# Certified approximation bounds for minimum sensor sets in semiconductor test data

Luis Eduardo Muñoz Guerrero

Universidad Tecnológica de Pereira, Facultad de Ingenierías, Pereira, Risaralda, Colombia.

Email ID: lemunozg@utp.edu.co

ORCID 0000-0002-9414-6187

**Abstract:**

Choosing the fewest sensors that separate failing from passing production lots is a combinatorial problem, and it is NP-hard. We study it on the SECOM dataset of a semiconductor manufacturing process, with 1,567 lots and 474 non-constant sensors. An exact integer-programming procedure did not certify an optimum within 600 seconds per solve. A verified cover of 7 sensors separates every separable pair, and the two rankings we tested need 17 or more sensors, so the excess of the ranking over the minimum is at least 1.42 (142 %) on this table; that bound rests on a single pair of lots, and an exploratory check on all 590 columns gives the same value. The excess is about half as large when the labels are randomly permuted (median lower bound 0.71, against 1.42 as ratios), so part of it does not depend on which lots failed. In an exploratory cross-validation, a cover chosen in each training fold shows no detectable advantage in these folds over random sensor sets of its size, with an area under the curve (AUC) of 0.537 against 0.541, while a mutual-information ranking of that size reaches 0.702. We report the planned confirmatory comparison as not decided, describe the failing validity checks, and benchmark the exact procedure

**Keywords**: minimum discriminating subset; set cover; approximation bounds; greedy algorithm; sensor selection; integer programming.

## INTRODUCTION

A production line that records many measurements per lot must decide which of them to keep. Every sensor that is kept has to be installed, calibrated, stored and watched, so a sensor set that separates failing lots from passing lots with fewer sensors may matter to the engineers who run the line. Feature selection is the usual name for this decision, and it is often made with a ranking: each sensor receives a score, and the top sensors are kept. Their cost is that nothing guarantees that the set they produce is small.

The question has a precise form. Given that a set of sensors separates every pair of a failing lot and a passing lot that can be separated at all, how small can that set be, and can its size be certified as minimal? For a pair to be separated, some sensor must read the two lots in different categories. The smallest separating set is a classical combinatorial object, the minimum discriminating subset. It is NP-hard, so, unless P = NP, no method finds the minimum quickly on every instance.

We study this question on the SECOM dataset of a semiconductor manufacturing process, with 1,567 lots, 474 non-constant sensors and 104 failing lots. Our main finding is not the one we planned. The exact procedure, which solves the integer program with the open-source solver HiGHS and certifies its answer, does not reach a certified minimum on the full table within a time limit of 600 seconds per solve. What the data do support is a verified bound. A greedy cover with 7 sensors separates every separable pair, and the best ranking needs 17. Because the optimum is at most the size of any verified cover, the ranking's excess over the minimum is at least 142 %, whatever the minimum turns out to be.

The paper makes four contributions. It applies the pair-separation formulation, which Section 2 traces to earlier work, to sensor selection. It then brings a greedy upper bound and a linear-relaxation lower bound to it, on a real table. It reports what a certified exact method achieves on a real manufacturing table of this size, together with the scaling of its cost, and it shows where it stops. And it compares a planned confirmatory analysis with an exploratory analysis built on the verified bounds, whose validity checks fail, and we report that.The claim is limited to this dataset and to this application of a formulation that is known. We do not claim that the method finds the best sensors for a plant, because the outcome is a label in a historical record and the sensors have no known physical meaning in the data. We do not claim that the greedy cover is near the minimum, because the lower bound of the linear relaxation is far below the upper bound.

The rest of the paper is organized as follows. Section 2 reviews the related work, and Section 3 states

the problem and the two rankings it is compared with. Section 4 describes the algorithms, Section 5 the data, and Section 6 the methods of the analysis. Section 7 reports the results, Section 8 discusses them and the threats to their validity, and Section 9 concludes.

**2. RELATED WORK**

Five literatures meet this problem, and, as far as our search found, none of them answers it on this dataset. The first is exact sparse selection for regression and classification. Best subset selection can be written as a mixed-integer program (Bertsimas et al., 2016). A related discrete formulation has been studied for learning sparse classifiers (Dedieu et al., 2020), and a study sets out a mixed-integer approach for selecting feature subsets in kernel support vector machine classification (Tamura et al., 2022). The simulations reported by Bertsimas et al. (2016) suggested that best subset selection performed better than the lasso and forward stepwise selection, but a later comparison (Hastie et al., 2017) found that neither dominates: best subset selection generally wins at high signal-to-noise ratios, the lasso at low ones. Discrete optimization has also been applied to rule lists over categorical features (Angelino et al., 2018). The lasso minimizes the residual sum of squares subject to a bound on the coefficients (Tibshirani, 1996). These methods optimize the fit of a model, not the smallest set of sensors that separates failing lots from passing ones.

The second literature is sensor selection, mainly by integer programming. Optimal sensor placement for model-based fault detection and isolation has been formulated as a mixed-integer optimization problem (Sarrate et al., 2007), and, when the sensors are mutually orthogonal, the static selection of sensors for least-squares estimation has been cast as a linear integer program (Moon and Başar, 2017). Sensor applications including radar and sonar have also been studied (Alqarni and Robertazzi, 2021). For spatial phenomena modeled as Gaussian processes, choosing sensor locations is a fundamental task (Krause et al., 2008). A convex relaxation chooses sensors and bounds the best achievable error (Joshi and Boyd, 2009). Optimal sensor placement for fault detection has been studied with binary integer linear programming (Rosich et al., 2008), and a dissertation investigates sensor placement for fault detectability and isolability with the least number of candidate sensors (Chi, 2014). The model-based works depend on a process model, so the distinguishability of faults comes from that model. Here it comes from labeled data, and no model is available. Adaptive sensor trees, which choose the sensors read for each case, aim to lower the acquisition cost at test time (Wang et al., 2015); they do not give one fixed set.

The third literature is data-driven feature selection for fault detection in semiconductor manufacturing. Feature selection has become a key step of fault detection (Wang et al., 2016), and a study examines 288 combinations of different approaches (Salem et al., 2018). In semiconductor manufacturing, metrology is generally costly, and sampling methods aim at minimizing the number of sites measured (Susto et al., 2018). Feature selection has also been applied to the performance screening of microcontrollers (Bellarmino et al., 2023). None of these studies, as far as our search could establish, certifies the size of the selected set, and none compares a ranking heuristic with a certified minimum.

The fourth literature states the same pair-separation condition for other purposes. The same condition also defines identifying codes in graphs, where every vertex must be uniquely identified by the vertices that cover it (Karpovsky et al., 1998). The discriminating sets of this paper are hitting sets of the family of separating-sensor sets, and generating inclusion-minimal hitting sets is a fundamental combinatorial problem, with applications to minimal unique column combinations in data profiling (Bläsius et al., 2022; Gainer-Dewar and Vera-Licona, 2017). That work concerns inclusion-minimal hitting sets; the problem here asks for the smallest one. Set covering methods, exact and heuristic, are surveyed by Caprara et al. (2000). Wieczorek et al. (2021) define a minimum query set, the smallest set of attribute-value pairs for which any two objects can be distinguished, and obtain it with a linear programming model in the first stage of their procedure. In rough set theory such sets are called reducts, and finding a minimum reduct is NP-hard (Wang et al., 2006); a review of that literature is beyond our search, so we cite only this paper.

The fifth literature covers filter criteria that score features one at a time, including the two rankings of Section 3. Sequential procedures do not guarantee an optimal subset (Narendra and Fukunaga, 1977): the two best single measurements need not form the best pair (Cover, 1974). Branch and bound finds the best subset without exhaustive search when the criterion is monotone (Narendra and Fukunaga, 1977). A feature subset is inconsistent when two instances share its feature values but carry different class labels (Dash and Liu, 2003). Mutual information has been used to select an informative subset (Battiti, 1994). A later criterion, minimum-redundancy maximum-relevance (mRMR), combines relevance and redundancy (Peng et al., 2005). Conditional mutual information gives a very fast binary selection rule (Fleuret, 2004), and a unifying framework brings almost two decades of research on heuristic filter criteria under a single theoretical

interpretation (Brown et al., 2012). The Relief algorithm selects relevant features using a statistical method (Kira and Rendell, 1992). Reviews give a taxonomy of feature selection techniques (Saeys et al., 2007; Li et al., 2017), and variable and feature selection have become the focus of much research (Guyon and Elisseeff, 2003).

Table I lists the closest of these works, with the source that each comparison rests on. Our search queried OpenAlex with the recorded query strings (first ten results each), checked DOIs and metadata with Crossref, and added dated manual web searches. Seventeen query strings were fixed before the plan froze, and fifteen more were added later; all ran before the manuscript was finished. Related formulations in sensor placement, and catalogues of NP-complete problems, may also contain this problem; our search did not cover them.

**Table I. Closest works cited in Section 2 and the source behind each comparison**

| Work | Research question | Data | Method | Metric | Main result | Source read |
|---|---|---|---|---|---|---|
| Bertsimas et al. (2016) | Best subset in linear regression | Synthetic and real (not read) | Mixed-integer optimization | Selection accuracy | Mixed-integer optimization solves best subset at larger scale | Abstract |
| Dedieu et al. (2020) | Sparse (l0l0) classifiers | Not SECOM | Exact mixed-integer and continuous optimization | Accuracy and scale | Discrete formulation for classifiers with few features | Abstract |
| Tamura et al. (2022) | Feature subsets for kernel support vector machines | Not read | Mixed-integer optimization | Not read | Mixed-integer approach for binary classification | Abstract |
| Hastie et al. (2017) | Best subset against stepwise and lasso | Simulations (not read) | Mixed-integer optimization against heuristics | Prediction error (not read) | Not read in full text | Not read |
| Sarrate et al. (2007) | Sensor placement for fault detection and isolation | Process model | Mixed-integer optimization | Number of sensors; detectability | Mixed-integer formulation of placement | Abstract |
| Rosich et al. (2008) | Sensor placement for fault detection and isolation | Models | Binary integer linear program | Detectability and isolability constraints | Specifications as linear inequalities | Abstract |
| Chi (2014), dissertation | Placement for detectability and isolability | Models | Computation-intensive algorithm | Not read | Not read in full text | Not read |
| Alqarni et al. (2021) | Optimal number of sensor signals | Simulation | Integer linear program and heuristic | Number of signals | Integer linear program selects the optimal number of signals | Abstract |
| Salem et al. (2018) | Fault diagnosis with unbalanced | SECOM | 288 combinations of pruning, | Not read | Experimental evaluation | Abstract |

| | | | | | | |
|---|---|---|---|---|---|---|
| | and incomplete data | | imputation, selection and classification | | without optimality | |
| Wang et al. (2016) | Discriminative selection for fault detection in semiconductors | Semiconductor data (SECOM in the abstract) | Support vector data description and heuristic selection | Not read | Selection is hard under class imbalance | Abstract |
| Moon et al. (2017) | Static optimal sensor selection | Signals, not manufacturing data | Integer linear program, orthogonal case | Trace of the estimator covariance | Choosing dd of ss sensors as an integer linear program | Abstract |
| Wieczorek et al. (2021) | Decision tree from a minimum set of queries | Eleven sets from a public repository and a survey; no SECOM | 0–1 integer program; genetic algorithm for the tree | Size of the query set | Pair-distinguishing formulation with linear programming | Full text |
| Karpovsky et al. (1998) | Identifying codes in graphs | Not applicable (graphs) | Covering with unique identification | Size of the code | Structure of separated pairs, as in the sensor problem | Abstract |
| Wang et al. (2006) | Minimum reduct of an information system | Not applicable | Discernibility matrix | Not applicable | NP-hardness of the minimum reduct | Abstract |
| Wang et al. (2015) | Sensor trees minimizing acquisition cost at test time | Not applicable (cost-sensitive classification) | Adaptive selection for each case | Acquisition cost | Chooses sensors per case, not one fixed set | Not stated |

Source read: abstract means only the abstract was read; full text means the article was read; not read means the work is cited for its formulation and no result is compared; not stated means the source is not recorded.

**3. PROBLEM FORMULATION**

Each lot in the dataset is a vector of readings from the 474 non-constant sensors, some of them missing, and a binary outcome: fail or pass. Every reading is replaced by its quantile category, one of at most four per sensor, and missing values form a fifth category. A failing lot and a passing lot are separated by a sensor when the two lots fall in different categories of that sensor. A pair is separable when at least one sensor separates it.

Let q be the number of quantile intervals, here four, and let $x_{is}$ be the reading of lot i for sensor s. The cut points of sensor s are the empirical quantiles of its observed readings at the levels k/q, as Equation (1) gives.

$$\theta_{s,k} = Q_s\left(\frac{k}{q}\right), \quad k = 1, \ldots, q-1. \tag{1}$$

The category of a reading is 0 when the reading is missing, and otherwise one plus the number of cut points that the reading reaches, as Equation (2) states.

$$c_{is} = \begin{cases} 0 & \text{if } x_{is} \text{ is missing,} \\ 1 + \sum_{k=1}^{q-1} \mathbf{1}\left[x_{is} \geq \theta_{s,k}\right] & \text{otherwise.} \end{cases} \tag{2}$$

Let F be the set of failing lots, P the set of passing lots, and C the set of separable pairs, C = {(i, j) ∈ F ×

P : the set of separating sensors is not empty}. For each pair (i, j) let Δ(i, j) be the set of sensors that separate it, as Equation (3) gives.

$$\Delta(i,j) = \{\, s : c_{is} \neq c_{js} \,\}, \quad (i,j) \in F \times P. \tag{3}$$

A set S of sensors is discriminating when S meets Δ(i, j) for every pair in C. The minimum discriminating subset problem asks for a discriminating set of smallest size. With one binary variable $y_s$ per sensor, it is the integer program in Equation (4).

$$\min \sum_{s} y_s \quad \text{subject to} \quad \sum_{s \in \Delta(i,j)} y_s \geq 1 \;\; \forall (i,j) \in C, \quad y_s \in \{0,1\}. \tag{4}$$

It is NP-hard even when every sensor has two categories, by a polynomial reduction from minimum set cover, which is itself NP-hard (Feige, 1998). We may assume that every element lies in some subset, since otherwise the instance has no cover. Given a universe U of m elements and r subsets of U, create one sensor per subset, one passing lot that every sensor reads in category 0, and one failing lot per element u, which the sensors of the subsets containing u read in category 1 and the other sensors in category 0. Then C consists of the pairs ($f_u$, $p_0$), the set Δ($f_u$, $p_0$) is the family of subsets that contain u, and a discriminating set is exactly a cover of U. The reduction does not settle the case of four categories cut at quartiles. The decision version is in NP, because a candidate set can be checked in polynomial time.

