts of $W(16, 14)$ and $W(18, 16)$, and these two are *not* new: they are announced, with exactly these numbers, in a classification of square weighing matrices by Lampio and Ganzhinov [9] ("we classify weighing matrices of orders 16, 17, 18, 19, 20 and 21 for all weights"), presented at the 8th Hadamard Workshop in May 2025 and, at the time of writing, unpublished — its own summary slide says "[t]hese results are preliminary". Their results slide gives 17 at order 16 weight 14 and 4 at order 18 weight 16, in both cases without a reference, i.e. as new there. We report the agreement: these two values are replications, reached independently and by a different route. Nothing earlier gives either count: Harada–Munemasa [8] exclude order 16 and reach order 18 only at weight 9, and the closest thing to a table at these parameters, Appendix A of Chan–Rodger–Seberry [4] ("Inequivalent $W(n, k)$ for $n \leq 20$"), records "$\geq 1$" in its count column at both $(16, 14)$ and $(18, 16)$. For the rectangular levels ($z < m$) with $m - k \in \{2, 3\}$ — everything else below — we searched for a prior enumeration and found none. The source's own literature review (its Section 2) lists the classifications of *square* weighing matrices — Chan, Rodger and Seberry [4], whose own abstract says that "[t]he inequivalent weighing matrices are classified for $k \leq 5$" inside the range $1 \leq k \leq n \leq 20$ ([1] describes it as order $\leq 20$ and weight $< 5$), Ohmori [5, 6] and Ohmori–Miyamoto [7] for $W(14, 8)$, $W(13, 9)$, $W(17, 9)$, Harada–Munemasa [8] for the orders $n \leq 17$ *except* $n = 16$, and beyond that only the parameters $(16, 6)$, $(16, 9)$, $(16, 12)$, $(18, 9)$, Araya et al. [11] for unbiased weighing matrices of weight 9 — and the enumerations of Núñez Ares–Goos [2] (orthogonal minimally aliased designs, complete to five columns) and Schoen–Eendebak–Goos [3] (conference designs, i.e. exactly one zero per column); none covers $m - k \in \{2, 3\}$ with $z < m$. A full-text search of 890,835 arXiv e-prints for "weighing design" fires on 129 lines, 70 of them in the source itself and 18 in a paper on fMRI designs built from circulant square weighing matrices, the rest passing mentions; among the 5385 lines mentioning "weighing" in the 154 e-prints that mention weighing matrices or weighing designs, none contains any of the exact counts reported below; the OpenAlex index showed nothing citing the source on 2026-09-07; and OEIS queries for several complete rows of the table and for the settled $m = 16$ and corrected $m = 23$ rows return nothing. These searches see e-prints, indexed articles and integer sequences; they do not see a conference talk, which is how [9] was found and is a reason to read the negative as "not found" rather than "new".

**Verification.** Every theorem below rests on declarations checked in Lean 4 against *Mathlib*; their statements are reproduced verbatim in Appendix A and Section 7 records the modules, the axioms and the cost. The formal content of a lower-bound theorem is exactly this: an explicit list of $N$ matrices, each verified to satisfy Definition 2.1, whose isomorphism invariants (Definition 2.4) are pairwise distinct, hence which are pairwise non-isomorphic by Proposition 2.5. The complementary half — that no class is missing — is formalised only at $(5, 3)$. Everywhere else the exact counts are the output of the enumeration, cross-checked as described in Section 2.2, and are labelled *Computation*, not *Theorem.*

## 2. Preliminaries

### 2.1. The objects.

**Definition 2.1** (weighing design; [1], Section 2, verbatim)**.** "A weighing design $W(m, z, k)$ is defined as a matrix of size $m \times z$ with entries from $\{0, \pm 1\}$, in which each column contains exactly $k$ non-zero entries and each row contains at most $m - k$ zeros. [...] The columns are required to be mutually orthogonal, which is expressed by the condition $W'W = kI_z$."

For $z = m$ every row of such a matrix has exactly $m - k$ zeros and the object is a classical weighing matrix $W(m, k)$ ([15, 17]); $k = m - 1$ gives conference matrices and $k = m$ Hadamard matrices. Table 1 of [1] is the two-parameter array indexed by $(m, z)$, with $k = m - 2$ for even $m$ and $k = m - 3$ for odd $m$; those are the only two values of $m - k$ that occur, and we write $(m, z)$ for a cell of it. The row condition is easy to overlook (a first description of the problem we worked from omitted it) and is not decorative: the $5 \times 4$ matrix with columns $(1, 1, 0, 0, 0)$, $(1, -1, 0, 0, 0)$, $(0, 0, 1, 1, 0)$, $(0, 0, 1, -1, 0)$ has four pairwise orthogonal columns of weight 2 and is rejected only because its last row carries four zeros, more than $m - k = 3$; this is the reason no $W(5, 4, 2)$ exists (the † at $(5, 3)$, Theorem 4.1(iv)).

**Definition 2.2** (isomorphism; [1], Section 2)**.** Two $m \times z$ matrices over $\{0, \pm 1\}$ are *isomorphic* if one can be transformed into the other by a sequence of row permutations, column permutations, and sign changes of whole rows or whole columns.

The group acting is $(S_m \ltimes \{\pm 1\}^m) \times (S_z \ltimes \{\pm 1\}^z)$ modulo the global sign; the four kinds of move each preserve Definition 2.1, so it acts on the set of weighing designs $W(m, z, k)$, and Table 1 counts its orbits. The source's Section 2 also fixes a normal form — "the first column of the design is fixed. The first $m-k$ entries are set to zero and the last $k$ entries are set to one", and "the first row is standardized so that its entries can only take the values 0 or 1" — but that costs nothing: a row permutation and row sign flips bring any design's first column to that shape, and sign flips of the columns $2, \ldots, z$ then fix the first row without disturbing the first column. So the normal form loses no class, and the source is counting the same orbits. The formal development (Appendix A) carries a design as the list of its rows, and defines $\mathrm{Iso}_z$ as the equivalence relation generated by three constructors — a permutation of the list of rows, a negation of any subset of the rows, and a *column move* $(\pi, \varepsilon)$ with $\pi$ a permutation of $\{0, \ldots, z-1\}$ and $\varepsilon \in \{\pm 1\}^z$ sending row $r$ to $(\varepsilon_t \, r_{\pi(t)})_{t<z}$ — closed under symmetry and transitivity. This is Definition 2.2.

### 2.2. Canonical augmentation.

If $W$ is a $W(m, z, k)$ and $z \geq 2$, deleting any column leaves a $W(m, z-1, k)$: the column conditions are column-local, orthogonality is inherited, and the number of zeros in a row can only decrease. Hence every isomorphism class at $z$ columns restricts, on its first $z-1$ columns, to a class at $z-1$ columns, and extending one representative of every $(z-1)$-class by *every* admissible column and identifying isomorphic results yields exactly the $z$-classes. This is the recursion of the source (its Appendix B); the only difference is how "identifying isomorphic results" is done.

*The column search.* For a parent $A$ with $z-1$ columns and a chosen zero pattern (a $k$-subset of the rows respecting the row bound), the sign vector must be orthogonal to each column of $A$; a parity prefilter (each column of $A$ must meet the support in an even number of positions) discards most patterns, and the survivors are solved by meet-in-the-middle on two halves of the support. A column $c$ and $-c$ give isomorphic designs, so the first support entry is pinned to $+1$.

*Two canonical forms, cross-checked.* (1) The coloured graph $G(M)$ of the source's Appendix C: vertices $R_i^{\pm}$ (colour 0) and $C_j^{\pm}$ (colour 1), twin edges $R_i^+ R_i^-$ and $C_j^+ C_j^-$, and for $M_{ij} = +1$ the edges $R_i^+ C_j^+$, $R_i^- C_j^-$, for $M_{ij} = -1$ the edges $R_i^+ C_j^-$, $R_i^- C_j^+$. Its canonical labelling is computed by nauty 2.8.9 [13], and the canonical matrix is read off the labelling (rows in order of first appearance of a twin pair, the earlier vertex of the pair taking the + role), so it is a function of the canonical graph and hence a complete invariant. (2) A direct form needing no graph library: normalise every row to $\min(r, -r)$ in the lexicographic order, sort the rows, and minimise over the $z!\, 2^{z-1}$ column permutations and column sign patterns. Form (2) is the default for $z \leq 3$ and form (1) above; the two were run against each other on the complete rows $m = 17$ and $m = 19$ with identical results. An all-nauty re-run of the $m = 23$ row was started for this paper and did not finish: at $z = 2$ that row's designs have very large automorphism groups, and its 55.5 million canonical labellings are far slower than the 26–30 microseconds of Section 6. A third, recursion-free census (walk every $z$-tuple of admissible columns, bucket by form (2)) agrees on every small cell it reaches, including $(5, 3)$ (Section 4).

