

EPR-C3: A Deterministic Constraint-Aware Heuristic for High-Dimensional Subset Selection in Multiple Linear Regression

Jackson J. Alcázar

`jackson.alcazar@udd.cl`

Universidad del Desarrollo

Research Article

Keywords: Feature subset selection, Data science and analytics, Interpretable regression, High-dimensional regression, Constraint-aware model selection, Heuristic optimization

Posted Date: June 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10121603/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at International Journal of Data Science and Analytics on September 24th, 2026. See the published version at <https://doi.org/10.1007/s41060-026-01298-0>.

EPR-C3: A Deterministic Constraint-Aware Heuristic for High-Dimensional Subset Selection in Multiple Linear Regression

Jackson J. Alcázar^{1*}

^{1*}Centro de Química Médica, Facultad de Medicina Clínica Alemana, Universidad del Desarrollo, Santiago, 7780272, Chile.

Corresponding author(s). E-mail(s): jackson.alcazar@udd.cl;

Abstract

Subset selection in multiple linear regression (MLR) is a statistical-computing challenge in high-dimensional data because candidate predictors expand the combinatorial search space and increase risks of multicollinearity, overfitting, and statistically inadmissible models. This study introduces EPR-C3, a deterministic multi-start neighborhood-search heuristic for constraint-aware subset selection in ordinary least squares (OLS)-based MLR. EPR-C3 combines structured exploration with reproducible local refinement through expansion, perturbation, reduction, and C3 constraint refinement, comprising correlation cleanup, replacement recovery, and variance inflation factor pruning, while preserving the native OLS formulation and explicit candidate equations. Benchmark analyses show that EPR-C3 recovers a high proportion of Top-K admissible subsets identified by exhaustive all-subsets regression while reducing computational cost in highly combinatorial problems. An empirical utility threshold is proposed to identify when heuristic search becomes preferable to exhaustive enumeration. Comparisons with stepwise procedures, penalized-regression preselection, filter-wrapper screening, genetic algorithms, and branch-and-bound subset selection show that embedding admissibility constraints within the search workflow improves the recovery–efficiency tradeoff. An applied case study demonstrates reproducible recovery of a published MLR equation with reduced runtime, supporting EPR-C3 as a deterministic statistical-computing heuristic for tractable subset selection in regression modeling and prediction.

Keywords: Feature subset selection, Data science and analytics, Interpretable regression, High-dimensional regression, Constraint-aware model selection, Heuristic optimization

1 Introduction

Multiple linear regression (MLR) remains one of the most widely used modeling frameworks in applied statistics, data science, scientific research, and analytics workflows because of its interpretability, computational simplicity, and transparent inferential structure [1–5]. In many applied settings, MLR is preferred because regression coefficients can be interpreted directly, assumptions can be inspected using established diagnostics, and final models can be communicated to nonspecialist audiences with relative clarity [6–9]. These properties remain important in data-rich domains where model transparency is not merely desirable but necessary for interpretation, auditability, decision support, and scientific reporting.

Despite these advantages, feature or predictor subset selection in MLR becomes particularly challenging in high-dimensional data regimes, where the number of candidate variables produces a combinatorial explosion of possible subsets while simultaneously increasing the risk of multicollinearity, overfitting, and statistically inadmissible models [10–13]. Data-science applications often contain many candidate features or predictors, including weakly informative, redundant, or highly correlated variables [14, 15]. Including too many predictors can increase variance, destabilize coefficient estimates, inflate multicollinearity, and reduce interpretability, while overly aggressive selection can discard predictors with weak marginal effects but meaningful joint contributions [13, 16–18]. Consequently, subset selection in high-dimensional MLR is both a computational and statistical challenge, requiring methods that balance predictive performance, interpretability, reproducibility, and admissibility constraints.

Several families of methods have been proposed to address this problem [19]. Screening methods can efficiently reduce dimensionality, often using marginal or low-order association criteria [20]. Penalized regression methods incorporate regularization into the estimation procedure and can provide scalable shrinkage or sparsity; examples include ridge regression, least absolute shrinkage and selection operator (Lasso), and Elastic Net [21–26]. Testing-based procedures, such as stepwise regression, preserve the classical ordinary least squares (OLS) framework, but are typically greedy and may be sensitive to early inclusion or exclusion decisions [27–29].

Exact subset selection methods represent a classical approach to variable selection in MLR. The most direct strategy, known as all-subsets regression, evaluates all possible predictor combinations, guaranteeing identification of the globally optimal model under a chosen criterion. However, the number of candidate models grows exponentially with the number of predictors, making exhaustive enumeration computationally prohibitive in high-dimensional data settings. To alleviate this limitation, branch-and-bound procedures, such as the leaps-and-bounds (Leaps) algorithm [30], exploit monotonicity properties of the residual sum of squares to prune large portions of the search space without explicit evaluation. These methods retain optimality guarantees for criteria based on monotonic functions of the residual sum of squares. Nevertheless, their efficiency deteriorates in high-dimensional predictor spaces and becomes more limited when additional constraints, such as multicollinearity control or coefficient significance, are imposed, as these constraints are not generally monotonic and cannot be directly incorporated into safe pruning rules.