The analysis compares two sizes. The first is the certified minimum $k_{exact}$, the size of a smallest discriminating set that a solver proves optimal. The second is $k_{rank}$, the smallest k such that the first k sensors of a ranking separate every pair in C, as Equation (5) defines for a ranking r = (r_1, r_2, …) of the sensors.

$$k(r) = \min\{\, k : \{r_1, \ldots, r_k\} \cap \Delta(i,j) \neq \emptyset \;\; \forall (i,j) \in C \,\}. \tag{5}$$

We use two rankings that a practitioner would plausibly use. The first scores each sensor by the mutual information between its categories and the outcome (a plug-in estimate (Paninski, 2003), in bits), as Equation (6) gives for two discrete variables A and B.

$$I(A;B) = \sum_{a,b} p\,(a,b) \log_2 \frac{p(a,b)}{p(a)\,p(b)}. \tag{6}$$

The second scores each sensor by its Fisher score on the raw readings, with missing values imputed by the sensor median. The imputed readings are the readings themselves when observed and the median of the sensor otherwise, as Equation (7) states.

$$\hat{x}_{is} = \begin{cases} x_{is} & \text{if } x_{is} \text{ is observed,} \\ \text{median}_s & \text{otherwise.} \end{cases} \tag{7}$$

The Fisher score is the squared difference of the mean imputed readings of failing and passing lots, divided by the sum of their sample variances, as Equation (8) gives. Each sample variance uses a divisor one less than the number of lots in its class.

$$\phi_s = \frac{(\bar{x}_s^F - \bar{x}_s^P)^2}{\mathrm{var}_s^F + \mathrm{var}_s^P}. \tag{8}$$

A constant sensor scores zero; ties in either ranking go to the smaller sensor index. The ranking is the better of the two, the conservative choice, that is the smaller $k_{rank}$, as Equation (9) states.

$$k_{rank} = \min\{\, k(r^{MI}),\; k(r^F) \,\}. \tag{9}$$

The relative excess of the ranking over the minimum is defined in Equation (10).

$$G = \frac{k_{rank} - k_{exact}}{k_{exact}}. \tag{10}$$

The greedy cover starts from the empty set and repeatedly adds the sensor that separates the largest number of pairs not yet separated. Write $n_s(M)$ for the number of pairs in a set M of pairs that sensor s separates, as Equation (11) defines.

$$n_s(M) = |\{\, (i,j) \in M : s \in \Delta(i,j) \,\}|. \tag{11}$$

Let $M_0 = C$ be the pairs not yet separated. At step t the greedy cover adds the sensor $s_t$ with the largest count $n_s(M_{t-1})$, with ties going to the smaller index, as Equation (12) states.

$$s_t = \min\{\, s : n_s(M_{t-1}) = \max_r n_r(M_{t-1}) \,\}. \tag{12}$$

The pairs that $s_t$ separates leave the uncovered set, which gives $M_t$ as Equation (13) states.

$$M_t = \{\,(i,j) \in M_{t-1} : s_t \notin \Delta(i,j)\,\}. \tag{13}$$

Let $S_g$ be the resulting set, and let d be the largest number of pairs that a single sensor separates, which Equation (14) gives as the largest size of the sets $C_s$ of pairs that s separates.

$$d = \max_s |C_s|, \qquad C_s = \{\,(i,j) \in C : s \in \Delta(i,j)\,\}. \tag{14}$$

For a set X of pairs, a set S of sensors covers the share $\sigma_X(S)$ of X, the fraction of its pairs that S separates, as Equation (15) defines. A set S is discriminating exactly when $\sigma_C(S) = 1$.

$$\sigma_X(S) = \frac{|\{\,(i,j) \in X : S \cap \Delta(i,j) \neq \emptyset\,\}|}{|X|}. \tag{15}$$

The greedy cover satisfies the bound in Equation (16), where the optimal value of Equation (4) (OPT) is the number of sensors in an optimal set. $H_d$ grows only logarithmically in d.

$$|S_g| \le H_d \cdot \mathrm{OPT}, \qquad H_d = \sum_{k=1}^{d} \frac{1}{k}. \tag{16}$$

The argument charges each pair, when the greedy cover first separates it, with a share of one over the number of new pairs that the chosen sensor separates at that step, so the shares add up to the size of the greedy cover. Let t(i,j) be the step at which the pair (i, j) is first separated, and let $w_{ij}$ be its share, as Equation (17) states.

$$|S_g| = \sum_{(i,j)\in C} w_{ij}, \qquad w_{ij} = \frac{1}{n_{s_t}(M_{t-1})}, \quad t = t(i,j). \tag{17}$$

Fix an optimal set $S^*$ and one of its sensors s, and let $C_s$ be the pairs that s separates, so that $|C_s|$ is at most d. Greedy chooses the sensor with the most new pairs. When it covers the j-th pair of $C_s$, in the order in which greedy covers them, at least $|C_s| - j + 1$ pairs of $C_s$ are still uncovered, so the chosen sensor separates at least that many new pairs, and the share of that pair is at most $1/(|C_s| - j + 1)$. The shares of the pairs in $C_s$ therefore add up to at most $1 + 1/2 + \ldots + 1/|C_s|$, which is at most $H_d$, as Equation (18) states.

$$\sum_{(i,j)\in C_s} w_{ij} \le \sum_{k=1}^{|C_s|} \frac{1}{k} \le H_d. \tag{18}$$

Every pair is separated by some sensor of the optimal set, so the total share, which equals the size of the greedy cover, is at most $H_d$ OPT, as Equation (19) shows.

$$|S_g| \le \sum_{s\in S^*} \sum_{(i,j)\in C_s} w_{ij} \le H_d\,|S^*|. \tag{19}$$

A second, independent bound comes from the linear relaxation of Equation (4), in which each $y_s$ may take any value in [0, 1], as Equation (20) gives.

$$z_{LP} = \min\{\sum_s y_s : \sum_{s\in\Delta(i,j)} y_s \ge 1 \;\forall (i,j) \in C,\; 0 \le y_s \le 1\}. \tag{20}$$

Its optimal value $z_{LP}$ satisfies Equation (21).

$$z_{LP} \le \mathrm{OPT}. \tag{21}$$

Equation (16) and Equation (21) are the two guarantees that the analysis uses. Both are checked numerically on every instance whose optimum is certified, and none violates them.

Table II collects the notation of Sections 3 and 4.

**Table II. Notation of Sections 3 and 4**

| Symbol | Meaning | Where |
|---|---|---|
| F, P | failing lots; passing lots | Section 3 |
| C | separable pairs (failing, passing) that at least one sensor separates | Section 3 |

| Symbol | Meaning | Where |
|---|---|---|
| $c_{is}$ | category of lot i for sensor s: 0 if missing, otherwise one plus the number of cut points that the reading reaches | Equation (2) |
| $\theta_{s,k}$ | k-th cut point of sensor s, the empirical quantile at level k/q | Equation (1) |
| $\Delta(i, j)$ | sensors that separate the pair (i, j) | Equation (3) |
| $\sigma_X(S)$ | share of the pairs in X that the sensors of S separate | Equation (15) |
| $y_s$ | binary variable: sensor s is selected | Equation (4) |
| OPT | optimal value of Equation (4): the minimum number of sensors | Equation (16) |
| $k_{exact}$ | certified minimum size, from the exact procedure | Equation (10) |
| $k(r)$, $k_{rank}$ | smallest k such that the first k sensors of a ranking r separate all pairs; $k_{rank}$ is the better of the two rankings | Equation (5), Equation (9) |
| $I(A; B)$, $\varphi_s$ | mutual information in bits (plug-in); Fisher score of sensor s | Equation (6), Equation (8) |
| G | relative excess of the ranking over the minimum, $(k_{rank} - k_{exact}) / k_{exact}$ | Equation (10) |
| $S_g$ | set of sensors chosen by the greedy cover | Equation (12) |
| $n_s(M)$, $M_t$ | pairs of M that sensor s separates; pairs still uncovered after step t | Equation (11), Equation (13) |
| d | largest number of pairs that one sensor separates | Equation (14) |
| $H_d$ | harmonic number of d, the factor of the greedy guarantee | Equation (16) |
| $z_{LP}$ | optimal value of the linear relaxation of Equation (4) | Equation (20) |
| $G_{inf}$ | verified lower bound on G, from the greedy cover | Equation (28) |
| $G_{sup}$ | verified upper bound on G, from the relaxation | Section 4 |

Symbols as used in the text. Where gives the section or the equation that defines each one.

## 4. ALGORITHMS

**Exact procedure.** The certified procedure solves the integer program of Equation (4) by lazy constraint generation, which adds a constraint only when the current solution violates it. It starts from a working set W of 200 separable pairs drawn with a fixed seed. At each iteration it solves the program restricted to W with HiGHS through scipy.optimize.milp, with a time limit per solve. If the solver proves optimality, its solution y is checked against all separable pairs. The pairs that y leaves uncovered are added to W, up to 200 per iteration in index order, as Equation (22) states for the set $S = \{s : y_s = 1\}$ of selected sensors. The working set only grows, so each restricted program has at least as many constraints as the one before it. When no pair is left uncovered, the solution is the minimum. If the time limit is reached first, the procedure stops and reports the dual bound of its last solve.

$$W \leftarrow W \cup \{ (i,j) \in C : S \cap \Delta(i,j) = \emptyset \}. \tag{22}$$

The procedure is correct because the restricted program is a relaxation of Equation (4). Writing z(W) for its optimal value, as Equation (23) defines, z(W) is a lower bound on OPT, and when the solution covers every pair it is also an upper bound.

$$z(W) = \min\{\sum_s y_s : y \text{ binary}, \sum_{s \in \Delta(i,j)} y_s \geq 1 \ \forall (i,j) \in W\}. \tag{23}$$

Equation (24) states both bounds for every discriminating set S. Hence a run that ends with a certificate returns an optimal set, and a run that stops at the time limit returns no set, only its dual bound.

$$z(W) \leq OPT \leq |S|. \tag{24}$$

The time limit is 600 s per solve, the value fixed before the analysis. No exact run on the full table is recorded before the plan was frozen, and the full-table runs stored only dual bounds.

**Local search.** A simulated-annealing search (Kirkpatrick et al., 1983) over swaps left at least 5 pairs uncovered for six sensors in 200 restarts; as a calibration, the same search finds a verified cover of seven sensors in 3 restarts. Its cost is the number of separable pairs that a set S of sensors leaves uncovered, as Equation (25) defines.

$$f(S) = |\{\,(i,j) \in C : S \cap \Delta(i,j) = \emptyset\,\}|. \tag{25}$$

A move replaces one sensor of S by another sensor, and a move that changes the cost by $\delta$ is accepted with the probability in Equation (26). The temperature $T_t$ falls geometrically from $T_0 = 2$ to $T_1 = 0.05$ over the $N_{st} = 2{,}500$ steps of each restart.

$$\Pr(\text{accept}) = \min\{1,\ e^{-\delta/T_t}\}, \qquad T_t = T_0\left(\frac{T_1}{T_0}\right)^{t/(N_{st}-1)}. \tag{26}$$

**Greedy cover.** The greedy procedure repeatedly adds the sensor that separates the largest number of uncovered pairs, breaking ties by the smaller sensor index. Its output is always a discriminating set, so its size is an upper bound on OPT that can be verified directly. Each step scans every sensor, so the cost is $O(p\,|C|)$ per sensor added, with p the number of sensors. The size of the greedy cover is within a factor $H_d$ of the optimum, where d is the largest number of pairs that a sensor separates (Chvatal, 1979); this is the bound of Equation (16).

**A small example.** Table III gives a small instance with three failing lots, two passing lots and four sensors. Figure 1 follows the greedy steps on it and shows why three sensors are needed.

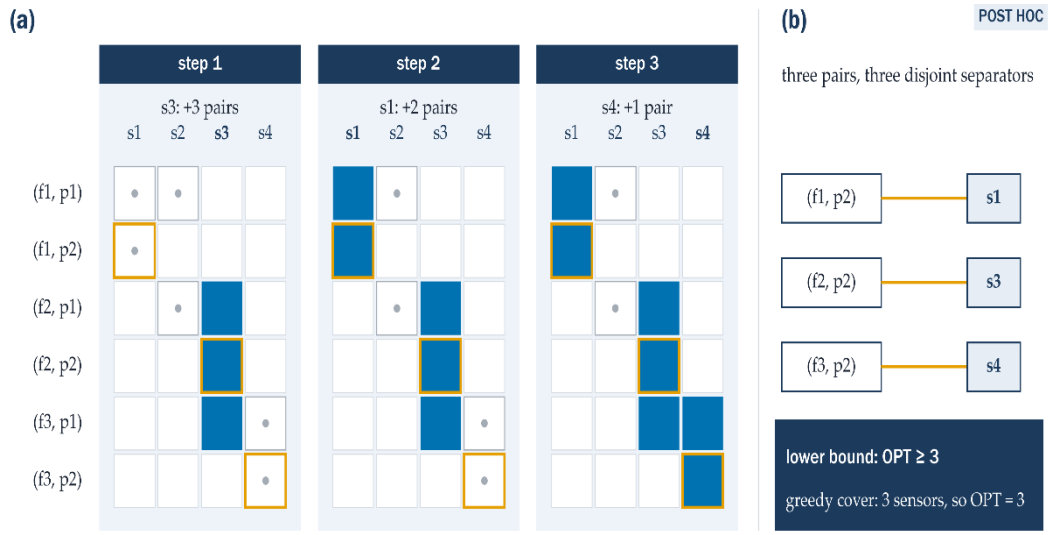



**Figure 1. The worked example of Section 4 as a process;**

Table III gives the matrix itself. (a) The three steps of the greedy cover. The sensor chosen at each step is written in bold. At each step the blue cells are the pairs that the chosen sensor separates; the dots mark the other sensors that separate a pair and are not yet chosen. The orange outlines mark the three pairs that only one sensor separates, which is the reason for the bound. (b) Those three pairs have three different separating sensors, so no two sensors suffice: the lower bound is three. The figure illustrates the algorithm; it reports no result and is post hoc.