*Two independent enumerators.* For this paper the generator was re-run on every row with $m \leq 19$, on $m = 21$, 23, 24 (to $z = 4$), and two further enumerators were written that share no enumeration code with it. The first, in Python, extends columns by meet-in-the-middle and rejects isomorphs by form (2); it reproduces every cell with $m \leq 15$ and $z \leq 5$ (45 cells, no mismatch). The second, in C, again does canonical augmentation, but with its own column search (zero patterns first, then a meet-in-the-middle sign solve) and its own two canonical forms: a graph-free minimisation over all $z!\, 2^z$ column moves after row normalisation, used at $z \leq 4$, and a canonical labelling of the $\pm$ graph above that. Nauty is the only thing it shares with the generator, and its two forms were run against each other on complete rows to $z = 6$ ($m = 7, 8, 10, 12, 13$) with identical counts. It reproduces, cell for cell: every row with $m \leq 15$, including $(5, 3) = 1$; the whole $m = 16$ row, out to $z = 16$ and empty at $z = 17$; the rows $m = 17$, 19, 21; the whole $m = 23$ row, $2, 5, 22, 119, 459, 1834$ and empty at $z = 8$ — and a second pass with the graph-free form raised to $z = 5$ returns $(23, 3) = 5$, $(23, 4) = 22$ and $(23, 5) = 119$ again, so all three corrected cells of that row stand with no graph library involved at all; the whole $m = 18$ row, out to $z = 18$ and empty at $z = 19$; $m = 20$ through $z = 6$, $m = 22$ through $z = 5$, and $m = 24$ through $z = 4$, where it too returns $(24, 4) = 77$ against the printed 72 — and that run also uses the graph-free form throughout; and the empty level that follows every † and ‡ of Table 1 except the one at $(23, 5)$. That covers all five corrected cells and 22 of the 25 starred cells of Table 2 — all ten of the $m = 16$ row, all nine of the $m = 18$ row (with its four unprinted columns), $(20, 5)$, $(20, 6)$ and $(22, 5)$ — leaving only $(20, 7)$, $(22, 6)$ and $(24, 5)$ outside its reach. Its per-level candidate counts agree with the generator's to the digit at every level of every row both reached. The invariant-reach numbers of Computations 4.4, 5.3 and 5.7 were recomputed from its representatives and agree. Independently of both enumerators, the 25 design lists shipped in Lean were parsed out of the Lean sources and re-checked against Definition 2.1 and, for pairwise non-isomorphism, against each canonical form separately — so once with no graph library at all. Under both, in every cell, all listed designs are weighing designs and the number of classes equals the shipped length, so every "at least $N$" below holds independently of the formal development.

**Lemma 2.3** (correctness of the graph encoding)**.** *A colour-preserving isomorphism $G(M) \to G(M')$ exists if and only if $M$ and $M'$ are isomorphic in the sense of Definition 2.2.*

*Proof.* The twin edges are the only edges inside a colour class, so a colour-preserving isomorphism $\varphi$ maps twin pairs to twin pairs: $\varphi(R_i^s) = R_{\sigma(i)}^{s\epsilon_i}$ and $\varphi(C_j^t) = C_{\tau(j)}^{t\delta_j}$ for permutations $\sigma, \tau$ and signs $\epsilon_i, \delta_j$. The edge condition "$M_{ij} = st$ iff $M'_{\sigma i,\tau j} = st\,\epsilon_i\delta_j$" says exactly $M'_{\sigma i,\tau j} = \epsilon_i\delta_j M_{ij}$, i.e. $M'$ is obtained from $M$ by the row permutation $\sigma$, the column permutation $\tau$ and the sign changes $\epsilon, \delta$. Conversely such data define a colour-preserving isomorphism. (This is the argument the source gives informally in its Appendix C; it is not part of the formal development, which never relies on the graph.) □

2.3. **A proved isomorphism invariant.** The formal certificates do not use canonical forms at all. They use an invariant that is cheap to evaluate and whose invariance has a short proof.

**Definition 2.4.** Let $D$ have rows $r_1, \dots, r_m \in \mathbb{Z}^z$. The *profile* of a row is the sorted multiset $p(r_i) = \{|\langle r_i, r_j\rangle| : 1 \le j \le m\}$ (the term $j = i$ is the number of non-zero entries of $r_i$). The *one-round invariant* is the sorted multiset $\kappa_z(D) = \{p(r_1), \dots, p(r_m)\}$, and the *two-round invariant* is $\kappa_{2z}(D) = \{p_2(r_1), \dots, p_2(r_m)\}$ where $p_2(r_i) = \{(|\langle r_i, r_j\rangle|, p(r_j)) : 1 \le j \le m\}$ sorted. In the formal development each sorted multiset is packed into a natural number by a positional encoding (`invKey`, `invKey2` in Appendix A); injectivity of the packing is irrelevant for what follows, since only the direction "distinct value ⇒ not isomorphic" is used.

**Proposition 2.5** (the invariant, and the shape of a certificate)**.** *Let $z \ge 1$ and let $D, D'$ be lists of rows.*

- (i) *If* $\mathrm{Iso}_z(D, D')$ *then* $\kappa_z(D) = \kappa_z(D')$ *and* $\kappa_{2z}(D) = \kappa_{2z}(D')$ (`invKey_of_iso`, `invKey2_of_iso`)*.*
- (ii) *If $L$ is a list of designs whose values of $\kappa_z$ (resp. $\kappa_{2z}$) are pairwise distinct, then no two members of $L$ are related by* $\mathrm{Iso}_z$ (`pairwise_not_iso`, `pairwise_not_iso2`)*.*
- (iii) *Controls: the $4 \times 2$ matrix with rows $(0, 0), (0, 0), (1, 1), (1, -1)$ is a $W(4, 2, 2)$; changing its last row to $(1, 1)$ (columns no longer orthogonal) or to $(0, -1)$ (first column of weight $1$) gives a matrix that is not one; the $4 \times 3$ matrix with rows $(0, 0, 0), (0, 0, 1), (1, 1, 0), (1, -1, 1)$ is not a $W(4, 3, 2)$; and both invariants separate the two classes of $W(4, 2, 2)$, so neither is constant.*

*Proof.* (i) is an induction over the three generators of $\mathrm{Iso}_z$. A row permutation permutes the outer multiset and each inner one; a sign change of a row changes each $\langle r_i, r_j\rangle$ involving it by a sign, which the absolute value removes; a column move $(\pi, \varepsilon)$ preserves every $\langle r_i, r_j\rangle$ because $\pi$ is a permutation of the coordinates and $\varepsilon_t^2 = 1$ (`dotZ_applyRow`). The two-round invariant is handled by the same three arguments applied to pairs. (ii) is immediate from (i). (iii) is decided by evaluation. The packaged form used by every lower-bound theorem below is `lower_bound_of_cert` (`…2` for the two-round invariant): from "every member of $L$ satisfies Definition 2.1" and "the invariants of $L$ are pairwise distinct", both discharged by evaluation, it returns "every member of $L$ is a $W(m, z, k)$ and $L$ is pairwise non-isomorphic". The distinctness test sorts the invariants and checks strict increase, so it is $n \log n$ rather than quadratic. One caveat about the last control: its $4 \times 3$ matrix fails Definition 2.1 twice over — its first row carries three zeros, more than $m - k = 2$, and its first and third columns have inner product 1 — so what is proved is that the predicate rejects it, not that the row clause alone does. The example that isolates the row clause is the $5 \times 4$ matrix after Definition 2.1, whose columns *are* pairwise orthogonal; it is not in the formal development. □

*Remark* 2.6. For a cell $(m, z)$ the theorem shipped is, in the notation of Appendix A,

$$\exists L, \quad |L| = N \ \wedge\ (\forall D \in L,\ \texttt{IsWD}\ m\ z\ k\ D) \ \wedge\ L \text{ pairwise } \neg\mathrm{Iso}_z,$$

with $L$ read out of one string literal of $N\,m\,z$ characters in column-major order. The designs in $L$ are the first $N$ representatives produced by the enumeration whose invariants are distinct, and $N$ always exceeds the printed value, which is what makes the theorem a refutation of it. At the starred cells $N$ is exactly one more than the printed value; at the corrected cells it is the exact count where the invariant separates all of it (5, 22 and 77 at $(23, 3)$, $(23, 4)$ and $(24, 4)$) and as much of it as the invariant separates otherwise (111 of 119 at $(23, 5)$); at $(23, 6)$ and $(23, 7)$, where nothing is printed, five suffice. The exact counts are reported separately and are never smaller.

## 3. The completely enumerated cells

Before any new value is claimed, every cell of Table 1 printed without a star was recomputed. Table 1 lists the outcome; it is a computation, not a theorem, and it is what licenses reading the disagreements of Section 4 as errors rather than as a difference of definitions.

| $m$ | this enumeration $(z = 2, 3, \dots)$ | Table 1 of [1] | |
|---|---|---|---|
| 4 | 2 1 1 | 2 1 1 | agree |
| 5 | 2 **1** | 2 **3**$^\dagger$ | **differ** |
| 6 | 2 2 2 1 1 | 2 2 2 1 1 | agree |
| 7 | 2 4 2 1 1 1 | 2 4 2 1 1 1 | agree |
| 8 | 2 3 6 3 3 1 1 | 2 3 6 3 3 1 1 | agree |
| 9 | 2 4 4 2 | 2 4 4 2$^\ddagger$ | agree |
| 10 | 2 3 8 7 8 4 3 1 1 | same | agree |
| 11 | 2 5 7 5 2 1 | 2 5 7 5 2 1$^\dagger$ | agree |
| 12 | 2 4 19 38 81 61 61 27 20 5 5 | same | agree |
| 13 | 2 5 9 8 2 | 2 5 9 8 2$^\dagger$ | agree |
| 14 | 2 3 21 94 216 15 4 | 2 3 21 94 216 15 4$^\dagger$ | agree |
| 15 | 2 5 12 19 11 10 | 2 5 12 19 11 10$^\ddagger$ | agree |
| 16 | 2 4 37 352 3244 | 2 4 37 352 3244 | agree |
| 17 | 2 5 14 26 15 4 | 2 5 14 26 15 4$^\ddagger$ | agree |
| 18 | 2 3 37 884 | 2 3 37 884 | agree |
| 19 | 2 5 17 51 66 71 | 2 5 17 51 66 71$^\ddagger$ | agree |
| 20 | 2 4 57 | 2 4 57 | agree |
| 21 | 2 5 19 69 135 198 | 2 5 19 69 135 198$^\ddagger$ | agree |
| 22 | 2 3 52 | 2 3 52 | agree |
| 23 | 2 **5** **22** **119** 459 1834 | 2 **2** **4** **59**$^\ddagger$ | **differ** |
| 24 | 2 4 **77** | 2 4 **72** | **differ** |

Table 1. The unstarred cells of Table 1 of [1] against the present enumeration. The starred cells of the rows 16–24 follow in Section 5. Of the 111 unstarred cells, 106 agree; of the ten †/‡ marks, the nine at $m = 5, 9, 11, 13, 14, 15, 17, 19, 21$ hold (the next level is empty) and the one at $(23, 5)$ fails.

Every odd row stops short of $z = m$ except $m = 7$, and the classical theory says exactly why. Geramita, Geramita and Seberry Wallis prove ([10], Propositions 2 and 23) that if $n$ is odd and a $W(n, s)$ exists then $s$ is a perfect square, and that $W(n, k)$ with $n$ odd requires $(n-k)^2-(n-k)+2 > n$. On the odd rows of Table 1 the weight is $k = m - 3$, so $m - k = 3$ and the second condition reads $3^2 - 3 + 2 > m$, that is $8 > m$: no $W(m, m-3)$ exists for any odd $m \geq 9$, which is every odd row of the table but $m = 5$ and $m = 7$. At $m = 5$ the first condition finishes the job, $k = 2$ not being a square. At $m = 7$ both conditions hold ($k = 4 = 2^2$ and $8 > 7$), and the row does reach $z = m$: $(7, 7) = 1$ is the unique $W(7, 4)$. The case worth naming is $m = 19$, where $k = 16$ *is* a perfect square, so the first condition says nothing and the second is what excludes $W(19,16)$ — which is why that row stops at $z=7$.

The even rows reach $z=m$ and their last cells are classical weighing matrices: $(4,4)=1$ is $W(4,2)$, $(6,6)=1$ is $W(6,4)$, $(8,8)=1$ is $W(8,6)$, $(10,10)=1$ is $W(10,8)$, $(12,12)=5$ is $W(12,10)$, and the $m=14$ row ends at $z=8$, so no $W(14,12)$ exists. The count at $(12,12)$ can be checked against a published classification, and matches it: Harada–Munemasa [8] give 5 for $W(12,10)$ (their Table 4), and the file their database [16] lists under that name holds five matrices, each a weighing design in the sense of Definition 2.1 with $(m,z,k)=(12,12,10)$, pairwise non-isomorphic, and together with the five representatives found here still giving five classes — so the two lists are the same five classes. The four smaller square cells hold one class each and lie inside the range of the classical work quoted in Section 1, of which we read only the table in Appendix A of [4]; we make no cell-by-cell comparison there.

## 4. Four printed values are wrong, and one termination

### 4.1. **The cell** $(5,3)$.

**Theorem 4.1** ($(5,3)$ holds one class, not three). *Let* $W_0$ *be the* $5\times 3$ *matrix with rows* $(1,1,0)$, $(1,-1,0)$, $(0,0,1)$, $(0,0,1)$, $(0,0,0)$.

(i) $W_0$ *is a weighing design* $W(5,3,2)$ (`W0_isWD`).

(ii) *Every* $5\times 3$ *matrix satisfying Definition 2.1 with* $(m,z,k)=(5,3,2)$ *is isomorphic to* $W_0$ (`cell_5_3_one_class`). *Consequently there are no two non-isomorphic weighing designs* $W(5,3,2)$ (`cell_5_3_no_two`)*, and the number of isomorphism classes is exactly* 1*. Table 1 of* [1] *prints* 3.

(iii) *None of the* $40^4$ *matrices* $[a\,b\,c\,e]$ *whose columns are weight-*2 *vectors of length* 5 *is a weighing design* $W(5,4,2)$ (`cell_5_4_empty`).

(iv) *Hence no* $W(5,4,2)$ *exists, and the* † *printed at* $(5,3)$ *is correct.*

*Proof.* (i) is evaluation. For (ii), a $W(5,3,2)$ has three columns of length 5 with exactly two non-zero entries each, so it is one of the $40^3=64000$ matrices with columns from the set of 40 such vectors; the formal proof establishes this membership as a genuine lemma (`mem_allDesigns53`, by induction on the structure of the candidate vectors), and then sweeps the 64000 candidates: for each one that satisfies Definition 2.1 it finds a column move, a row sign pattern and a row permutation carrying $W_0$ to it, from a finite list of 48 column moves and 32 sign patterns (`sweep`, discharged by evaluation, and `iso_of_isoW0B`, which turns the found data into an $\mathrm{Iso}_3$ derivation). So both halves of "exactly one class" are machine-checked; this is the only cell of Table 1 small enough for that. (iii) is evaluation over $40^4$ candidates. (iv) follows from (iii) by the same membership argument as in (ii), which for four columns is not formalised (the $5\times 4$ example after Definition 2.1 shows the row condition is what excludes them).

The mathematics is elementary and explains why the printed 3 is a mistake rather than a different convention. Two weight-2 columns of length 5 are orthogonal iff their supports are disjoint or equal with anti-matching signs. Three pairwise orthogonal ones cannot have three pairwise disjoint supports (that needs six rows) nor three equal supports (a two-dimensional space holds only two mutually orthogonal $\pm 1$ vectors), so the supports are $\{S,S,T\}$ with $S\cap T=\emptyset$, which is one class. Independently of Lean, the C generator returns 1, and the recursion-free census finds 2880 labelled designs among the 64000 triples, all in one class. □

### 4.2. **The row** $m=23$.

**Theorem 4.2** (the $m=23$ row is understated). *There are at least* 5, 22 *and* 111 *pairwise non-isomorphic weighing designs* $W(23,3,20)$, $W(23,4,20)$ *and* $W(23,5,20)$ *respectively: for each of the three cells an explicit list of that many designs, each verified against Definition 2.1, has pairwise distinct one-round invariants* (`cell_23_3`, `cell_23_4`, `cell_23_5`)*. Table 1 of* [1] *prints* 2, 4 *and* 59 *at these cells, without a star.*

**Theorem 4.3** (the ‡ at $(23,5)$ is false). *Weighing designs* $W(23,6,20)$ *and* $W(23,7,20)$ *exist; for each of the two cells an explicit list of five designs, each verified against Definition 2.1, has pairwise distinct one-round invariants* (`cell_23_6`, `cell_23_7`)*. Table 1 of* [1] *prints* $59^{\ddagger}$ *at* $(23,5)$*, the legend of* ‡ *being "there are no more valid zero-pattern extensions allowing a larger design".*

*Proof.* Both are instances of Proposition 2.5(ii) in the packaged form of Remark 2.6; the lists are the first representatives with distinct invariants in the enumeration output for each cell. Refuting a termination claim needs one design; five are exhibited at each of $z = 6, 7$ so that the refutation does not rest on a single matrix. □

**Computation 4.4** (the exact $m = 23$ row; not formalised)**.** The enumeration gives 2, 5, 22, 119, 459, 1834 classes at $z = 2, \ldots, 7$ and none at $z = 8$, so the row terminates at $z = 8$. The five values 5, 22, 119, 459, 1834 were obtained with the nauty canonical form, form (1) of Section 2.2 at $z \geq 4$ and the direct form (2) below it; form (2), applied afterwards to the stored representatives of each level, confirms them pairwise non-isomorphic — which excludes a split class but not a merged one. Against merging stands the independent re-enumeration of Section 2.2, which reaches this row. The one-round invariant separates 111 of the 119 classes at $z = 5$ (re-checked for this paper on the 119 representatives of the re-run); that is why Theorem 4.2 certifies 111 rather than 119. The two-round invariant separates all 119, and at $z = 6, 7$ the two invariants separate 249 and 1171, resp. 457 and 1719, of the 459 and 1834 classes; the certificates of Theorem 4.3 ship five designs each because five suffice to refute a termination claim.