**Table III. A small instance: the greedy cover and the pairs each sensor separates**

| Pair of lots | s1 | s2 | s3 | s4 | Sensors that separate the pair |
|---|---|---|---|---|---|
| (f1, p1) | • | • | | | s1, s2 |
| (f1, p2) | • | | | | s1 |
| (f2, p1) | | • | • | | s2, s3 |
| (f2, p2) | | | • | | s3 |
| (f3, p1) | | | • | • | s3, s4 |
| (f3, p2) | | | | • | s4 |

Each dot marks a sensor that separates the pair of lots. The greedy cover takes s3, which separates three pairs, then s1, which separates two of the remaining pairs, then s4, which separates the last pair: three sensors. No two sensors suffice, because (f1, p2), (f2, p2) and (f3, p2) are separated only by s1, s3 and s4, which are different sensors.

**Linear relaxation.** The value $z_{LP}$ of the relaxation of Equation (4) can be computed by the same lazy scheme, with a linear program in place of the integer program. The restricted program, whose value $z_{LP}(W)$ is defined in Equation (27), has fewer constraints than the full one, so its value is a lower bound on $z_{LP}$, and when no pair is violated the two coincide. Lazy generation makes the computation feasible on the full table, where the matrix of all separable pairs has 152,152 rows.

$$z_{LP}(W) = \min\{\sum_{s} y_s : \sum_{s \in \Delta(i,j)} y_s \geq 1 \ \forall (i,j) \in W,\ 0 \leq y_s \leq 1\}. \tag{27}$$

**Combining the two bounds.** Together the two procedures bracket the optimum. The greedy size bounds OPT from above, and the relaxation bounds it from below. Because G decreases as OPT grows, the greedy size gives the verified lower bound on G stated in Equation (28):

$$G \geq G_{inf} = \frac{k_{rank} - |S_g|}{|S_g|}. \tag{28}$$

The relaxation gives an upper bound on G, valid up to the solver's numerical tolerance, $G \leq G_{sup} = (k_{rank} - z_{LP})/z_{LP}$. These bounds need no certificate from the exact solver. They feed the exploratory analysis in Section 6.

**5. DATA**

We use the SECOM dataset of the University of California, Irvine (UCI) Machine Learning Repository (McCann and Johnston, 2008). It records 1,567 production lots of a semiconductor manufacturing process, each with 590 unnamed sensor readings, a pass or fail label, and a timestamp. The lots were recorded between 19 July 2008, 11:55 and 17 October 2008, 06:07, a period of 89.8 days, at a median interval of 37 minutes. Of these lots, 104 (6.6 %) carry the fail label. The three files of the repository were downloaded with a fixed procedure, and their SHA-256 digests were checked against the values recorded before the analysis. The attribute description says that there are 591 attributes, but each row of the data file has 590 values; the analysis uses those 590 values in the file. Table IV summarizes the dataset and its missing values. We describe the dataset following the datasheet approach (Gebru et al., 2021).

**Table IV. The dataset and its missing values**

| Quantity | Value |
|---|---|
| Missing cells, all columns | 41,951 of 924,530 (4.5 %) |
| Missing cells, pool | 41,136 of 742,758 (5.5 %) |
| Missingness patterns (all columns) | 198 |
| Lots with no missing value | 0 |
| Pool sensors with no missing value | 52 |
| Pool sensors with less than 10 % missing | 370 |
| Pool sensors with 10 % or more missing | 52 (largest share 91 %) |
| Pool sensors with four observed categories | 452 |
| Pool sensors with three, two or one observed category | 9, 3 and 10 |

Counts from the derived table of the analysis. The pool is the set of non-constant sensors on which the bounds are computed. Shares refer to the cells of the corresponding set.

The readings are heterogeneous, and many are missing. Table IV gives the missing cells, the missingness patterns and the categories of the pool. Of the columns, 116 take a single observed value, and we exclude them before the analysis, as the plan specifies. Every one of them is missing for some lots and separates at least one pair of lots. Under the coding in which a missing reading is a category of its own, such a column can separate a lot with a missing reading from a lot with the observed value, so excluding it restricts the problem in principle; under this coding, the constant columns separate 7,640 pairs of lots. The pool is defined in Equation (29): it keeps the sensors whose observed readings take at least two distinct values.

$$\text{pool} = \{\, s : |\{\, x_{is} : x_{is} \text{ is observed} \,\}| \geq 2 \,\}. \tag{29}$$

The bounds below refer to the pool of the remaining 474 sensors, which the plan fixed before the

analysis. As an exploratory check, we ran the greedy cover and the two rankings on all 590 columns. The verified cover has 7 sensors, against 7 on the pool; the better of the two rankings needs 17, against 17; the cover separates all 152,152 separable (fail, pass) pairs; and the lower bound on the excess is 1.42, against 1.42. No row is removed for any reason, so the main analyses use all 1,567 lots.

Two features of the data matter for the method. First, the documentation says that null readings, coded NaN, vary in frequency across the features, but it does not say why, so we claim nothing about their mechanism: the cover and the mutual-information ranking keep the missing values as a category of their own, and the multiverse analysis of Section 7 includes median-imputation specifications for those two. Second, the documentation gives no units for the readings and no per-sensor costs, so the problem cannot be weighted by cost. The data end in 2008, and the analysis claims nothing about the plant today. The lots keep their recording order, on which the temporal folds and the moving blocks of Section 6 depend.

**6. METHODS**

**Design and plan.** The study is computational and descriptive. It does not estimate a causal effect, and it does not model the plant. Before the plan, a data-quality check computed each sensor's correlation with the outcome, only to flag columns that separate it alone; only the raw label, a copy of the outcome, was flagged, and no sensor was selected on that basis. We then fixed the formulation, the primary estimand, the decision rules, the validity checks and the secondary analyses in a written plan, and recorded it under the version tag plan-congelado.

**Primary analysis as planned.** The planned confirmatory estimand is G (Equation (10)) computed on all 1,567 lots, with the certified minimum $k_{exact}$ from the exact procedure. Its uncertainty is a percentile interval at the 95 % level from 200 block-bootstrap replicas (Efron, 1979; Künsch, 1989). Each replica resamples the lots in moving blocks of 50 consecutive lots (Künsch, 1989), with uniform start positions and the last block truncated, and recomputes everything: the quantile cut points, both rankings and the exact procedure. The blocks keep the short-range dependence of the recording order, which the moving-block bootstrap is designed to preserve. Equation (30) gives the lot at each position t of replica b, where $v_{b,k}$ is the start of its k-th block and $\ell = 50$ is the block length.

$$i_b(t) = v_{b,\lfloor t/\ell \rfloor} + ((t-1) \bmod \ell), \qquad t = 1, \dots, n. \tag{30}$$

A replica counts only if its exact solution is certified. The percentile interval of G is given by Equation (31), where $\alpha$ is the error level, here 5 %, and the limits are empirical quantiles of the B replicas $G^{(b)}$.

$$\mathcal{I} = \left[ Q_{\alpha/2}(G^{(1)}, \dots, G^{(B)}),\ Q_{1-\alpha/2}(G^{(1)}, \dots, G^{(B)}) \right]. \tag{31}$$

The planned decision is given by Equation (32), where L and U are the limits of the interval and c is the proportion of certified replicas. It is positive when the lower limit of the interval is at least 0.20 and at least 95 % of the replicas are certified; it is a null result when the upper limit is below 0.20; otherwise it is inconclusive. A conclusion also requires an interval narrower than 0.10, as the plan states for precision. The threshold of 0.20 is a judgment, and the multiverse analysis below tests how much the conclusion depends on it.

$$D = \begin{cases} \text{inconclusive} & \text{if the interval is undefined or } U - L \geq 0.10, \\ \text{positive} & \text{if } L \geq 0.20 \text{ and } c \geq 0.95, \\ \text{null} & \text{if } U < 0.20, \\ \text{inconclusive} & \text{otherwise.} \end{cases} \tag{32}$$

**Validity checks.** A positive primary result has to pass four checks. V1 requires at least 95 % certified replicas and optimal values for all reported exact solutions. V2 permutes the outcome labels at random, 20 times with a fixed seed, and requires that the median of G under permutation be below 0.10. Its permutation p-value (Phipson and Smyth, 2010) is one-sided, as Equation (33) gives, where $G^{(b)}$ is the value of G after the b-th permutation of the outcomes and $N_\pi$ is the number of permutations.

$$p_\pi = \frac{1 + |\{ b : G^{(b)} \geq \widehat{G} \}|}{1 + N_\pi}. \tag{33}$$

After the freeze, 1,000 permutations with the same seed were run as an exploratory extension. V3 is a temporal placebo: the lots are split into five contiguous blocks $B_1$, …, $B_5$ with $n_1$, …, $n_5$ lots, and within each block the outcome is replaced by the outcome of the next block, cyclically, with the sequence repeated or truncated to the block length, as Equation (34) states, where $o^{(m)}{}_u$ is the outcome of the u-th lot of block $B_m$.

$$\tilde{o}_t^{(k)} = o_{1+((t-1) \bmod n_{k^+})}^{(k^+)}, \qquad k^+ = k+1 \text{ for } k < 5, \quad 5^+ = 1. \tag{34}$$

The median of G over the five blocks must be below 0.10. This placebo does not isolate temporal alignment, because its labels change the number of failing lots in each block. V4 checks for leakage: in the out-of-sample analysis the cut points, the imputation medians, the cover and the ranking are computed on the training blocks only, and an automated test changes the test blocks and confirms that none of these quantities moves.

**Secondary analyses.** The out-of-sample comparison H2 belongs to the planned confirmatory family. As planned, it needs a certified minimum on each training block, which the exact procedure cannot provide, so it is run here with the greedy cover, on the same five blocks, and reported as exploratory. Its decision is positive when the lower limit of its 95 % percentile interval, from 2,000 resamples of the five folds, is at least zero; that rule is ours, not the plan's. The plan prescribes the Holm correction (Holm, 1979) for the confirmatory family. No p-value enters, so nothing is adjusted; this departs from the plan, and we declare it. The exploratory family (H3, the multiverse, the controls and the benchmark) is corrected with the Benjamini–Hochberg procedure (Benjamini and Hochberg, 1995), whose only planned p-value is the one-sided permutation p of V2; the p-value from 1,000 permutations is reported uncorrected. At the conventional significance level the permutation p-value would be significant, but the validity check, which uses the median, fails, so the result does not validate the positive. The plan lists V2 and V3 in both families; we follow the family definitions, under which the controls are exploratory.

For each fold k, the cover $S^{(k)}$ is computed on the other four blocks, before and after the test block, and it has $\kappa_k$ sensors. The ranking $r^{(k)}$ of the same size is taken from whichever of the two rankings separates more training pairs, and from the mutual-information ranking when they separate the same number, as Equation (35) states, where $\mathrm{sep}_k(S)$ is the number of training pairs of fold k that S separates and $r_{1:\kappa}$ denotes the first $\kappa$ sensors of a ranking r.

$$r^{(k)} = \begin{cases} r^{\mathrm{MI}}_{1:\kappa_k} & \text{if } \mathrm{sep}_k(r^{\mathrm{MI}}_{1:\kappa_k}) \geq \mathrm{sep}_k(r^{\mathrm{F}}_{1:\kappa_k}), \\ r^{\mathrm{F}}_{1:\kappa_k} & \text{otherwise.} \end{cases} \tag{35}$$

The out-of-sample effect is the difference between the proportions of test pairs separated by the two sets, pooled over the folds, as Equation (36) gives, with a percentile interval from resampling the five folds. Here $h^{c}_k$ and $h^{r}_k$ count the test pairs of fold k that the cover and the ranking separate, $N_k$ counts the separable test pairs of that fold, and J is the set of folds used: the five folds for the estimate, and the resampled folds for the interval.

$$\widehat{\Delta}_J = \frac{\sum_{k\in J} h^{c}_k - \sum_{k\in J} h^{r}_k}{\sum_{k\in J} N_k}. \tag{36}$$

The multiverse varies the number of intervals q over 3, 4 and 5 and the treatment of missing values (a category of its own, or the median), and it reports the lower bound on the excess for each of the six specifications against the thresholds 0.10, 0.20 and 0.30. In the median treatment the categories are computed from the imputed readings of Equation (7), so no category is reserved for missing values. We omit the plan's variants with each ranking alone or with indistinguishable pairs (pairs no sensor separates); the second would change nothing, since no specification has such pairs. Each specification also gets a 95 % interval from 200 block-bootstrap replicas, drawn once and recomputed within each; this exploratory addition is reported as such.

The benchmark records wall-clock time and the working-set memory above its base, for both procedures, over subsets of 50, 100, 200 and 474 sensors and of 400, 800 and 1,567 lots. Times depend on the machine and load; subsets are prefixes of the sensor and lot orders. A first run with a process pool was discarded: it stopped delivering its last tasks and wrote its results only at the end. The reported benchmark runs one process per combination, four at a time, with the same code. A smaller grid, from 10 to 30 sensors and from 100 to 400 lots, ran five repetitions per combination with the same procedure, since most sizes of the main benchmark did not certify. Each combination of this grid had its own process and a time limit of 900 s; none reached it. An earlier attempt at this grid, with a process pool and 800 lots, predated the per-combination limit and was stopped after 27 minutes.

The bias-corrected and accelerated interval (Efron, 1987) uses a block jackknife of 50 lots: the statistic is recomputed with each block of 50 consecutive lots left out, $\theta_{(m)}$ denotes the statistic without block m, and $\bar{\theta}$ is the mean of these values. Its bias correction is given by Equation (37).

$$z_0 = \Phi^{-1}\left(\frac{|\{\, b : G^{(b)} < \widehat{G} \,\}| + 1/2\ |\{\, b : G^{(b)} = \widehat{G} \,\}|}{B}\right). \tag{37}$$

The acceleration is given by Equation (38).

$$a = \frac{\sum_m (\bar{\theta} - \theta_{(m)})^3}{6\left(\sum_m (\bar{\theta} - \theta_{(m)})^2\right)^{3/2}}. \tag{38}$$

The adjusted levels are given by Equation (39), where ± stands for the lower and the upper level; the limits of the interval are the quantiles of the replicas at those two levels.

$$\omega_{\pm} = \Phi\left(z_0 + \frac{z_0 + z_{\pm}}{1 - a\,(z_0 + z_{\pm})}\right), \qquad z_{\pm} \in \{\, z_{\alpha/2},\ z_{1-\alpha/2} \,\}. \tag{39}$$

The log-log slope of the benchmark over its certified combinations is a least-squares fit of the logarithm of the median exact time on the logarithm of the product of sensors and lots, as Equation (40) gives.

$$\ln T_{p,n} = a_0 + \gamma \ln(p\,n) + \varepsilon, \qquad T_{p,n} = \text{median exact time}. \tag{40}$$

The interval of the slope is the t interval of Equation (41), where the standard error (SE) of the estimate enters, and ν is the number of certified combinations minus two.

$$\hat{\theta} \pm t_{1-\alpha/2,\,\nu}\, \mathrm{SE}(\hat{\theta}). \tag{41}$$

**What was executed.** The exact procedure did not certify the full table within its time limit, so the planned primary analysis cannot be run as specified: its 200 replicas would each require a certified optimum. We executed a pilot consisting of the point estimate and the first four replicas of the planned bootstrap, each with the same time limit and in parallel. We then ran an exploratory analysis that keeps the planned structure (the same replicas and secondary analyses) but replaces the certified minimum by the two verified bounds of Section 4: the greedy cover, which bounds G from below, and the linear relaxation, which bounds it from above. Lower bounds on G are certified without the solver, so the exploratory positive on G is a statement about a verified lower bound, not about the minimum.

After the freeze we added exploratory controls: 1,000 random orders of the sensors, an mRMR order, and a cross-validated logistic check (L2 penalty of 1, with penalties of one tenth and ten as sensitivity checks, standardized values, t intervals over the five folds) with choices made inside the training folds; the plan's optional single random set of the same size was not used. The mRMR order adds, at each step, the sensor with the largest mutual information with the outcome o minus the mean of its mutual information with the sensors already chosen, as Equation (42) gives, where the mutual information between sensors is that of Equation (6) and ties go to the smaller index.

$$s_{m+1} = \arg\max_{s \notin S_m} \left[ I(c_s;\, o) - \frac{1}{m} \sum_{t \in S_m} I\,(c_s;\, c_t) \right], \qquad S_m = \{s_1, \ldots, s_m\}. \tag{42}$$

The logistic check fits, on each training fold, the penalized log-likelihood of Equation (43), where $\eta_i$ is the linear predictor of lot i, with an intercept $\beta_0$ and the coefficients of the standardized predictors, and where the penalty applies to the coefficients but not to the intercept.

$$\hat{\beta} = \arg\min_{\beta} \sum_i \left[\log(1 + e^{\eta_i}) - o_i\, \eta_i\right] + \lambda \sum_{j \geq 1} \beta_j^2. \tag{43}$$

The predictors are standardized with training statistics only, as Equation (44) states: missing values are first imputed by the training median, and the training mean and standard deviation are then applied to the training and the test blocks alike, with a zero standard deviation replaced by one.

$$\tilde{x}_{ij} = \frac{\hat{x}_{ij} - \mu_j}{sd_j}. \tag{44}$$

Out-of-sample discrimination is measured by the area under the curve, which is the Mann–Whitney statistic of the test scores $\hat{s}$ with ties counted as one half, as Equation (45) defines, where $n_F$ and $n_P$ are the numbers of failing and passing test lots.

$$\mathrm{AUC} = \frac{1}{n_F\, n_P} \sum_{i:\, o_i = 1} \sum_{k:\, o_k = 0} \left[\mathbf{1}(\hat{s}_i > \hat{s}_k) + 1/2\ \mathbf{1}(\hat{s}_i = \hat{s}_k)\right]. \tag{45}$$

The paired difference of each fold is the AUC of the cover minus the median AUC of the twenty random sets of the same size, as Equation (46) gives, and its interval is the t interval of Equation (41) over the

five folds.

$$d_f = \mathrm{AUC}_f^{\mathrm{cov}} - \mathrm{median}_e\, \mathrm{AUC}_{f,e}^{\mathrm{rnd}}, \qquad f = 1, \ldots, 5. \tag{46}$$

H3 (the greedy size within $H_d$ and 5 % of the exact minimum) cannot be decided as planned, because that minimum is not certified here; only the bound $H_d$ of Equation (16) is checked. The exact procedure's working set takes violated pairs in index order, whereas the plan takes the least covered ones; this changes the iterations, not the optimum at convergence. After the freeze we also ran an exploratory check on all 590 columns, including the 116 constant sensors that the plan excludes. After the freeze we changed the seed of the multiverse's block-bootstrap replicas, for all its specifications, to the seed of the exploratory bootstrap, so that the primary specification uses the same replicas. The plan reserved a separate seed and did not require this; the first plan amendment records the change. The plan asks for five repetitions of each combination and one discarded warm-up run. Only the combination with 50 sensors and 400 lots was repeated five times, and every repetition certified; the others ran once, since each repetition would have cost the time limit again. No warm-up run was discarded. Table V lists the deviations of this section and of the secondary analyses, with the place where each is declared.

**Table V. Deviations from the plan, what was run instead, and where each is declared**

| Plan | What was executed | Where it is declared |
|---|---|---|
| Primary analysis with 200 certified replicas | Pilot: the point estimate and 4 replicas, none certified; the primary analysis was not run | Results, certification |
| H2 with a certified minimum on each training fold | Greedy cover on the five folds, with a rule of our own: a positive when the lower limit is at least zero | Methods, secondary analyses |
| Holm correction over H1 and H2 | Not applied: the confirmatory family has no p-values; each decision comes from its interval | Methods, secondary analyses |
| V2 and V3 counted with the confirmatory family | Counted as exploratory, following the family definitions | Methods, secondary analyses |
| Working set with the least covered pairs | Pairs taken in index order; this changes the iterations, not the optimum at convergence | Methods, what was executed |
| One random set of the same size | 1,000 random orders, an mRMR order and a cross-validated logistic check (exploratory) | Methods, what was executed |
| H3: greedy size within $H_d$ and five percent of the exact minimum | Not decidable: the minimum is not certified; only the bound $H_d$ is checked | Methods, what was executed |
| Pool of non-constant sensors | All columns checked too (exploratory), including the constant sensors | Methods, what was executed |
| Blocks of fifty lots | Blocks of twenty-five and of one hundred lots (exploratory), with the same seed | Methods, what was executed |
| Seed of the multiverse replicas fixed in the plan | Changed after the freeze, to the seed of the exploratory bootstrap | Methods, what was executed |
| Five repetitions per combination and one discarded warm-up | Five repetitions only for fifty sensors and four hundred lots; no warm-up | Methods, what was executed |
| Benchmark run with a process pool | Discarded: it stopped delivering its last tasks and wrote its results only at the end | Methods, secondary analyses |
| Permutations of V2 (twenty, fixed seed) | Extended to 1,000 permutations (exploratory) | Methods, validity checks |

Each row states a planned element, what was executed in its place, and the place where the paper declares it. The confirmatory and exploratory families are kept apart as the plan defines them.

**Software and hardware.** The analysis is written in Python, with NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020) and pandas (McKinney, 2010). The milp and linprog functions of SciPy use the HiGHS solver, whose parallel dual simplex is described by Huangfu and Hall (2017); the figures use matplotlib (Hunter, 2007). The first plan amendment sets the Basic Linear Algebra Subprograms and OpenMP threads to one; HiGHS manages its own, so the timings include its parallelism. Figure 2 summarizes the pipeline.

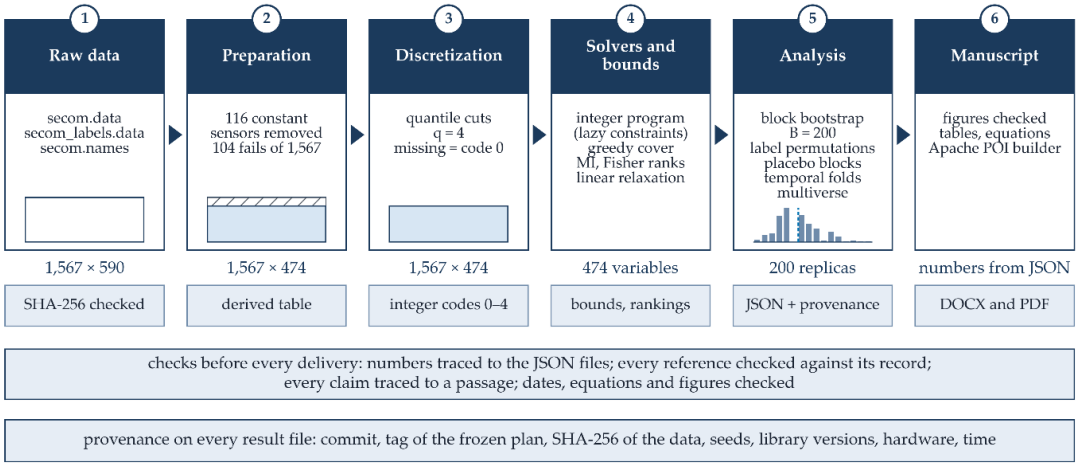



**Figure 2. Pipeline of the study, from left to right;**

each numbered stage lists what it does. The glyphs are drawn to scale, with lots across and sensors up: the raw data have 590 sensors (outline), the preparation and discretization stages keep 474 of them (blue), and the hatched strip is the 116 constant sensors that are removed. The analysis stage shows a histogram of the lower bound $G_{inf}$ over its 200 block-bootstrap replicas, scaled to the tallest bar; the dashed line is their median. Below each stage, the size of what it passes on and, in grey, what it writes. The bottom bands list the checks and the provenance fields.

Table VI lists the seeds, limits, sizes, software and machine of the study.

**Table VI. Seeds, limits, sizes, software and machine**

| Item | Value |
|---|---|
| Seed, block bootstrap | 20261101 |
| Seed, permutations of V2 | 2 |
| Seed, multiverse sample | 3 |
| Seed, random orders | 1 |
| Time limit per exact solve | 600 s |
| Time limit per combination, small grid | 900 s |
| Concurrent processes in the benchmark | 4 |
| Replicas of the planned bootstrap | 200 |
| Block length of the planned bootstrap | 50 lots |
| Resamples of the five-fold interval (H2) | 2,000 |
| Permutations (exploratory) | 1,000 |
| Penalties of the logistic check | 0.1, 1.0 and 10 |
| Discretization of each sensor | quartiles of the observed readings, with missing values as a category |
| Derived data table | digest recorded in each result file |
| Python | 3.11.7 |
| NumPy, SciPy and pandas | 1.26.2, 1.11.4 and 2.1.4 |
| Processor | 8 cores, 16 logical threads |
| Memory visible to the system | 15.6 GB |
| Operating system | Windows, 10.0.26200 |

Values recorded in the result files and in the plan. The processor is described by its number of cores and threads, not by its model.

**Exploratory additions.** After the freeze, the paper adds exploratory analyses from the same data, seeds and code: the worked example of Section 4, the bracket on the minimum, the sensors chosen by each order and the random orders, the out-of-sample AUC, the failing readings by block and the composition of the bootstrap replicas. Each is labelled exploratory in its figure or table; none is part of the confirmatory analysis.

**7. RESULTS**

**Certification.** The exact procedure was run on the full table, in the first repetition of the benchmark, with a limit of 600 s per solve. It stopped without a certificate (Table VII). The primary analysis needs 200 certified replicas; at the pilot's 1,173 s for a batch of five problems, and if every batch takes as long, the point and the replicas would take about 13.4 h (Equation (47)), so it was not run and its confirmatory outcome is not decided.

$$H = \left\lceil \frac{1 + B}{5} \right\rceil \frac{t_p}{3{,}600}. \tag{47}$$

Table VII. Certificates and bounds on the minimum number of sensors on the full SECOM table

| Quantity | Value | How it is obtained |
|---|---|---|
| Exact procedure, full table, limit of 600 s per solve | not certified after 6 iterations; dual bound 2.0; 1,342.8 s | HiGHS through scipy |
| Pilot: point estimate and 4 replicas | 0 of 5 certified | HiGHS through scipy |
| Greedy cover (upper bound on the optimum) | 7 sensors | verified cover |
| Linear relaxation (lower bound on the optimum) | $z_{LP} = 1.6265$ | linear program |
| Ranking by mutual information / by Fisher score | 17 / 18 sensors | sensors added in rank order |
| Ranking by mRMR (redundancy-aware) | 14 sensors | sensors added in mRMR order |
| Random orders: median, and the central interval of 1,000 orders | 10 sensors, between 9 and 14 | random permutations of the sensors |
| Local search for a cover of six sensors | not found: at least 5 pairs uncovered in 200 restarts | simulated annealing over swaps |
| Local search for a cover of seven sensors | found in 3 restarts | verified cover |
| Optimum | between 2 and 7 sensors | bracket from the bounds above |

Exact procedure: iterations and dual bound at termination. Cover and relaxation bounds are computed on the full table. The random orders and the local searches are exploratory controls.

**Verified bounds.** A greedy cover of 7 sensors separates all 152,152 pairs, since every failing–passing pair is separable here. The ranking by mutual information needs 17 sensors, the Fisher ranking needs 18, and the better of the two gives $k_{rank} = 17$. With sixteen sensors the mutual-information ranking leaves one pair uncovered, so its size rests on a single pair. The lower bound on the excess of the ranking is $G_{inf} = 1.42$ (Equation (28)), and the upper bound is $G_{sup} = 9.46$ (lower bounds and lower limits truncated, upper ones rounded up). Since the optimum is an integer, the tighter upper bound, with 2 in place of $z_{LP}$, is $(k_{rank} - 2)/2 = 7.50$. The greedy guarantee of Equation (16) is also checked on this table: since $z_{LP}$ is at most the optimum, the inequality $7 \le H_d\, z_{LP} = 19.9$ implies $7 \le H_d\, \mathrm{OPT}$; since $H_d$ is large for this table, the check is weak. Figure 3a shows how each order separates the pairs, with the controls.