*Remark* 4.5*.* Two circumstantial facts point to a truncated run rather than a definitional difference. Every other odd row with $m \geq 11$ has 5 classes at $z = 3$ ($m = 11, 13, 15, 17, 19, 21$ all print 5, and all reproduce here), and $m = 23$ is the only one printed as 2. And the source's Table 2 records 3 hours for the step $z = 2 \to 3$ at $m = 23$, 16.6 minutes for $z = 3 \to 4$ and 46.4 minutes for $z = 4 \to 5$, so the row was not abandoned for lack of time.

## 4.3. **The cell** $(24, 4)$**.**

**Theorem 4.6** ($(24, 4)$ holds 77 classes, not 72)**.** *There are at least* 77 *pairwise non-isomorphic weighing designs* $W(24, 4, 22)$*: an explicit list of* 77 *designs, each verified against Definition 2.1, has pairwise distinct two-round invariants* $\kappa_{24}$ (`cell_24_4`)*. Table 1 of* [1] *prints* 72 *at this cell, without a star.*

**Computation 4.7** (not formalised)**.** The exact count is 77: the enumeration returns 77 classes, the cells $(24, 2) = 2$ and $(24, 3) = 4$ of the same row reproduce, and the 77 stored representatives are confirmed pairwise non-isomorphic by the direct canonical form as well. The independent enumerator of Section 2.2 reaches this cell and returns 77 too, from the same 7,217,640 candidate columns, using the graph-free canonical form throughout.

*Remark* 4.8 (a coincidence that would have hidden the correction)*.* The one-round invariant $\kappa_4$ takes exactly 72 distinct values on the 77 representatives — the printed number. Had the certificate been built from $\kappa$ alone, it would have certified "at least 72", consistent with the table, and the discrepancy would have gone unnoticed. The two-round invariant $\kappa_{24}$ separates all 77, and Theorem 4.6 is the one theorem of this paper that needs it. (Re-checked for this paper on the shipped list: 72 distinct values of $\kappa_4$, 77 of $\kappa_{24}$.)

# 5. The starred cells

Table 2 collects the 25 starred cells settled here. In every case the exact value is larger than the printed one, as the legend of $\star$ requires; the ratio "printed/exact" is the share of the classes the source's partial enumeration reached at that cell, and it should not be confused with the retention percentage of the source's Table 4, which is the share of the *previous* level it chose to extend.

The theorems below all have the same form and the same proof — Proposition 2.5(ii) in the packaged form of Remark 2.6, with a list read out of the enumeration output — and we state them without repeating it. In each, "at least $N$" means: an explicit list of $N$ designs, each verified to satisfy Definition 2.1 with the stated $(m, z, k)$, whose one-round invariants are pairwise distinct.

## 5.1. **The row** $m = 16$**,** $k = 14$**.**

**Theorem 5.1.** *There are at least* 2211 *pairwise non-isomorphic weighing designs* $W(16, 7, 14)$ (`cell_16_7`)*. Table 1 prints* $2210^{\star}$*.*

| cell | printed | exact | printed/exact | certified | theorem |
|---|---|---|---|---|---|
| $(16, 7)$ | $2210^\star$ | 8165 | 27.1% | $\geq 2211$ | 5.1 |
| $(16, 8)$ | $1889^\star$ | 9931 | 19.0% | $\geq 1890$ | 5.2 |
| $(16, 9)$ | $1049^\star$ | 8158 | 12.9% | $\geq 1050$ | 5.2 |
| $(16, 10)$ | $821^\star$ | 6316 | 13.0% | $\geq 822$ | 5.2 |
| $(16, 11)$ | $472^\star$ | 3404 | 13.9% | — | Comp. 5.3 |
| $(16, 12)$ | $396^\star$ | 1729 | 22.9% | — | Comp. 5.3 |
| $(16, 13)$ | $147^\star$ | 519 | 28.3% | — | Comp. 5.3 |
| $(16, 14)$ | $67^\star$ | 202 | 33.2% | — | Comp. 5.3 |
| $(16, 15)$ | $11^\star$ | 30 | 36.7% | — | Comp. 5.3 |
| $(16, 16)$ | $7^\star$ | 17 | 41.2% | — | Comp. 5.3 |
| $(18, 6)$ | $3112^\star$ | 21923 | 14.2% | $\geq 3113$ | 5.4 |
| $(18, 7)$ | $3375^\star$ | 114374 | 2.95% | $\geq 3376$ | 5.5 |
| $(18, 8)$ | $1157^\star$ | 114178 | 1.01% | $\geq 1158$ | 5.5 |
| $(18, 9)$ | $375^\star$ | 48044 | 0.78% | $\geq 376$ | 5.5 |
| $(18, 10)$ | $793^\star$ | 25613 | 3.10% | $\geq 794$ | 5.5 |
| $(18, 11)$ | $597^\star$ | 12376 | 4.82% | $\geq 598$ | 5.5 |
| $(18, 12)$ | $318^\star$ | 5838 | 5.45% | $\geq 319$ | 5.5 |
| $(18, 13)$ | $80^\star$ | 2162 | 3.70% | $\geq 81$ | 5.6 |
| $(18, 14)$ | $18^\star$ | 794 | 2.27% | $\geq 19$ | 5.6 |
| $(20, 5)$ | $603^\star$ | 2071 | 29.1% | $\geq 604$ | 5.8 |
| $(20, 6)$ | $5694^\star$ | 167231 | 3.40% | $\geq 5695$ | 5.9 |
| $(20, 7)$ | $2236^\star$ | 4038672 | 0.055% | $\geq 2237$ | 5.9 |
| $(22, 5)$ | $2301^\star$ | 4248 | 54.2% | $\geq 2302$ | 5.10 |
| $(22, 6)$ | $8510^\star$ | 909508 | 0.94% | $\geq 8511$ | 5.11 |
| $(24, 5)$ | $488^\star$ | 7857 | 6.21% | $\geq 489$ | 5.12 |

TABLE 2. The starred cells of Table 1 of [1] settled here. "Exact" is the output of the enumeration (a computation); "certified" is what is verified in Lean (a theorem): an explicit list of that many pairwise non-isomorphic designs. One exact value is a replication: $(16, 16)$ is the classical $W(16, 14)$, and 17 is the count announced by Lampio–Ganzhinov [9] (Computation 5.3).

**Theorem 5.2.** *There are at least* 1890, 1050 *and* 822 *pairwise non-isomorphic weighing designs* $W(16, 8, 14)$, $W(16, 9, 14)$ *and* $W(16, 10, 14)$ *respectively* (`cell_16_8`, `cell_16_9`, `cell_16_10`). *Table 1 prints* $1889^\star$, $1049^\star$, $821^\star$.

**Computation 5.3** (the whole $m = 16$ row; not formalised)**.** The row is $2, 4, 37, 352, 3244$ (complete in the source) followed by 8165, 9931, 8158, 6316, 3404, 1729, 519, 202, 30, 17 at $z = 7, \dots, 16$ and 0 at $z = 17$; the whole row takes 24 seconds. The last value is the number of classical weighing matrices $W(16, 14)$, and it agrees with the 17 announced for that parameter pair by Lampio–Ganzhinov [9]. The six cells $z = 11, \dots, 16$ carry no theorem because the invariants of Definition 2.4 cannot separate the classes there: the numbers of distinct values of $(\kappa, \kappa_2)$ over the representatives are $(4764, 7326)$ at $z = 7$, $(4610, 7198)$ at $z = 8$, $(2727, 4065)$ at $z = 9$, $(1211, 1711)$ at $z = 10$, then $(193, 267)$ at $z = 11$, $(32, 35)$ at $z = 12$, $(4, 4)$, $(2, 2)$, $(1, 1)$, $(1, 1)$ at $z = 13, \dots, 16$. So at $z \leq 10$ the invariant reaches past the printed value and the theorem above certifies one more than it, while from $z = 11$ on it does not even reach the printed value. A certificate that would work at every cell is column-orbit minimality (each shipped design is the minimum of its own orbit under column moves after row normalisation, i.e. form (2) of Section 2.2 is the identity on it), which needs the closure of the column-move group in the formal development; it was priced and not attempted here.