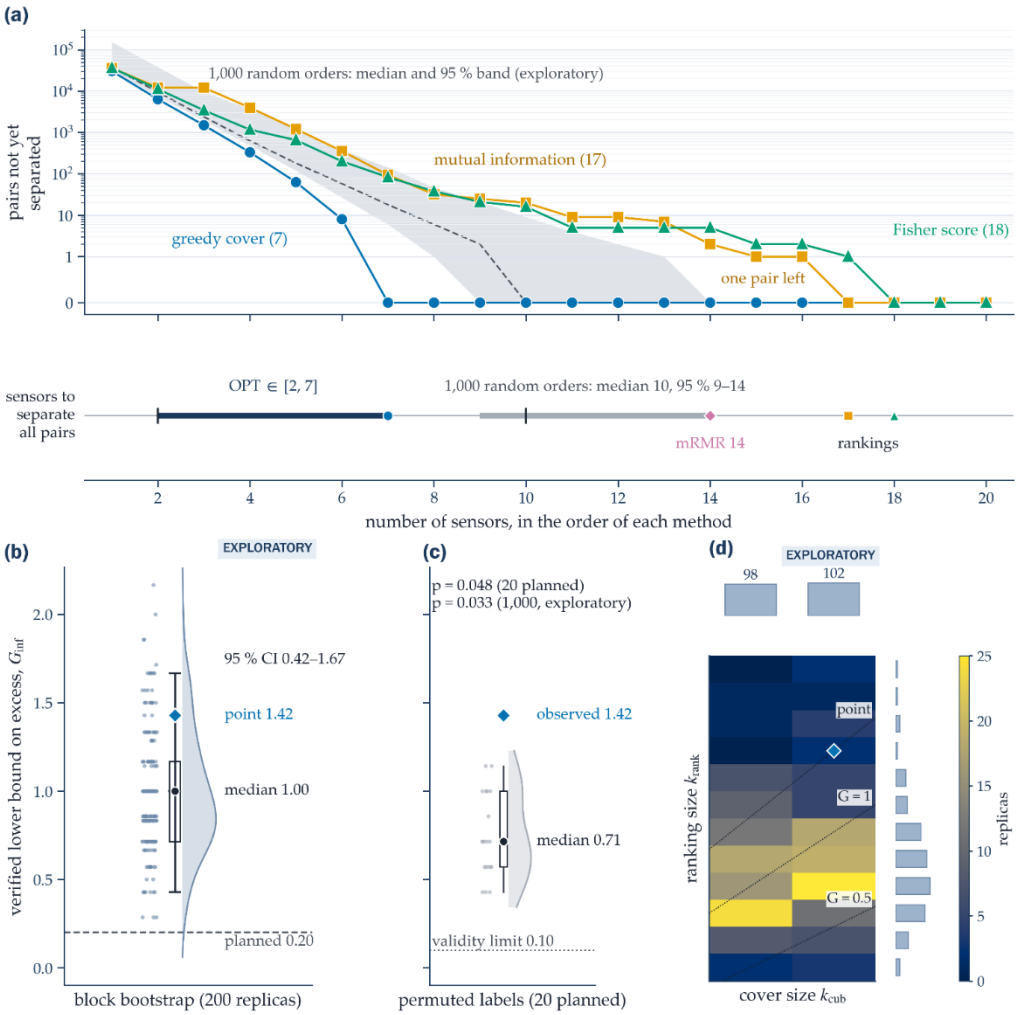



**Figure 3. Verified bounds and controls on the SECOM table. (a) Separable pairs not yet separated after each sensor of the greedy cover (circles), the mutual-information ranking (squares) and the Fisher ranking (triangles), on a symmetric logarithmic scale, linear between 0 and 1; in parentheses, the size at which each order separates all pairs. The strip below gives the optimum (thick bar, from 2 to 7), the central 95 % of the 1,000 random orders (gray bar, with its median) and the mRMR order (diamond); the random orders and the mRMR order are exploratory controls. (b) Exploratory lower bound $G_{inf}$ over 200 block-bootstrap replicas: points, density, quartile box, median, 95 % percentile whiskers, point estimate (blue diamond) and planned threshold (dashed). (c) The same bound with 20 permuted outcomes: density, quartile box, observed value (blue diamond), validity limit (dotted) and one-sided permutation p-values for the planned permutations and for an exploratory run of 1,000. (d) Exploratory: the 200 replicas by cover size and ranking size, the colour giving the count; the dotted lines are constant excess, (ranking size minus cover size) over cover size. Lower bounds are truncated to two decimals.**

**Table VIII lists the instances whose optimum is certified.**

**Table VIII. Instances with a certified optimum, with the bounds and the greedy cover (exploratory)**

| Sensors (pp) | Lots (nn) | OPT | Greedy | zLPzLP | ⌈zLP⌉⌈zLP⌉ | Greedy / OPT | OPT / zLPzLP | Iter. | dd | HdzLPHdzLP |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 100 | 5 | 5 | 4.00 | 4 | 1.00 | 1.25 | 6 | 1,112 | 30.3 |
| 10 | 200 | 5 | 6 | 4.66 | 5 | 1.20 | 1.07 | 4 | 3,818 | 41.1 |
| 10 | 400 | 7 | 7 | 7.00 | 7 | 1.00 | 1.00 | 5 | 14,169 | 70.9 |
| 15 | 100 | 4 | 5 | 3.04 | 4 | 1.25 | 1.31 | 19 | 1,112 | 23.1 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 200 | 5 | 6 | 3.33 | 4 | 1.20 | 1.50 | 19 | 3,818 | 29.4 |
| 15 | 400 | 6 | 7 | 4.50 | 5 | 1.17 | 1.33 | 6 | 14,169 | 45.6 |
| 20 | 100 | 4 | 5 | 2.38 | 3 | 1.25 | 1.67 | 24 | 1,116 | 18.1 |
| 20 | 200 | 5 | 5 | 2.49 | 3 | 1.00 | 2.00 | 15 | 3,851 | 22.0 |
| 20 | 400 | 6 | 7 | 2.97 | 3 | 1.17 | 2.02 | 37 | 14,301 | 30.1 |
| 25 | 100 | 4 | 5 | 2.21 | 3 | 1.25 | 1.81 | 32 | 1,116 | 16.7 |
| 25 | 200 | 5 | 5 | 2.36 | 3 | 1.00 | 2.11 | 18 | 3,851 | 20.9 |
| 25 | 400 | 6 | 7 | 2.75 | 3 | 1.17 | 2.18 | 25 | 14,301 | 27.9 |
| 30 | 100 | 4 | 4 | 2.08 | 3 | 1.00 | 1.92 | 12 | 1,116 | 15.8 |
| 30 | 200 | 5 | 5 | 2.29 | 3 | 1.00 | 2.18 | 16 | 3,851 | 20.2 |
| 30 | 400 | 6 | 6 | 2.56 | 3 | 1.00 | 2.34 | 57 | 14,301 | 25.9 |
| 50 | 400 | 6 | 6 | 2.25 | 3 | 1.00 | 2.66 | 105 | 14,301 | 22.9 |

Sixteen instances: fifteen of the small grid and one of the main benchmark. Equation (16) and Equation (21) hold in every row. Lower bounds are truncated, so the bounds shown do not exceed the values they bound. Iter. counts the iterations of the exact procedure; d is the largest number of pairs that one sensor separates (Equation (14)); the last column, $H_d\, z_{LP}$, is the value that the greedy size must not exceed (Equation (16)).

Table IX gives the quantities behind the bracket on the minimum on the full table. Figure 4 shows the bracket from three sides: the optimum over the linear bound in each certified repetition, the weights of the linear solution, and the number of sensors that separate each pair.

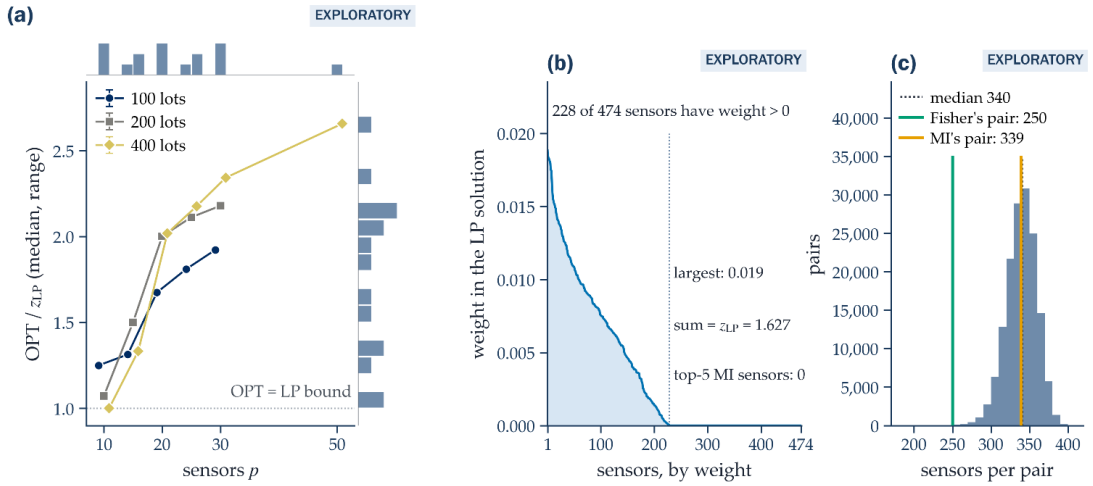



Figure 4. Why the bracket on the minimum is wide (exploratory). (a) The optimum divided by the bound of the linear relaxation, in each certified repetition (five per combination) of the sixteen combinations of the two grids, by number of sensors p and lots n; each repetition is a dot, the symbol is the median and the bar the range over repetitions, and the horizontal line at one marks the case where the bound equals the optimum. The ratio grows with the number of sensors. Above (a), the number of certified repetitions at each sensor count; to the right of (a), the distribution of the ratio over all certified repetitions, the same ones plotted as points. (b) The weight that the optimal linear solution of the full table gives to each sensor, sorted; 228 of the 474 sensors carry weight, and the five highest-ranked sensors by mutual information carry none; the weights sum to the bound of the linear relaxation, which is spread over many sensors. (c) The number of

sensors that separate each separable pair; the dotted vertical line is the median, and the two solid coloured lines are the pairs that decide the Fisher and the mutual-information rankings. Weights and counts are computed for this analysis and are exploratory.

Table IX. The bracket on the minimum for the full table (exploratory)

| Quantity | Value | How it is obtained |
|---|---|---|
| Trivial bound on $z_{LP}$: 474 / 182 | 2.60 | every pair has at least 182 separating sensors, so setting each $y_s$ to 1 / 182 is feasible |
| $H_d$ with d = 121,736 | 12.29 | harmonic number of the largest separating set |
| Separating sensors per pair: minimum / median / maximum | 182 / 340 / 409 | over the 152,152 separable pairs, missing values as a category |
| Separating sensors per pair: 1st and 99th percentiles | 287 / 380 | percentiles of the same counts |
| Pairs separated by a sensor: median (share of separable pairs) | 114,347 (75 %) | over the 474 sensors of the pool |
| Pairs separated by the largest sensor (d) | 121,736 | first sensor of the greedy cover |
| Relaxation: sensors with positive weight | 228 | solution returned by the solver |
| Relaxation: largest weight | 0.019 | solution returned by the solver |
| Relaxation: pairs in the final constraint set | 3,669 | out of 152,152 separable pairs |

Every quantity is computed on the full table. The separating counts run over all separable pairs, with missing values as a category.

**Table X gives the seven sensors of the greedy cover, in order of choice.**

**Table X. The seven sensors of the greedy cover, in order of choice (exploratory)**

| Step | Sensor (index in the pool) | New pairs separated | Cumulative share of pairs (%) | Rank in the mutual-information order |
|---|---|---|---|---|
| 1 | 437 | 121,736 | 80.01 | 30 |
| 2 | 450 | 24,054 | 95.82 | 73 |
| 3 | 182 | 4,866 | 99.02 | 162 |
| 4 | 84 | 1,164 | 99.78 | 225 |
| 5 | 262 | 269 | 99.96 | 276 |
| 6 | 173 | 55 | 99.99 | 314 |
| 7 | 5 | 8 | 100.00 | 360 |

New pairs are those first separated at each step. The index is the position of the sensor in the pool and has no physical meaning. The rank is that of the sensor in the mutual-information order.

Table XI gives the sensors shared by the chosen sets. Figure 5 shows which sensors each order chooses and whether they repeat across folds. The overlaps are exploratory and descriptive: they are computed on the full table and carry no interval.

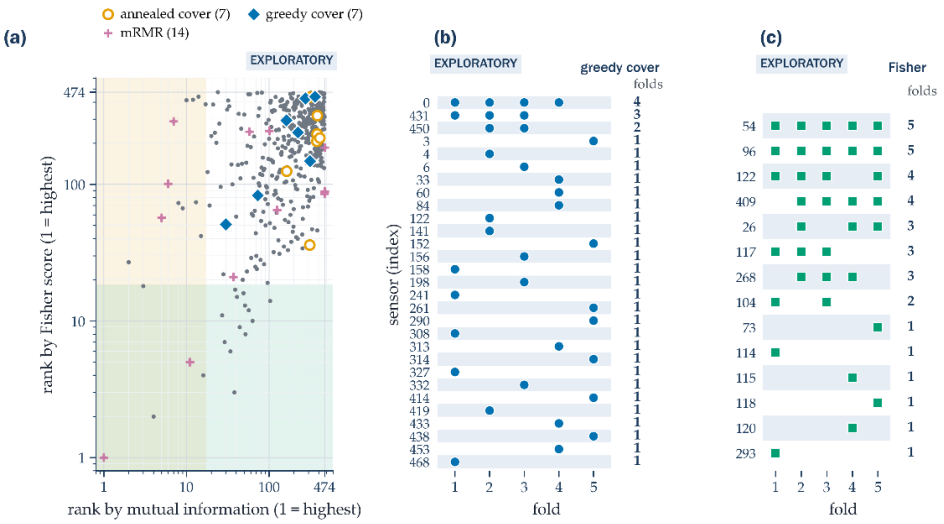



**Figure 5. Which sensors each order chooses, and whether they repeat across folds (exploratory). (a) The 474 sensors by their rank in mutual information (horizontal) and in the Fisher score (vertical), on logarithmic axes. The greedy cover (blue diamonds), the annealed cover (open orange circles) and the mRMR order (reddish-purple crosses) are marked; the shaded column and row are the top 17 by mutual information and the top 18 by Fisher score. (b) The sensors of the greedy cover in each fold, one row per sensor, sorted by the number of folds in which it appears (right column). (c) The top seven sensors by Fisher score in each fold, in the same format; Fisher was the ranking chosen in every fold. The greedy cover and the annealed cover share no sensor.**

Table XI. Sensors shared by the chosen sets (exploratory)

| Shared sensors | Greedy cover, 7 sensors | Annealing, 7 sensors | Mutual information, 17 sensors | Fisher score, 18 sensors | mRMR, 14 sensors |
|---|---|---|---|---|---|
| Greedy cover, 7 sensors | 7 | 0 | 0 | 0 | 1 |
| Annealing, 7 sensors | 0 | 7 | 0 | 0 | 0 |
| Mutual information, 17 sensors | 0 | 0 | 17 | 5 | 5 |
| Fisher score, 18 sensors | 0 | 0 | 5 | 18 | 2 |
| mRMR, 14 sensors | 1 | 0 | 5 | 2 | 14 |

Each entry counts the sensors two sets have in common; the diagonal gives the size of each set. The two sets of seven sensors share none.