5.2. **The row $m = 18$, $k = 16$.**

**Theorem 5.4.** *There are at least* 3113 *pairwise non-isomorphic weighing designs* $W(18,6,16)$ (`cell_18_6`)*. Table 1 prints* $3112^{\star}$*.*

**Theorem 5.5.** *There are at least* 3376*,* 1158*,* 376*,* 794*,* 598 *and* 319 *pairwise non-isomorphic weighing designs* $W(18,z,16)$ *for* $z = 7,8,9,10,11,12$ *respectively* (`cell_18_7`, …, `cell_18_12`)*. Table 1 prints* $3375^{\star}, 1157^{\star}, 375^{\star}, 793^{\star}, 597^{\star}, 318^{\star}$*.*

**Theorem 5.6.** *There are at least* 81 *and* 19 *pairwise non-isomorphic weighing designs* $W(18,13,16)$ *and* $W(18,14,16)$ *respectively* (`cell_18_13`, `cell_18_14`)*. Table 1 prints* $80^{\star}$ *and* $18^{\star}$*.*

**Computation 5.7** (the whole $m = 18$ row; not formalised)**.** The row is $2, 3, 37, 884$ (complete in the source), then 21923, 114374, 114178, 48044, 25613, 12376, 5838, 2162, 794 at $z = 6, \ldots, 14$, and it does not stop there: 184, 52, 4, 4 at $z = 15, 16, 17, 18$ and 0 at $z = 19$. The whole row takes 320 seconds. Table 1 prints nothing after $z = 14$ in this row and carries no † or ‡ there; the source's Table 4 (Appendix D) marks the step to $z = 15$ with a dash, and the legend of that dash reads

> "A dash indicates a step to which no retention percentage applies. This occurs either when $z > m$, since a weighing design cannot have more columns than rows, or when the designs retained at the previous step admitted no valid one-column extension, so that the enumeration produced no design with $z$ columns."

Here $z = 15 \leq m = 18$, so the second case is the one that applies: the designs the source retained at $z = 14$ admitted no one-column extension. The source does not print how many of its 18 classes at $z = 14$ were retained for that step, nor which; what the dash rules out is the reading that the blank is a termination claim. The row is in fact four columns longer than printed, and its last value, 4, is the number of classical weighing matrices $W(18, 16)$, agreeing with the 4 announced for that parameter pair by Lampio–Ganzhinov [9]. The tail is not certified because the invariants collapse there, as in the $m = 16$ row: the numbers of distinct values of $(\kappa, \kappa_2)$ are $(3, 3)$, $(2, 2)$, $(1, 1)$, $(1, 1)$ over the 184, 52, 4, 4 classes at $z = 15, \ldots, 18$.

### 5.3. The rows $m = 20$, 22, 24.

**Theorem 5.8.** *There are at least* 604 *pairwise non-isomorphic weighing designs* $W(20,5,18)$ (`cell_20_5`)*. Table 1 prints* $603^{\star}$*.*

**Theorem 5.9.** *There are at least* 5695 *and* 2237 *pairwise non-isomorphic weighing designs* $W(20,6,18)$ *and* $W(20,7,18)$ *respectively* (`cell_20_6`, `cell_20_7`)*. Table 1 prints* $5694^{\star}$ *and* $2236^{\star}$*.*

**Theorem 5.10.** *There are at least* 2302 *pairwise non-isomorphic weighing designs* $W(22,5,20)$ (`cell_22_5`)*. Table 1 prints* $2301^{\star}$*.*

**Theorem 5.11.** *There are at least* 8511 *pairwise non-isomorphic weighing designs* $W(22,6,20)$ (`cell_22_6`)*. Table 1 prints* $8510^{\star}$*.*

**Theorem 5.12.** *There are at least* 489 *pairwise non-isomorphic weighing designs* $W(24,5,22)$ (`cell_24_5`)*. Table 1 prints* $488^{\star}$*.*

**Computation 5.13** (not formalised)**.** The exact values are $(20, 5) = 2071$, $(20, 6) = 167231$, $(20, 7) = 4038672$, $(22, 5) = 4248$, $(22, 6) = 909508$ and $(24, 5) = 7857$. The cell $(20, 7)$ carries the largest gap among the cells settled here: the source's partial enumeration reached 2236 of 4,038,672 classes, one in 1806. Three of the six, $(20, 5)$, $(20, 6)$ and $(22, 5)$, were reproduced by the independent enumerator of Section 2.2; the other three come from the same generator and canonical form as the rest and were not re-run for this paper (Section 6 gives their cost). The certified lists of all five theorems above were re-checked here independently of Lean for the definition and for the distinctness of the invariants.

*Remark* 5.14 (a starred cell outside our range that the literature already settles). The starred cells left open here are not settled by this paper, but one of them can be read off a published table. The cell $(20, 20)$, printed $8^{\star}$, is the number of classical weighing matrices $W(20, 18)$: every row of a square $W(20, 20, 18)$ carries exactly $m - k = 2$ zeros. Harada–Munemasa [8] give 53 for $W(20, 18)$ (their table of other orders and weights), and the file their database [16] lists under that name holds 53 matrices, each a weighing design in the sense of Definition 2.1 and pairwise non-isomorphic — so the printed $8^{\star}$ reaches 15% of that cell. The same phenomenon as in Table 2, at a cell this enumeration does not reach.

## 6. Cost, and what was not computed

All enumeration was single-threaded, under `nice`, on a shared 20-core machine whose load stayed between 12 and 24; the times are therefore upper bounds. The cost of one level is the number of candidate columns times the cost of one canonical form. On the graphs that arise here ($2m + 2z \le 88$ vertices) a canonical labelling by nauty takes 26–30 microseconds, so the step $z = 5 \to 6$ at $m = 16$, which is 38127 candidates, takes 1.3 seconds against the 9.6 days of the source's Table 2; the step $z = 3 \to 4$ at $m = 24$ (the cell $(24, 4)$, 7.2 million candidates) took 753 seconds in the re-run for this paper, on a machine under load, against 11.6 hours. Table 3 gives the per-row totals of the run that produced Sections 3–5, obtained by summing the per-level times the generator prints; they add up to 9939 seconds, i.e. 2.76 processor-hours, plus 315 seconds for the all-nauty cross-checks of $m = 17$ and $m = 19$, and the peak resident set of any run was 1.3 GB. For this paper the rows $m \le 19$, 21, 23 and 24 (to $z = 4$) were re-run from the committed source of the generator with a freshly built nauty, single-threaded under `nice` at machine load 9–24; every count agrees. Their wall times were 25 s ($m = 16$, whole row), 5 s ($m = 17$), 342 s ($m = 18$, whole row), 25 s ($m = 19$), 146 s ($m = 21$), 677 s ($m = 23$, whole row) and 2035 s ($m = 24$ to $z = 4$); the differences from Table 3 are the difference between wall time under load and the summed per-level processor times.

| row | seconds | row | seconds |
|---|---|---|---|
| $m \le 15$ (all rows) | 2 | $m = 20$ (to $z = 7$) | 2698 |
| $m = 16$ (whole row) | 24 | $m = 21$ | 134 |
| $m = 17$ | 5 | $m = 22$ (to $z = 6$) | 2370 |
| $m = 18$ (whole row) | 320 | $m = 23$ (whole row) | 674 |
| $m = 19$ | 24 | $m = 24$ (to $z = 5$) | 3688 |

Table 3. Processor time per row of the enumeration reported here.

Three rows were stopped at a measured price rather than at their end. At $m = 20$, the level $z = 7$ cost 2410 seconds for 39.6 million candidates and returned 4,038,672 classes; the next level scales to about $1.8 \times 10^8$ candidates and more than $10^7$ classes to store, and the row runs to $z = 20$. At $m = 22$, the level $z = 6$ cost 1833 seconds for 29.6 million candidates; $z = 7$ extrapolates to about $1.1 \times 10^9$ candidates. At $m = 24$, the level $z = 5$ alone cost 1599 seconds for 18.6 million candidates. The remaining starred cells of Table 1 — $(20, 8..20)$, $(22, 7..14)$, $(24, 6..12)$ — are therefore still lower bounds, and the exact values reported above suggest that the printed ones are far below the truth there too.

The source's quadratic isomorph rejection is not the whole story of its runtimes — its Table 2 records 16.6 hours for the step $z = 1 \to 2$ at $m = 24$, a cell whose answer is 2 and whose 85.7 million candidates take 413 seconds here even with the direct canonical form, which never calls nauty — but it is the part that turns days into seconds at the cells that matter.

## 7. Verification

Theorems 4.1, 4.2, 4.3, 4.6, 5.1–5.2, 5.4–5.6 and 5.8–5.12 and Proposition 2.5 rest on declarations verified in Lean 4 against *Mathlib* [18, 19] (Lean `v4.32.0`, Mathlib at its `v4.32.0` tag), whose statements are reproduced verbatim in Appendix A. The development is fourteen modules. `Defs` (375 lines) defines the design predicate `IsWD` of Definition 2.1, the relation `Iso` of Definition 2.2, the invariant `invKey`, proves its invariance and the packaged certificate, and holds five controls; `Refine` adds the two-round invariant `invKey2` and its invariance; `Cell53` holds Theorem 4.1; and eleven generated modules (`Row23`, `Row24`, `Frontier16`, …, `Frontier24`; 4.7 MB in all, almost entirely the design lists as string literals) hold the 25 lower-bound theorems. The reasoning layer — `invKey_of_iso`, `pairwise_not_iso`, `lower_bound_of_cert` and their two-round versions — is proved without any evaluation and depends only on the axioms `propext`, `Classical.choice`, `Quot.sound`. The four definitional controls of Proposition 2.5(iii), `W0_isWD` and `cell_5_3_nonempty` are decided by the kernel (`decide`; axiom `propext` only). Everything that evaluates a large finite check — the sweep over $40^3$ candidates and the $40^4$ control at $(5, 3)$, the two invariant controls, and, for each of the 25 cells, the two facts "every listed design satisfies `IsWD`" and "the listed invariants are pairwise distinct" — is discharged by `native_decide`, i.e. by the compiled evaluator; each such theorem therefore carries, besides some or all of the standard three axioms, one or two named axioms of the form `cell_m_z._native.native_decide.ax_1_1` / `ax_1_2`, and no other. (The two invariant controls carry `propext`, `Quot.sound` and one named axiom; `cell_5_4_empty` carries `propext` and one named axiom; the 25 cell theorems and the two $(5, 3)$ exhaustiveness theorems carry the full triple.) We re-ran `#print axioms` on all 40 tagged declarations (and the two packaged certificates) for this paper from the compiled modules: the output is exactly as described, and contains no `sorryAx`. Checking the fourteen modules from source takes 7.5 minutes in all, with a peak resident set of 9.1 GB for `Cell53` and under 7.1 GB for every other module, against a baseline of 6.6 GB for `import Mathlib` alone; one measured trap is that a design list shipped as a chain of concatenated short literals exhausts the elaborator's recursion depth past about 1 MB and is slower below it, so each list is a single literal. The development is currently in a private repository: repository reference to be added at submission.

## Appendix A. The formal statements

The definitions and the statements of the 40 tagged declarations, verbatim (docstrings, attributes and proofs omitted). A `Design` is a list of rows, each a `List ℤ`; `IsWD m z k d` is Definition 2.1 as a Boolean; `Iso z` is Definition 2.2; `invKey` and `invKey2` are the invariants of Definition 2.4; `readDesigns s m z n` reads $n$ designs of size $m \times z$ out of the character array `s` in column-major order, one character (`0`, `+`, `-`) per entry.

**Definitions (`Defs`, `Refine`, `Cell53`).**

```
abbrev Row := List ℤ
abbrev Design := List Row
def entryOf (r : Row) (j : ℕ) : ℤ := r.getD j 0
def colOf (d : Design) (j : ℕ) : List ℤ := d.map (fun r => entryOf r j)
def dotZ (z : ℕ) (u v : Row) : ℤ :=
  ((List.range z).map (fun j => entryOf u j * entryOf v j)).sum
def colDot (d : Design) (j₁ j₂ : ℕ) : ℤ :=
  (d.map (fun r => entryOf r j₁ * entryOf r j₂)).sum

def IsWD (m z k : ℕ) (d : Design) : Bool :=
  (d.length == m) &&
  (d.all fun r => r.length == z) &&
  (d.all fun r => r.all fun x => x == 0 || x == 1 || x == -1) &&
  (d.all fun r => decide ((r.filter (fun x => x == 0)).length ≤ m - k)) &&
  ((List.range z).all fun j => ((colOf d j).filter (fun x => !(x == 0))).length == k) &&
  ((List.range z).all fun j₁ => (List.range z).all fun j₂ =>
      colDot d j₁ j₂ == (if j₁ == j₂ then (k : ℤ) else 0))

def negRow (r : Row) : Row := r.map (fun x => -x)
structure ColOp where
  perm : List ℕ
  sign : List ℤ
def applyRow (op : ColOp) (r : Row) : Row :=
  (List.range op.perm.length).map
    (fun t => op.sign.getD t 1 * entryOf r (op.perm.getD t 0))
def applyCols (op : ColOp) (d : Design) : Design := d.map (applyRow op)
def ValidOp (z : ℕ) (op : ColOp) : Prop :=
  op.perm.Perm (List.range z) ∧ op.sign.length = z ∧ ∀ s ∈ op.sign, s = 1 ∨ s = -1

inductive Iso (z : ℕ) : Design → Design → Prop
  | rowPerm {d d' : Design} : d.Perm d' → Iso z d d'
  | rowSign {d d' : Design} :
      List.Forall₂ (fun r r' => r' = r ∨ r' = negRow r) d d' → Iso z d d'
```

```
  | colOp {d : Design} {op : ColOp} : ValidOp z op → Iso z d (applyCols op d)
  | symm {d d' : Design} : Iso z d d' → Iso z d' d
  | trans {a b c : Design} : Iso z a b → Iso z b c → Iso z a c

def sortNat (l : List ℕ) : List ℕ := l.mergeSort (fun a b => decide (a ≤ b))
def encodeList (base : ℕ) (l : List ℕ) : ℕ := l.foldl (fun a x => a * base + x) 1
def innerCode (z : ℕ) (d : Design) (r : Row) : ℕ :=
  encodeList 64 (sortNat (d.map (fun r' => (dotZ z r r').natAbs)))
def invKey (z : ℕ) (d : Design) : ℕ :=
  encodeList (2 ^ 200) (sortNat (d.map (innerCode z d)))
def innerCode2 (z : ℕ) (d : Design) (r : Row) : ℕ :=
  encodeList (2 ^ 160)
    (sortNat (d.map (fun r' => innerCode z d r' * 64 + (dotZ z r r').natAbs)))
def invKey2 (z : ℕ) (d : Design) : ℕ :=
  encodeList (2 ^ 4000) (sortNat (d.map (innerCode2 z d)))

def charToInt (c : Char) : ℤ := if c = '+' then 1 else if c = '-' then -1 else 0
def readDesign (s : Array Char) (m z off : ℕ) : Design :=
  (List.range m).map (fun i =>
    (List.range z).map (fun j => charToInt (s.getD (off + j * m + i) '0')))
def readDesigns (s : Array Char) (m z n : ℕ) : List Design :=
  (List.range n).map (fun t => readDesign s m z (t * (m * z)))

-- Cell53: the 40 weight-2 columns of length 5, the 40^3 candidates, the representative
def cols53 : List (List ℤ) :=
  (allVecs 5).filter (fun c => (c.filter (fun x => !(x == 0))).length == 2)
def rowsOfCols (m : ℕ) (cs : List (List ℤ)) : Design :=
  (List.range m).map (fun i => cs.map (fun c => entryOf c i))
def allDesigns53 : List Design :=
  cols53.flatMap (fun a => cols53.flatMap (fun b => cols53.map (fun c => rowsOfCols 5 [a, b, c])))
def W0 : Design := [[1, 1, 0], [1, -1, 0], [0, 0, 1], [0, 0, 1], [0, 0, 0]]
```

**The packaged certificates (`Defs`, `Refine`; not tagged).**

```
theorem lower_bound_of_cert {m z k : ℕ} (L : List Design)
    (hwd : L.all (fun d => IsWD m z k d) = true)
    (hnd : nodupFast (L.map (invKey z)) = true) :
    (∀ d ∈ L, IsWD m z k d = true) ∧ L.Pairwise (fun a b => ¬ Iso z a b)

theorem lower_bound_of_cert2 {m z k : ℕ} (L : List Design)
    (hwd : L.all (fun d => IsWD m z k d) = true)
    (hnd : nodupFast (L.map (invKey2 z)) = true) :
    (∀ d ∈ L, IsWD m z k d = true) ∧ L.Pairwise (fun a b => ¬ Iso z a b)
```