Table XII gives the number of sensors each order needs for each share of pairs, as Equation (48) defines.

$$k_r(\tau) = \min\{\, k : \sigma_C(S_k^r) \geq \tau \,\}, \quad S_k^r = \text{the first } k \text{ sensors of the order } r. \tag{48}$$

Table XII. Sensors needed to separate a share of the separable pairs (exploratory)

| Order | 50 % | 90 % | 95 % | 99 % | 100 % |
|---|---|---|---|---|---|
| Greedy cover | 1 | 2 | 2 | 3 | 7 |
| Mutual information | 1 | 2 | 4 | 5 | 17 |
| Fisher score | 1 | 2 | 3 | 4 | 18 |

For each order, the smallest number of sensors whose separated share reaches the share in the column.

**Exploratory excess over the bootstrap.** Across 200 block-bootstrap replicas, the lower bound on the excess has a 95 % percentile interval of 0.42 to 1.67 (endpoints rounded outward; used descriptively; a bias-corrected and accelerated interval (Efron, 1987) is 1.00 to 2.17 but rests on the largest replicate), and every

replica yields a verified cover. The bootstrap distribution lies below the point estimate: its median is 1.00, and only 25 of the 200 replicas exceed the point estimate. Resampling repeats lots; the interval need not be centered on the point estimate. In one specification of the multiverse (q = 5, missing values as a category), the point estimate lies above its interval. The planned threshold of 0.20 lies below the interval, so the lower limit clears it. Blocks of 25 and 100 lots give intervals of 0.42 to 1.67 and 0.42 to 1.84, and their lower limits also clear the planned threshold. The interval is much wider than the 0.10 precision that the plan requires for a conclusion, so the result is reported as a bound, not as a decision. The exploratory rule, with the planned threshold, gives a positive, but the validity checks fail, so the positive cannot stand. Table XIII gives the intervals of each resampling scheme.

**Table XIII. Lower bound on the excess under each resampling scheme (exploratory)**

| Resampling scheme | Replicas | Median | 95 % interval |
|---|---|---|---|
| Blocks of 50 lots (plan) | 200 | 1.00 | 0.42 to 1.67 |
| Blocks of 25 lots (exploratory) | 200 | 1.00 | 0.42 to 1.67 |
| Blocks of 100 lots (exploratory) | 200 | 0.86 | 0.42 to 1.84 |
| Stratified by class (exploratory) | 1,000 | 1.00 | 0.42 to 1.84 |
| BCa, block jackknife of 32 blocks (diagnostic) | 200 | not used | 1.00 to 2.17 |
| | | | |
| | | | |

Intervals are rounded outward: lower limits are truncated and upper limits rounded up. Only the plan's scheme is confirmatory; the others are exploratory, and the BCa interval is a diagnostic that rests on the largest replicate.

**Out-of-sample.** On the five temporal folds, the verified cover separates 99.98 % of the out-of-sample separable pairs and the Fisher ranking of the same size, which separated more training pairs in every fold, separates 99.47 %. The pooled difference, given by Equation (36), is 0.51 percentage points, with a 95 % interval from 0.007 to 1.014 points. Most of the difference comes from the first fold, which accounts for 148 of the 151 net pairs in favor of the cover, and the lower limit is close to zero. No fold favors the ranking. The interval describes these five folds rather than independent evidence of an effect. Both sets are near the ceiling. Under the rule we used for this exploratory computation (a positive when the lower limit of the interval is at least zero), the rule returns a positive. The rule is not in the plan, the computation is not the planned one, and the validity checks fail, so we report H2 as exploratory and not as a confirmed effect.

Table XIV gives the bootstrap replicas by cover and ranking size.

Table XIV. Bootstrap replicas by ranking size and cover size (exploratory)

| Ranking size $k_{rank}$ | Replicas | Cover of 6 sensors | Cover of 7 sensors | Fisher attains $k_{rank}$ | Mutual information attains $k_{rank}$ | Tier |
|---|---|---|---|---|---|---|
| 9 | 5 | 2 | 3 | 2 | 3 | 0 |
| 10 | 15 | 8 | 7 | 8 | 6 | 1 |
| 11 | 35 | 24 | 11 | 17 | 18 | 0 |
| 12 | 41 | 16 | 25 | 24 | 11 | 6 |
| 13 | 37 | 18 | 19 | 21 | 14 | 2 |
| 14 | 30 | 12 | 18 | 8 | 14 | 8 |
| 15 | 14 | 9 | 5 | 8 | 4 | 2 |
| 16 | 12 | 7 | 5 | 4 | 7 | 1 |

| Ranking size $k_{rank}$ | Replicas | Cover of 6 sensors | Cover of 7 sensors | Fisher attains $k_{rank}$ | Mutual information attains $k_{rank}$ | Tier |
|---|---|---|---|---|---|---|
| 17 | 2 | 0 | 2 | 0 | 2 | 0 |
| 18 | 5 | 1 | 4 | 0 | 5 | 0 |
| 19 | 2 | 1 | 1 | 0 | 1 | 1 |
| 20 | 2 | 0 | 2 | 1 | 1 | 0 |
| Total | 200 | 98 | 102 | 93 | 86 | 21 |

Each row counts the replicas whose better ranking needs that many sensors. A tie means that both rankings need the same number.

Table XV gives the five folds.

**Table XV. The five temporal folds of the out-of-sample comparison (exploratory)**

| Fold | Lots | Failures | Test pairs | Net pairs (cover − ranking) | AUC, cover | AUC, random (median) | Cover minus random |
|---|---|---|---|---|---|---|---|
| 1 | 314 | 44 | 11,880 | 148 | 0.527 | 0.515 | 0.011 |
| 2 | 314 | 21 | 6,153 | 0 | 0.586 | 0.516 | 0.070 |
| 3 | 313 | 11 | 3,322 | 1 | 0.496 | 0.578 | −0.083 |
| 4 | 313 | 11 | 3,322 | 1 | 0.520 | 0.556 | −0.037 |
| 5 | 313 | 17 | 5,032 | 1 | 0.559 | 0.541 | 0.018 |
| Total or mean | 1,567 | 104 | 29,709 | 151 | 0.537 | 0.541 | −0.004 |

Test-block AUC of the cover and of a random set of the same size, whose median over twenty draws is shown. Net pairs count the test pairs that the cover separates and the ranking does not, minus the reverse.

Table XVI gives the sensitivity to the penalty.

**Table XVI. Sensitivity of the out-of-sample comparison to the penalty (exploratory)**

| Penalty (lambda) | AUC, greedy cover | AUC, random, median | AUC, mutual information | Cover minus random |
|---|---|---|---|---|
| 0.1 | 0.535 | 0.543 | 0.699 | −0.008 |
| 1.0 | 0.537 | 0.541 | 0.702 | −0.004 |
| 10.0 | 0.543 | 0.539 | 0.707 | 0.005 |

Penalties of the logistic model; the primary value is one. The cover is compared with a random set and with a mutual-information set of the same size.

Table XVII gives the AUC at other set sizes. Figure 6 shows the same curves with their spread across folds, and each fold on its own.

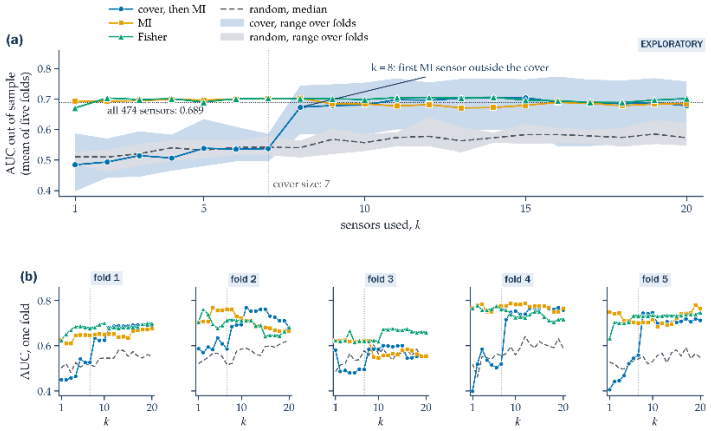



**Figure 6. Out-of-sample AUC by number of sensors (exploratory).**

(a) The mean over the five temporal folds of the AUC on the test block, for the greedy cover followed by mutual information (its first seven sensors are the cover), for mutual information, for the Fisher score and for the median over random sets of each size; the shading is the range over folds for the cover and for the random sets, and the dotted line is the AUC with all 474 sensors. The cover curve rises at eight sensors, when the first mutual-information sensor outside the cover enters. The dotted vertical line marks seven sensors, the size of every fold's cover. (b) The same four orders in each fold, so that the pattern of (a) can be read fold by fold.

**Table XVII. Out-of-sample AUC by number of sensors (exploratory)**

| Sensors (k) | Greedy cover order | Mutual information | Fisher score | Random, median |
|---|---|---|---|---|
| 1 | 0.484 | 0.694 | 0.671 | 0.511 |
| 2 | 0.493 | 0.693 | 0.703 | 0.510 |
| 3 | 0.514 | 0.695 | 0.698 | 0.521 |
| 7 | 0.537 | 0.702 | 0.702 | 0.541 |
| 8 | not defined | 0.703 | 0.701 | 0.541 |
| 20 | not defined | 0.685 | 0.701 | 0.573 |

Means over the five folds. Every fold's cover has seven sensors, so the cover order is not defined beyond that size.

**Validity checks.** With permuted labels the median lower bound is 0.71, far above the planned limit of 0.10. The observed excess is larger than every one of the planned permuted values (permutation p = 0.048, as Equation (33) gives; with 1,000 permutations, exploratory, p = 0.033). With the placebo blocks of Equation (34) the median is 1.25, again above 0.10; with the true labels, the median is 1.00. The placebo does not reduce the excess: its block values exceed the true ones in 3 of the 5 blocks. So V2 and V3 fail; both are computed on the lower bound $G_{inf}$, since G needs the exact minimum. The planned permuted values range from 0.42 to 1.14, against an observed 1.42, so part of the excess does not depend on which lots failed. V1 fails on the pilot, which certified 0 of its 4 replicas against the 95 % required. V4 is an automated test in the code's test suite that checks that the cuts, medians, cover and ranking of the out-of-sample analysis use only the training folds, and it passes. Table XVIII lists the blocks, with their true and placebo labels.

**Table XVIII. Blocks of the placebo check, with the true labels and the placebo labels of each block (exploratory)**

| Block | Lots | Failures, true labels | G_inf, true labels | Failures, placebo | G_inf, placebo |
|---|---|---|---|---|---|
| 1 | 3 14 | 44 | 1.00 | 21 | 0.40 |

| Block | Lots | Failures, true labels | G_inf, true labels | Failures, placebo | G_inf, placebo |
|---|---|---|---|---|---|
| 2 | 3 14 | 21 | 1.20 | 11 | 1.25 |
| 3 | 3 13 | 11 | 0.60 | 11 | 1.25 |
| 4 | 3 13 | 11 | 0.75 | 17 | 1.40 |
| 5 | 3 13 | 17 | 1.25 | 44 | 0.40 |
| Median | | | 1.00 | | 1.25 |

Each placebo block takes the failure count of the next block, cyclically. The lower bound is truncated to two decimals.

**Multiverse.** Across the six specifications of the number of intervals and the treatment of missing values, the lower limit of each interval exceeds the planned threshold, and the point estimates lie between 1.00 and 1.83. The primary specification (q = 4 intervals, missing values as a category) uses the same block-bootstrap replicas as the exploratory analysis, so its interval, 0.42 to 1.67, is the one in Figure 3; both are exploratory. The permutation and placebo controls were computed for the primary specification only (Figure 3c and Figure 7b). Figure 7c shows where the failing readings lie in recording order: the first five runs of 50 lots hold a larger share of the failing readings than of the lots.

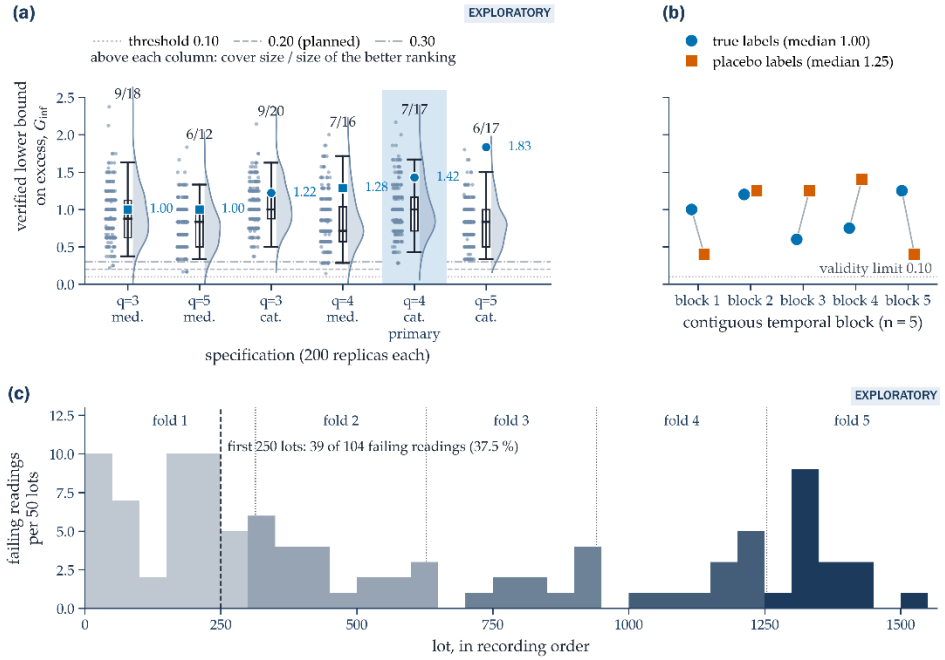



Figure 7. Lower bound on the excess across specifications and with placebo labels. (a) Exploratory (the multiverse): the six specifications of the number of intervals q and of the treatment of missing values, ordered by value: 200 block-bootstrap replicas (gray points, shown as a shaded density with a box for the quartiles and the median), 95 % percentile interval (bar with caps), point estimate (circle: missing values as a category, cat.; square: median imputation, med.) and, above each column, the cover size over the size of the better of the two rankings; the shaded column is the primary specification. Dotted, dashed and dash-dot lines mark the thresholds 0.10, 0.20 and 0.30; the dashed one (0.20) is the planned threshold. Point values are truncated to two decimals. (b) On the same vertical scale, block by block, the bound with true labels (circles) and with placebo labels (squares), joined by a line; one value per block, without an interval; the key gives the median of each series; dotted line: validity limit 0.10. (c) Exploratory: the failing readings in each run of 50 lots, in recording order; the bars are shaded by fold, from lightest (first fold) to darkest (last fold), and the five folds are those of the out-of-sample comparison, their boundaries are dotted, and the dashed line marks the end of the first five runs, which hold a larger share of the failing readings than of the lots.