**The one-round invariant and the controls (`Defs`; Proposition 2.5).**

```
theorem invKey_of_iso {z : ℕ} {d d' : Design} (h : Iso z d d') :
    invKey z d = invKey z d'

theorem pairwise_not_iso {z : ℕ} {L : List Design}
    (h : (L.map (invKey z)).Nodup) : L.Pairwise (fun a b => ¬ Iso z a b)

theorem control_nonvacuous :
    IsWD 4 2 2 [[0, 0], [0, 0], [1, 1], [1, -1]] = true

theorem control_not_orthogonal :
    IsWD 4 2 2 [[0, 0], [0, 0], [1, 1], [1, 1]] = false

theorem control_wrong_weight :
    IsWD 4 2 2 [[0, 0], [0, 0], [1, 1], [0, -1]] = false

theorem control_row_bound :
    IsWD 4 3 2 [[0, 0, 0], [0, 0, 1], [1, 1, 0], [1, -1, 1]] = false

theorem control_invariant_separates :
```

```
    invKey 2 [[0, 0], [0, 0], [1, 1], [1, -1]] ≠
      invKey 2 [[0, 1], [0, -1], [1, 0], [1, 0]]
```

**The two-round invariant (`Refine`; Proposition 2.5).**

```
theorem invKey2_of_iso {z : ℕ} {d d' : Design} (h : Iso z d d') :
    invKey2 z d = invKey2 z d'

theorem pairwise_not_iso2 {z : ℕ} {L : List Design}
    (h : (L.map (invKey2 z)).Nodup) : L.Pairwise (fun a b => ¬ Iso z a b)

theorem control_invKey2_separates :
    invKey2 2 [[0, 0], [0, 0], [1, 1], [1, -1]] ≠
      invKey2 2 [[0, 1], [0, -1], [1, 0], [1, 0]]
```

**The cell $(5, 3)$ (`Cell53`; Theorem 4.1).**

```
theorem W0_isWD : IsWD 5 3 2 W0 = true

theorem cell_5_3_one_class : ∀ d : Design, IsWD 5 3 2 d = true → Iso 3 W0 d

theorem cell_5_3_no_two :
    ¬ ∃ d₁ d₂ : Design, IsWD 5 3 2 d₁ = true ∧ IsWD 5 3 2 d₂ = true ∧ ¬ Iso 3 d₁ d₂

theorem cell_5_3_nonempty : ∃ d : Design, IsWD 5 3 2 d = true

theorem cell_5_4_empty :
    (cols53.flatMap (fun a => cols53.flatMap (fun b => cols53.flatMap (fun c =>
      cols53.map (fun e => rowsOfCols 5 [a, b, c, e]))))).all
        (fun x => !(IsWD 5 4 2 x)) = true
```

**The row $m = 23$ (`Row23`; Theorems 4.2 and 4.3).**

```
theorem cell_23_3 :
    ∃ L : List Design, L.length = 5 ∧
      (∀ d ∈ L, IsWD 23 3 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 3 a b)

theorem cell_23_4 :
    ∃ L : List Design, L.length = 22 ∧
      (∀ d ∈ L, IsWD 23 4 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 4 a b)

theorem cell_23_5 :
    ∃ L : List Design, L.length = 111 ∧
      (∀ d ∈ L, IsWD 23 5 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 5 a b)

theorem cell_23_6 :
    ∃ L : List Design, L.length = 5 ∧
      (∀ d ∈ L, IsWD 23 6 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 6 a b)

theorem cell_23_7 :
    ∃ L : List Design, L.length = 5 ∧
      (∀ d ∈ L, IsWD 23 7 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 7 a b)
```

**The cell $(24, 4)$ (`Row24`; Theorem 4.6).**

```
theorem cell_24_4 :
    ∃ L : List Design, L.length = 77 ∧
      (∀ d ∈ L, IsWD 24 4 22 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 4 a b)
```

**The cell $(16, 7)$ (`Frontier16`; Theorem 5.1).**

```
theorem cell_16_7 :
    ∃ L : List Design, L.length = 2211 ∧
      (∀ d ∈ L, IsWD 16 7 14 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 7 a b)
```

**The cells $(16, 8)$–$(16, 10)$ (`Frontier16b`; Theorem 5.2).**

```
theorem cell_16_8 :
    ∃ L : List Design, L.length = 1890 ∧
      (∀ d ∈ L, IsWD 16 8 14 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 8 a b)

theorem cell_16_9 :
    ∃ L : List Design, L.length = 1050 ∧
      (∀ d ∈ L, IsWD 16 9 14 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 9 a b)

theorem cell_16_10 :
    ∃ L : List Design, L.length = 822 ∧
      (∀ d ∈ L, IsWD 16 10 14 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 10 a b)
```

**The cells $(18, 6)$, $(18, 13)$, $(18, 14)$ (`Frontier18`; Theorems 5.4 and 5.6).**

```
theorem cell_18_6 :
    ∃ L : List Design, L.length = 3113 ∧
      (∀ d ∈ L, IsWD 18 6 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 6 a b)

theorem cell_18_13 :
    ∃ L : List Design, L.length = 81 ∧
      (∀ d ∈ L, IsWD 18 13 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 13 a b)

theorem cell_18_14 :
    ∃ L : List Design, L.length = 19 ∧
      (∀ d ∈ L, IsWD 18 14 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 14 a b)
```

**The cells $(18, 7)$–$(18, 12)$ (`Frontier18b`; Theorem 5.5).**

```
theorem cell_18_7 :
    ∃ L : List Design, L.length = 3376 ∧
      (∀ d ∈ L, IsWD 18 7 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 7 a b)

theorem cell_18_8 :
    ∃ L : List Design, L.length = 1158 ∧
      (∀ d ∈ L, IsWD 18 8 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 8 a b)

theorem cell_18_9 :
    ∃ L : List Design, L.length = 376 ∧
      (∀ d ∈ L, IsWD 18 9 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 9 a b)

theorem cell_18_10 :
    ∃ L : List Design, L.length = 794 ∧
      (∀ d ∈ L, IsWD 18 10 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 10 a b)

theorem cell_18_11 :
    ∃ L : List Design, L.length = 598 ∧
      (∀ d ∈ L, IsWD 18 11 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 11 a b)
```

```
theorem cell_18_12 :
    ∃ L : List Design, L.length = 319 ∧
      (∀ d ∈ L, IsWD 18 12 16 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 12 a b)
```

**The cell $(20, 5)$ (`Frontier20`; Theorem 5.8).**

```
theorem cell_20_5 :
    ∃ L : List Design, L.length = 604 ∧
      (∀ d ∈ L, IsWD 20 5 18 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 5 a b)
```

**The cells $(20, 6)$, $(20, 7)$ (`Frontier20b`; Theorem 5.9).**

```
theorem cell_20_7 :
    ∃ L : List Design, L.length = 2237 ∧
      (∀ d ∈ L, IsWD 20 7 18 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 7 a b)

theorem cell_20_6 :
    ∃ L : List Design, L.length = 5695 ∧
      (∀ d ∈ L, IsWD 20 6 18 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 6 a b)
```

**The cell $(22, 5)$ (`Frontier22`; Theorem 5.10).**

```
theorem cell_22_5 :
    ∃ L : List Design, L.length = 2302 ∧
      (∀ d ∈ L, IsWD 22 5 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 5 a b)
```

**The cell $(22, 6)$ (`Frontier22b`; Theorem 5.11).**

```
theorem cell_22_6 :
    ∃ L : List Design, L.length = 8511 ∧
      (∀ d ∈ L, IsWD 22 6 20 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 6 a b)
```

**The cell $(24, 5)$ (`Frontier24`; Theorem 5.12).**

```
theorem cell_24_5 :
    ∃ L : List Design, L.length = 489 ∧
      (∀ d ∈ L, IsWD 24 5 22 d = true) ∧
      L.Pairwise (fun a b => ¬ Iso 5 a b)
```

## References

[1] J. Lejeune Herman and P. Goos, *Constructing large orthogonal minimally aliased response surface designs through enumeration and combination of weighing designs*, arXiv:2608.04814v1 [stat.ME], 5 August 2026 (still version 1 on 2026-09-07). Read in full. Section, table and appendix numbers are those of the arXiv PDF of version 1, checked against a tectonic build of the e-print, whose `.aux` gives `tab:rc-single` = Table 1 and `tab:rc-single2` = Table 2. Pages of the arXiv PDF: Table 1 on 16, Table 2 on 20, Appendix A on 32, Appendix B (the completeness of the column-by-column procedure) on 33, Appendix C (the graph encoding) on 35, Appendix D with its Table 4 on 37, Appendix E on 38. The rebuild agrees on Tables 1 and 2 and runs one page later from Appendix A on. The quotations in Sections 1 and 2 are verbatim from its Sections 2, 3.2 and 4.1.1, from the legend of Table 1 and from the legend of Table 4.