**Decisions.** Table XIX sets out every decision and validity check, with its family and its status.

**Table XIX. Decisions, validity checks and their status**

| Check | Family | Rule | Result | Status |
|---|---|---|---|---|
| H1 (primary) | Confirmatory | 200 certified replicas; positive if the lower limit is at least 0.20 and at least 95 % are certified | 0 of 5 problems certified (pilot) | Not concluded |
| H2 (out of sample) | Confirmatory, run as exploratory | Lower limit of the interval at least zero (own rule) | 0.007 to 1.014 percentage points | Positive under our rule; exploratory |
| H3 (greedy size) | Plan | Greedy size within $H_d$ and five percent of the exact minimum | Only $H_d$ checked; the minimum is not certified | Not decidable |
| Multiverse | Exploratory | Lower limit above 0.20 | 6 of 6 specifications | Exploratory |
| V1 | Validity | At least 95 % of replicas certified | 0 of 4 replicas certified | Fails |
| V2 | Validity, exploratory family | Median G under permuted labels below 0.10 | Median 0.71; permutation p = 0.048 with the planned permutations; p = 0.033 with 1,000; 95th percentile of the null 1.29 | Fails |
| V3 | Validity, exploratory family | Median G with placebo labels below 0.10 | Median 1.25, against 1.00 with true labels | Fails |
| V4 | Validity | No leakage between training and test blocks | Automated test of the out-of-sample code | Passes |

Confirmatory and exploratory families are kept apart as the plan defines them. No Holm adjustment is applied, because the confirmatory family has no p-values.

**Scaling.** The median time of the greedy procedure is at most 519.9 ms on any size measured. In the main benchmark, the exact procedure certified every repetition in only 1 of the 11 finished combinations, none with more than 50 sensors; in the other 10, at least one solve reached the time limit of 600 s (Figure 8b). Memory was not the constraint: in the first repetition of each finished combination, the exact procedure's peak memory was at most 104 MB above its base. The benchmark was stopped with 1 combination unfinished, 50 × 800 (sensors × lots). On a smaller grid (open symbols in Figure 8a), where the exact procedure certified every repetition of all 15 combinations, the log-log slope (Equation (40)) of the median exact time against the product of sensors and lots is 2.69 (95 % interval from 0.87 to 4.51, over 15 points); the interval is wide. Across the 80 certified repetitions of both grids, the greedy cover matched the optimum in 40 of them, and in every one it was at most 1.25 times the optimum.

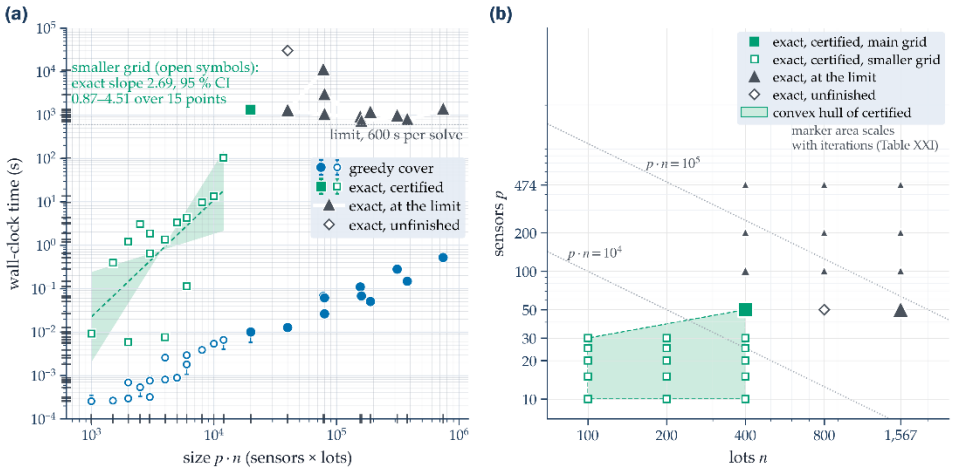



**Figure 8. (a) Wall-clock time of the greedy cover (blue circles) and of the exact procedure**

(green squares if certified; grey triangles if stopped at the limit of 600 s per solve, plotted at the total time of their first repetition), on log-log axes, with short marks on the axes at each measured size and time. Circles and squares are medians over repetitions, with interquartile bars where a combination has more than one repetition. Filled symbols are the main grid and open symbols the smaller grid, where every combination certified; the dashed line is the least-squares fit of the smaller grid, and the band spans the slopes of its 95 % interval through the same centroid. Combinations with the same product p·n share a horizontal position, so their times spread vertically; panel (b) gives their sensors and lots. The diamond is the unfinished combination at its elapsed time, a lower bound. A point above the dotted line is a total over several solves. (b) Frontier map in lots n and sensors p, on log-log axes, with minor grid lines. Green filled squares are certified combinations of the main grid, green open squares certified combinations of the smaller grid, grey triangles combinations stopped at the time limit, and the open diamond the one that did not finish. The shaded region is the convex hull of the certified combinations, computed in logarithmic coordinates; it describes them and is not a bound. The two dotted lines are lines of constant product of lots and sensors, labelled in the figure. The area of the filled squares and grey triangles of the main grid grows with the lazy-generation iterations (Table XXI gives them). Four combinations ran at a time, so times compare sizes rather than isolated runs.

Table XX gives the time of each computation.

**Table XX. Wall-clock time of each computation, in seconds, on the machine of the study**

| Computation | Time (s) | Outcome |
|---|---|---|
| Greedy cover, 474 sensors and 1,567 lots (median) | 0.52 | certified cover |
| Exact, 50 sensors and 400 lots (median of 5 repetitions) | 1,333 | certified in 5 of 5 |
| Exact, 474 sensors and 1,567 lots (first repetition) | 1,343 | limit reached, not certified |
| Exact, 50 sensors and 1,567 lots (first repetition) | 10,837 | limit reached, not certified |
| Pilot: point estimate and 4 replicas | 1,173 | 0 of 5 certified |
| Primary analysis, estimated from the pilot's pace | 13.4 h | not run |
| Greedy cover and both rankings, full curves | 0.83 | exploratory |
| Exploratory bounds (replicas, controls and folds) | 239.8 | exploratory |
| Random orders, mRMR and annealing controls | 5.04 | exploratory |
| Local search for a cover of six sensors | 651.8 | not found |
| Local search for a cover of seven sensors | 9.38 | found |

| Computation | Time (s) | Outcome |
|---|---|---|
| Permutations (1,000) | 913.0 | exploratory |
| Multiverse, six specifications | 943.7 | exploratory |
| Stratified bootstrap (1,000 replicas) | 895.8 | exploratory |
| Block lengths of twenty-five and one hundred lots (200 replicas each) | 413.0 | exploratory |
| Small grid, 15 combinations | 553.4 | certified in 75 of 75 repetitions |

Times depend on the machine and its load, and several benchmark combinations ran at a time. Each entry is a single run unless the entry says otherwise.

Table XXI gives the status and bounds at every size measured.

**Table XXI. Certification status and bounds at each size measured (exploratory)**

| Grid | Sensors (p) | Lots (n) | Status | Iter. | Dual bound | Time (s) | Repetitions | Peak memory (MB) | Greedy (ms) |
|---|---|---|---|---|---|---|---|---|---|
| small | 10 | 100 | certified | 6 | 5 | 0.009 | 5 | 4 | 0.3 |
| small | 10 | 200 | certified | 4 | 5 | 0.006 | 5 | 3 | 0.7 |
| small | 10 | 400 | certified | 5 | 7 | 0.008 | 5 | 4 | 2.6 |
| small | 15 | 100 | certified | 19 | 4 | 0.40 | 5 | 6 | 0.3 |
| small | 15 | 200 | certified | 19 | 5 | 0.65 | 5 | 6 | 0.8 |
| small | 15 | 400 | certified | 6 | 6 | 0.11 | 5 | 4 | 2.9 |
| small | 20 | 100 | certified | 24 | 4 | 1.20 | 5 | 6 | 0.3 |
| small | 20 | 200 | certified | 15 | 5 | 1.35 | 5 | 5 | 0.8 |
| small | 20 | 400 | certified | 37 | 6 | 9.68 | 5 | 7 | 3.9 |
| small | 25 | 100 | certified | 32 | 4 | 3.08 | 5 | 6 | 0.5 |
| small | 25 | 200 | certified | 18 | 5 | 3.31 | 5 | 6 | 0.9 |
| small | 25 | 400 | certified | 25 | 6 | 13.4 | 5 | 8 | 5.4 |
| small | 30 | 100 | certified | 12 | 4 | 1.88 | 5 | 5 | 0.3 |
| small | 30 | 200 | certified | 16 | 5 | 4.29 | 5 | 6 | 1.8 |

| Grid | Sensors (p) | Lots (n) | Status | Iter. | Dual bound | Time (s) | Repetitions | Peak memory (MB) | Greedy (ms) |
|---|---|---|---|---|---|---|---|---|---|
| small | 30 | 400 | certified | 57 | 6 | 102.8 | 5 | 11 | 6.6 |
| main | 50 | 400 | certified | 105 | 6 | 1,333 | 5 | 17 | 10.0 |
| main | 50 | 1,567 | limit reached | 132 | 6 | 10,837 | 1 | 29 | 65.8 |
| main | 100 | 400 | limit reached | 21 | 4 | 1,256 | 1 | 21 | 12.7 |
| main | 100 | 800 | limit reached | 12 | 3 | 1,015 | 1 | 24 | 61.2 |
| main | 100 | 1,567 | limit reached | 12 | 4 | 887.1 | 1 | 38 | 108.9 |
| main | 200 | 400 | limit reached | 11 | 3 | 2,899 | 1 | 25 | 26.2 |
| main | 200 | 800 | limit reached | 5 | 2 | 699.3 | 1 | 26 | 67.6 |
| main | 200 | 1,567 | limit reached | 6 | 2 | 947.8 | 1 | 52 | 279.7 |
| main | 474 | 400 | limit reached | 6 | 2 | 1,147 | 1 | 37 | 50.4 |
| main | 474 | 800 | limit reached | 5 | 2 | 771.4 | 1 | 50 | 147.5 |
| main | 474 | 1,567 | limit reached | 6 | 2 | 1,343 | 1 | 104 | 519.9 |
| main | 50 | 800 | cut before finishing | not recorded | not recorded | at least 30,500 | 1 | not measured | not measured |

Certified rows give the optimum. Limit rows give the dual bound at the time limit and the time of the first repetition. The unfinished combination gives only its elapsed time.

**8. DISCUSSION**

In the data used to choose them, a verified cover of 7 sensors separates every separable pair, while the two rankings we tested need 17 or more; out of sample the difference is small (Section 7). A shortlist built from these rankings therefore asks for more sensors than a short set that separates the same pairs. Covering pairs does not mean predicting failures: out of sample, a cover of this size shows no detectable advantage over random sensors, while the mutual-information ranking does much better (Section 7), so the bound describes pair separation rather than prediction.

The excess is about half as large when the outcome labels are randomly permuted, and the planned validity checks fail. The observed excess is larger than every one of the 20 planned permuted values, so part of it depends on the actual labels; the median permuted value, 0.71, shows that about half of the observed 1.42 is present under random labels, and these data do not show whether that part comes from the covering problem or from the rankings. Univariate rankings can pick sensors that are redundant with one another, and redundant sensors cover the same pairs. If so, the ranking comparison measures how a ranking covers pairs, not how well it finds the sensors that matter for failures. The mRMR order needs 14 sensors, fewer than the better ranking, as this reading predicts. We therefore do not claim that rankings miss the sensors

that explain failures, and the paper's planned confirmatory question remains unanswered.

The exact method did not certify an optimum on this table within the time limit. The gap is large: the linear relaxation gives 1.6265, and the greedy cover is 7. Symmetry among sensors that cover similar pairs is a plausible reason, because the relaxation is weak when many sensors meet every constraint. Stronger formulations could close the gap, and we did not test them; the annealing search of Section 4 found no six-sensor cover, and we do not claim that certification is impossible at this size.

The discretization and the treatment of missing values can change the cover size, and the multiverse keeps the lower limit above the planned threshold in every specification we tried, but other choices are not exhausted. The out-of-sample gap in pairs is positive but small, and near the ceiling, so it shows at most a small practical advantage of the cover over a ranking of the same size.

The bounds describe the pool of 474 sensors that the plan fixed before the analysis. The 116 sensors excluded from that pool each have a single observed value; including them leaves the cover and the better ranking unchanged (Section 5). Table XXII lists the threats to validity, with their consequences and their treatment. The analysis is descriptive. It does not estimate what a sensor reduction would do to the plant, and it makes no causal claim about any sensor. With 20 permutations, the smallest attainable permutation p-value is one divided by the number of permutations plus one (Phipson and Smyth, 2010). At the time limit the exact procedure reports only its dual bound, and it adds no dominance reductions or symmetry breaking of its own.

**Table XXII. Threats to validity, their consequences and their treatment**

| Threat | Consequence | Treatment in this paper |
|---|---|---|
| One dataset from one production process | The bounds describe these lots only | No external data; claims limited to this dataset |
| Pairs share lots, so they are not independent | Intervals may be too narrow; the plan assumes independence, which we declare | Block bootstrap; intervals reported as descriptive |
| The lower bound takes few values (a ratio of integers) | Coarse intervals; the precision of 0.10 is not reached | Reported as a bound, not as a conclusion |
| Missingness mechanism unknown | Missing values may carry information; 39 of 152,152 pairs of the greedy cover are separated only by the missing category | Multiverse over the treatment of missing values |
| Sensor meanings and costs not given | The problem cannot be weighted by cost, and sensors are identified only by index | Objective is the number of sensors; no sensor is interpreted |
| Historical outcome label | The results describe the labels, not the causes of failure | Descriptive analysis with no causal claim |
| Time limit: an unfinished exact run gives only a dual bound | No certified optimum on the full table | Lower bounds certified without the solver; exact runs reported as bounds |

Each threat is stated with its consequence for the results and the treatment applied in this paper. The last row concerns the exact procedure only.