[2] J. Núñez Ares and P. Goos, *Enumeration and multicriteria selection of orthogonal minimally aliased response surface designs*, Technometrics **62** (2020), no. 1, 21–36. Bibliographic data checked against Crossref (doi:10.1080/00401706.2018.1549103); cited by the source as the previous enumeration (complete to five columns, partial for six and seven); not read for this work.

[3] E. D. Schoen, P. T. Eendebak and P. Goos, *A classification criterion for definitive screening designs*, Ann. Statist. **47** (2019), no. 2, 1179–1202. Volume, issue and year checked against Crossref (doi:10.1214/18-AOS1723), whose record carries no page range; the pages are those of the publisher's article page at Project Euclid ("Ann. Statist. 47(2): 1179–1202 (April 2019)"). Cited by the source for conference designs (one zero per column); not read for this work.

[4] H. C. Chan, C. A. Rodger and J. Seberry, *On inequivalent weighing matrices*, Ars Combin. **21A** (1986), 299–333. The journal is not indexed by Crossref; authors, title, volume, year and pages were checked against the zbMATH record (Zbl 0599.05013, "Ars Comb. 21A, 299–333 (1986)"), which also carries the authors' abstract quoted in Section 1. Of the paper itself only Appendix A ("Inequivalent $W(n,k)$ for $n \leq 20$", pp. 315–323) was read here, from the University of Wollongong repository copy: its count column gives "$\geq 1$" at $(16,14)$ (p. 319) and at $(18,16)$ (p. 320), and "5 (M.H.)" at $(16,16)$.

[5] H. Ohmori, *Classification of weighing matrices of small orders*, Hiroshima Math. J. **22** (1992), no. 1, 129–176. Journal, volume, issue and year checked against Crossref (doi:10.32917/hmj/1206128615); pages as given by the source; not read for this work.

[6] H. Ohmori, *Classification of weighing matrices of order 13 and weight 9*, Discrete Math. **116** (1993), no. 1–3, 55–78. Checked against Crossref (doi:10.1016/0012-365X(93)90394-9); not read for this work.

[7] H. Ohmori and T. Miyamoto, *Construction of weighing matrices $W(17,9)$ having the intersection number 8*, Des. Codes Cryptogr. **15** (1998), no. 3, 259–269. Checked against Crossref (doi:10.1023/A:1008321209012); not read for this work.

[8] M. Harada and A. Munemasa, *On the classification of weighing matrices and self-orthogonal codes*, J. Combin. Des. **20** (2012), no. 1, 40–57; arXiv:1011.5382. Checked against Crossref (doi:10.1002/jcd.20295). Read from the arXiv e-print for this paper: its abstract and introduction state that it classifies weighing matrices of "orders $n \leq 17$ except $n = 16$" and, beyond that, the parameters $(n,k) = (16,6),(16,9),(16,12),(18,9)$.

[9] P. Lampio and M. Ganzhinov, *Classification of weighing matrices*, contributed talk, 8th Workshop on Design Theory, Hadamard Matrices and Applications (Hadamard 2025), Sevilla, 26–30 May 2025 (Thursday 15:10). Not published as of 2026-09-07; no arXiv version was found. Read from the speaker's slides (`https://us.ticmeet.com/assets/archivos/d6f1d9b8-d39f-4888-925a-7eb81c4905cc/Lampio.pdf`, PDF metadata author "Pekka Lampio Joint work with Mikhail Ganzhinov", created 2025-05-28) and from the workshop's book of abstracts (same host, `Book_of_abstracts.pdf`, p. 50), both read on 2026-09-07. Abstract, verbatim: "We provide a computational classification method for weighing matrices. With this method, we classify weighing matrices of orders 16, 17, 18, 19, 20 and 21 for all weights and matrices of some higher orders for some weights. The classification of weighing matrices of order 16 and weight 10 is also revised." The results slide (12/18) lists 17 at order 16 weight 14 and 4 at order 18 weight 16, both without a reference; the summary slide (16/18) reads "We have classified 32 cases.", "These results are preliminary.", "The work continues." The same slides are reference [22] of arXiv:2603.17552v1 (Goldberger, Ben-Av, Dula and Strassler, *Constructing, classifying and studying the space of small integer weighing matrices*, 18 March 2026), whose Section 1 says, verbatim, "A recent unpublished work of Lampio and Ganzhinov [22] classifies completely weighing matrices of orders 16, 18, 19, 20, 21 and partially 22, 24, 28."

[10] A. V. Geramita, J. M. Geramita and J. Seberry Wallis, *Orthogonal designs*, Linear and Multilinear Algebra **3** (1975/76), no. 4, 281–306. Checked against Crossref (doi:10.1080/03081087608817121; the Crossref title carries the typo "Orthogonal desingns"). Read for this paper from the University of Wollongong repository copy. Proposition 2 (p. 283), verbatim: "If $n$ is odd and a $W(n,s)$ exists, then $s$ is a square." Proposition 23 (Section 4, p. 292), verbatim: "If $n$ is odd then a necessary condition that $W(n,k)$ exist is that $(n-k)^2 - (n-k) + 2 > n$."

[11] M. Araya, M. Harada, H. Kharaghani, S. Suda and W.-H. Yu, *Unbiased weighing matrices of weight 9*, Des. Codes Cryptogr. **93** (2025), no. 10, 4235–4273. Checked against Crossref (doi:10.1007/s10623-025-01676-y); not read for this work.

[12] B. D. McKay, *Isomorph-free exhaustive generation*, J. Algorithms **26** (1998), no. 2, 306–324. Checked against Crossref (doi:10.1006/jagm.1997.0898).

[13] B. D. McKay and A. Piperno, *Practical graph isomorphism, II*, J. Symbolic Comput. **60** (2014), 94–112. Checked against Crossref (doi:10.1016/j.jsc.2013.09.003). The canonical labellings of Section 2.2 are computed by nauty version 2.8.9, built from the authors' source distribution.

[14] P. Kaski and P. R. J. Östergård, *Classification Algorithms for Codes and Designs*, Algorithms and Computation in Mathematics, vol. 15, Springer, Berlin/Heidelberg, 2006. Book record checked against Crossref (doi:10.1007/3-540-28991-7), which carries no series volume; the series volume 15 is from the Springer book page and from zbMATH (Zbl 1089.05001). Cited for the general account of isomorph-free generation and canonical augmentation: its Chapter 4, "Isomorph-Free Exhaustive Generation", pp. 105–143 (chapter title and pages from the registered chapter DOI 10.1007/3-540-28991-7_4).

[15] R. Craigen and H. Kharaghani, *Orthogonal designs*, Chapter V.2 in: Handbook of Combinatorial Designs, 2nd ed. (C. J. Colbourn and J. H. Dinitz, eds.), Chapman & Hall/CRC, Boca Raton, 2007, pp. 280–295. Book record checked against Crossref (doi:10.1201/9781420010541, dated November 2006; the printed second edition carries the copyright year 2007, and zbMATH, Zbl 1101.05001, dates it 2007). Chapter number and page range read off the book's own table of contents (Part V, "Hadamard Matrices and Related Designs"; the following chapter, V.3, begins on p. 296). Cited for the standard definitions of weighing and conference matrices and for the existence tables, which record existence and not counts.

[16] M. Harada and A. Munemasa, *Database of small weighing matrices*, `https://www.math.is.tohoku.ac.jp/~munemasa/research/matrices/`, created 2010, last modified 2017; read on 2026-09-07. The companion data of [8]; each linked file is stated to contain, unless otherwise noted, all weighing matrices of the given order and weight up to equivalence.

The file $W(12, 10)$ was downloaded and checked here (five matrices, pairwise inequivalent, each a $W(12, 12, 10)$ in the sense of Definition 2.1), and likewise $W(20, 18)$ (53 matrices, pairwise inequivalent).

[17] J. Seberry, *Orthogonal Designs: Hadamard Matrices, Quadratic Forms and Algebras*, Springer, Cham, 2017. Book record checked against Crossref (doi:10.1007/978-3-319-59032-5); cited for background on weighing matrices; not read for this work.

[18] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction – CADE 28, Lecture Notes in Computer Science **12699**, Springer, 2021, pp. 625–635.

[19] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), ACM, 2020, pp. 367–381.