The novelty claim of Section 2 is narrow: in the sources we searched, no work reports a certified exact minimum for this dataset. We did not read the full texts of some related articles, so a complete review could still find one.

**9. CONCLUSIONS**

Selecting the smallest set of sensors that separates failing lots from passing lots is a combinatorial problem, and it is NP-hard. On SECOM, a standard open solver did not certify an optimum within a time limit of 600 seconds per solve, and the planned confirmatory comparison therefore could not be decided. What can be certified is smaller. A verified cover of 7 sensors separates every separable pair; the two rankings we tested need 17 or more, but in an exploratory cross-validation a cover chosen in each training fold shows no detectable advantage over random sets of its size, and the optimum lies between 2 and 7.

In the data, the excess of the better of the two rankings over the minimum is at least 142 % (an exploratory bound, not a decision), and this bound is larger than in any of 20 planned random relabelings (one-sided $p = 0.033$ in an exploratory run of 1,000 relabelings). About half of it remains when the labels are

randomly permuted (as ratios: median 0.71, against 1.42), so part of it does not depend on which lots failed; these data do not show whether that part comes from how the sensors cover pairs or from how the rankings behave. We suggest that a sensor-selection study report a verified cover and a permutation control alongside any ranking, and state its bounds before the comparison.

Two directions follow: closing the gap between the greedy cover and the relaxation with stronger formulations, and testing the design on other manufacturing tables. Neither is done here.

**Declarations**

**Ethics.** The study uses an anonymized public dataset of industrial measurements, and it involves no human participants and no personal data. No ethics approval was sought for this work.

**Funding.** None.

**Conflicts of interest.** None.

**Data availability.** The SECOM dataset is publicly available in the UCI Machine Learning Repository, at https://archive.ics.uci.edu/dataset/179/secom, with DOI 10.24432/C54305 (McCann and Johnston, 2008). The three files used can be downloaded from that page; their SHA-256 digests are listed in the datasheet that accompanies the analysis code. The code and the result files of the analysis are available from the corresponding author.

**REFERENCES**

1. Alqarni, Abdulaziz M.; Robertazzi, Thomas G. 2021. Optimal Signal Selection for Sensors. arXiv preprint. https://doi.org/10.48550/arxiv.2107.01676
2. Angelino, Elaine; Larus-Stone, Nicholas; Alabi, Daniel; Seltzer, Margo; Rudin, Cynthia. 2018. Learning certifiably optimal rule lists for categorical data. Journal of Machine Learning Research 18, 1–78. https://www.jmlr.org/papers/v18/17-716.html
3. Battiti, Roberto. 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5, 537–550. https://doi.org/10.1109/72.298224
4. Bellarmino, Nicolò; Cantoro, Riccardo; Huch, Martin; Kilian, Tobias; Schlichtmann, Ulf; Squillero, Giovanni. 2023. Feature Selection for Cost Reduction In MCU Performance Screening. In 2023 IEEE 24th Latin American Test Symposium (LATS), pp. 1–6. https://doi.org/10.1109/lats58125.2023.10154495
5. Benjamini, Yoav; Hochberg, Yosef. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society Series B (Statistical Methodology) 57, 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x
6. Bertsimas, Dimitris; King, Angela; Mazumder, Rahul. 2016. Best subset selection via a modern optimization lens. The Annals of Statistics 44, 813–852. https://doi.org/10.1214/15-aos1388
7. Bläsius, Thomas; Friedrich, Tobias; Lischeid, Julius; Meeks, Kitty; Schirneck, Martin. 2022. Efficiently enumerating hitting sets of hypergraphs arising in data profiling. Journal of Computer and System Sciences 124, 192–213. https://doi.org/10.1016/j.jcss.2021.10.002
8. Brown, Gavin; Pocock, Adam; Zhao, Ming-Jie; Luján, Mikel. 2012. Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. Journal of Machine Learning Research 13, 27–66. https://www.jmlr.org/papers/v13/brown12a.html
9. Caprara, Alberto; Toth, Paolo; Fischetti, Matteo. 2000. Algorithms for the Set Covering Problem. Annals of Operations Research 98, 353–371. https://doi.org/10.1023/A:1019225027893
10. Chi, Guoyi. 2014. Optimal sensor placement for model-based fault detectability and isolability. Dissertation, Nanyang Technological University. https://doi.org/10.32657/10356/60470
11. Chvatal, V. 1979. A Greedy Heuristic for the Set-Covering Problem. Mathematics of Operations Research 4, 233–235. https://doi.org/10.1287/moor.4.3.233
12. Cover, Thomas M. 1974. The Best Two Independent Measurements Are Not the Two Best. IEEE Transactions on Systems, Man, and Cybernetics SMC-4, 116–117. https://doi.org/10.1109/TSMC.1974.5408535
13. Dash, Manoranjan; Liu, Huan. 2003. Consistency-based search in feature selection. Artificial Intelligence 151, 155–176. https://doi.org/10.1016/S0004-3702(03)00079-1
14. Dedieu, Antoine; Hazimeh, Hussein; Mazumder, Rahul. 2020. Learning Sparse Classifiers: Continuous and Mixed Integer Optimization Perspectives. arXiv preprint. https://doi.org/10.48550/arxiv.2001.06471

15. Efron, B. 1979. Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics 7, 1–26. https://doi.org/10.1214/aos/1176344552
16. Efron, B. 1987. Better Bootstrap Confidence Intervals. Journal of the American Statistical Association 82, 171–185. https://doi.org/10.1080/01621459.1987.10478410
17. Feige, Uriel. 1998. A threshold of ln n for approximating set cover. Journal of the ACM 45, 634–652. https://doi.org/10.1145/285055.285059
18. Fleuret, François. 2004. Fast Binary Feature Selection with Conditional Mutual Information. Journal of Machine Learning Research 5, 1531–1555. https://www.jmlr.org/papers/v5/fleuret04a.html
19. Gainer-Dewar, Andrew; Vera-Licona, Paola. 2017. The Minimal Hitting Set Generation Problem: Algorithms and Computation. SIAM Journal on Discrete Mathematics 31, 63–100. https://doi.org/10.1137/15M1055024
20. Gebru, Timnit; Morgenstern, Jamie; Vecchione, Briana; Vaughan, Jennifer Wortman; Wallach, Hanna; Daumé III, Hal; Crawford, Kate. 2021. Datasheets for datasets. Communications of the ACM 64, 86–92. https://doi.org/10.1145/3458723
21. Guyon, Isabelle; Elisseeff, André. 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182. https://www.jmlr.org/papers/v3/guyon03a.html
22. Harris, Charles R.; Millman, K. Jarrod; van der Walt, Stéfan J.; Gommers, Ralf; Virtanen, Pauli; Cournapeau, David; Wieser, Eric; Taylor, Julian; Berg, Sebastian; Smith, Nathaniel J.; Kern, Robert; Picus, Matti; Hoyer, Stephan; van Kerkwijk, Marten H.; Brett, Matthew; Haldane, Allan; del Río, Jaime Fernández; Wiebe, Mark; Peterson, Pearu; Gérard-Marchant, Pierre; Sheppard, Kevin; Reddy, Tyler; Weckesser, Warren; Abbasi, Hameer; Gohlke, Christoph; Oliphant, Travis E. 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2
23. Hastie, Trevor J.; Tibshirani, Robert John; Tibshirani, Ryan J. 2017. Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso. arXiv preprint. https://doi.org/10.48550/arxiv.1707.08692
24. Holm, Sture. 1979. A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics 6, 65–70. https://doi.org/10.2307/4615733
25. Huangfu, Q.; Hall, J. A. J. 2017. Parallelizing the dual revised simplex method. Mathematical Programming Computation 10, 119–142. https://doi.org/10.1007/s12532-017-0130-5
26. Hunter, John D. 2007. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 9, 90–95. https://doi.org/10.1109/mcse.2007.55
27. Joshi, S.; Boyd, S. 2009. Sensor Selection via Convex Optimization. IEEE Transactions on Signal Processing 57, 451–462. https://doi.org/10.1109/tsp.2008.2007095
28. Karpovsky, M.G.; Chakrabarty, K.; Levitin, L.B. 1998. On a new class of codes for identifying vertices in graphs. IEEE Transactions on Information Theory 44, 599–611. https://doi.org/10.1109/18.661507
29. Kira, Kenji; Rendell, Larry A. 1992. The feature selection problem: traditional methods and a new algorithm. In National Conference on Artificial Intelligence, pp. 129–134.
30. Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. 1983. Optimization by Simulated Annealing. Science 220, 671–680. https://doi.org/10.1126/science.220.4598.671
31. Krause, Andreas; Singh, Ajit; Guestrin, Carlos. 2008. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. Journal of Machine Learning Research 9, 235–284. https://www.jmlr.org/papers/v9/krause08a.html
32. Künsch, Hans Rudolf. 1989. The Jackknife and the Bootstrap for General Stationary Observations. The Annals of Statistics 17, 1217–1241. https://doi.org/10.1214/aos/1176347265
33. Li, Jundong; Cheng, Kewei; Wang, Suhang; Morstatter, Fred; Treviño, Robert P.; Tang, Jiliang; Liu, Huan. 2017. Feature Selection. ACM Computing Surveys 50, 1–45. https://doi.org/10.1145/3136625
34. McCann, Michael; Johnston, Adrian. 2008. SECOM. UCI Machine Learning Repository. https://doi.org/10.24432/C54305
35. McKinney, Wes. 2010. Data Structures for Statistical Computing in Python. In Proceedings of the Python in Science Conference, pp. 56–61. https://doi.org/10.25080/majora-92bf1922-00a
36. Moon, Jun; Başar, Tamer. 2017. Static Optimal Sensor Selection via Linear Integer Programming: The Orthogonal Case. IEEE Signal Processing Letters 24, 953–957. https://doi.org/10.1109/lsp.2017.2698465

37. Narendra, P. M.; Fukunaga, K. 1977. A Branch and Bound Algorithm for Feature Subset Selection. IEEE Transactions on Computers C-26, 917–922. https://doi.org/10.1109/TC.1977.1674939
38. Paninski, Liam. 2003. Estimation of Entropy and Mutual Information. Neural Computation 15, 1191–1253. https://doi.org/10.1162/089976603321780272
39. Peng, Hanchuan; Long, Fuhui; Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–1238. https://doi.org/10.1109/tpami.2005.159
40. Phipson, Belinda; Smyth, Gordon K. 2010. Permutation P-values Should Never Be Zero: Calculating Exact P-values When Permutations Are Randomly Drawn. Statistical Applications in Genetics and Molecular Biology 9, Art. 39. https://doi.org/10.2202/1544-6115.1585
41. Rosich, Albert; Sarrate, Ramón; Nejjari, Fatiha. 2008. Optimal Sensor Placement for FDI using Binary Integer Linear Programming. Open Repository and Bibliography (University of Luxembourg).
42. Saeys, Yvan; Inza, Iñaki; Larrañaga, Pedro. 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517. https://doi.org/10.1093/bioinformatics/btm344
43. Salem, Milad; Taheri, Shayan; Yuan, Jiann-Shiun. 2018. An Experimental Evaluation of Fault Diagnosis from Imbalanced and Incomplete Data for Smart Semiconductor Manufacturing. Big Data and Cognitive Computing 2, 30. https://doi.org/10.3390/bdcc2040030
44. Sarrate, Ramón; Puig, Vicenç; Escobet, Teresa; Rosich, Albert. 2007. Optimal sensor placement for model-based fault detection and isolation. In 2007 46th IEEE Conference on Decision and Control, pp. 2584–2589. https://doi.org/10.1109/cdc.2007.4434452
45. Susto, Gian Antonio; Maggipinto, Marco; Zocco, Federico; McLoone, Seán. 2018. A Dynamic Sampling Approach for Cost Reduction in Semiconductor Manufacturing. Procedia Manufacturing 17, 1031–1038. https://doi.org/10.1016/j.promfg.2018.10.083
46. Tamura, Ryuta; Takano, Yuichi; Miyashiro, Ryuhei. 2022. Feature subset selection for kernel SVM classification via mixed-integer optimization. arXiv preprint. https://doi.org/10.48550/arxiv.2205.14325
47. Tibshirani, Robert John. 1996. Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society Series B (Statistical Methodology) 58, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x
48. Virtanen, Pauli; Gommers, Ralf; Oliphant, Travis E.; Haberland, Matt; Reddy, Tyler; Cournapeau, David; Burovski, Evgeni; Peterson, Pearu; Weckesser, Warren; Bright, Jonathan; van der Walt, Stéfan J.; Brett, Matthew; Wilson, Joshua; Millman, K. Jarrod; Mayorov, Nikolay; Nelson, Andrew R. J.; Jones, Eric; Kern, Robert; Larson, Eric; Carey, C J; Polat, İlhan; Feng, Yu; Moore, Eric W.; VanderPlas, Jake; Laxalde, Denis; Perktold, Josef; Cimrman, Robert; Henriksen, Ian; Quintero, E. A.; Harris, Charles R.; Archibald, Anne M.; Ribeiro, Antônio H.; Pedregosa, Fabian; van Mulbregt, Paul; SciPy 1.0 Contributors. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2
49. Wang, Ruizhi; Miao, Duoqian; Hu, Guirong. 2006. Discernibility Matrix Based Algorithm for Reduction of Attributes. In 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 477–480. https://doi.org/10.1109/wi-iatw.2006.58
50. Wang, Joseph; Trapeznikov, Kirill; Saligrama, Venkatesh. 2015. Sensor Selection by Linear Programming. arXiv preprint. https://doi.org/10.48550/arxiv.1509.02954
51. Wang, Jian; Feng, Jian; Han, Zhiyan. 2016. Discriminative Feature Selection Based on Imbalance SVDD for Fault Detection of Semiconductor Manufacturing Processes. Journal of Circuits, Systems and Computers 25, 1650143. https://doi.org/10.1142/s0218126616501437
52. Wieczorek, Wojciech; Kozak, Jan; Strąk, Łukasz; Nowakowski, Arkadiusz. 2021. Minimum Query Set for Decision Tree Construction. Entropy 23, 1682. https://doi.org/10.3390/e23121682