Beyond exact and pruning-based strategies, broader search heuristics have also been used to explore large subset spaces when exhaustive enumeration becomes impractical in high-dimensional settings. Genetic algorithms and related metaheuristics explore complex model spaces through stochastic operations, but their behavior depends on algorithmic hyperparameters and may vary across runs [31–35].

These approaches involve different tradeoffs. Screening is efficient but can miss jointly relevant predictors. Penalized regression is scalable and often effective for prediction, but because it modifies the estimation objective, direct comparisons with classical OLS subset models are less straightforward unless an explicit OLS refit is performed after selection. Stepwise procedures are simple and interpretable but limited by local greedy behavior. Exact methods provide optimality guarantees but scale poorly in high-dimensional data regimes and are difficult to adapt to constraint-rich settings. Stochastic metaheuristics can broaden exploration but introduce run-to-run dependence and additional tuning burden. In applied data-science and analytics workflows, there is a practical need for a method that remains within the OLS-based MLR framework, evaluates explicit predictor subsets, and enforces statistical and multicollinearity constraints while remaining operationally feasible in high-dimensional settings. Such a method should also explore beyond a single greedy path without requiring exhaustive or exact branch-and-bound enumeration.

This work introduces EPR-C3, a deterministic multi-start, constraint-aware neighborhood-search heuristic for subset selection in OLS-based MLR. EPR-C3 is formulated as a statistical-computing algorithm for data-analytics workflows involving large combinatorial feature-subset spaces in which statistical admissibility constraints are embedded directly into the search workflow. More specifically, EPR-C3 combines multi-start exploration across trajectories with reproducible local neighborhood moves and constraint-aware filtering. Each trajectory begins from a small subset and follows deterministic local decisions through expansion, perturbation, reduction, and C3 constraint refinement comprising correlation cleanup, replacement recovery, and VIF pruning. Candidate models are evaluated using standard OLS estimation and are accepted only when they satisfy user-defined constraints, including maximum model size, target metric thresholds, pairwise correlation limits, variance inflation factor (VIF) restrictions, and coefficient-significance criteria. EPR-C3 is designed for high-dimensional data regimes in which all-subsets regression is computationally prohibitive, but the analyst still requires transparent, interpretable, reproducible, and statistically constrained MLR models. This deterministic and constraint-aware framework is particularly relevant to data-science and analytics settings that require a balance between algorithmic tractability, computational cost, interpretability, reproducibility, and statistical admissibility.

The contributions of this work are sixfold. First, EPR-C3 is introduced as a deterministic multi-start, constraint-aware neighborhood-search heuristic for high-dimensional OLS-based subset selection. Second, its search operators and admissibility rules are defined explicitly within an OLS subset selection framework. Third, benchmarking against exhaustive search is used to evaluate subset recovery performance and computational scaling under high-dimensional predictor regimes. Fourth, comparison with established subset-search paradigms shows that EPR-C3 provides higher

recovery of reference subsets with a favorable recovery–efficiency tradeoff. Fifth, an empirical utility threshold identifies when EPR-C3 becomes preferable to exhaustive enumeration in highly combinatorial problems. Sixth, an applied data-analytics case study demonstrates reproducible model recovery and runtime reduction. Together, these contributions position EPR-C3 as a deterministic approach for interpretable, constraint-aware model selection in large combinatorial search spaces.

2 Problem formulation

The present formulation is restricted to OLS regression with a continuous response variable. Let $\mathcal{X} = \{x_1, x_2, \dots, x_d\}$ denote the set of candidate features or predictors, where d is the total number of available predictors. Let y denote that response variable. Let $S \subseteq \mathcal{X}$ denote an active predictor subset of size $k = |S|$. For a given subset S , an OLS model is fitted as

$$y = \beta_0 + \sum_{x_j \in S} \beta_j x_j + \varepsilon. \quad (1)$$

The goal is to identify admissible subsets subject to model size and statistical constraints. Let $f(S)$ denote a selected performance metric evaluated for subset S , such as $R^2(S)$, adjusted $R^2(S)$, cross-validated $Q^2(S)$, or $\text{RMSE}(S)$. The subset search problem is then formulated as maximizing or minimizing $f(S)$. For notational simplicity, the formulation is presented in maximization form; metrics to be minimized can be transformed accordingly, or handled through an equivalent criterion such as $f(S) \leq \tau_f$ for error metrics.

A subset is considered reportable and admissible only if it satisfies the constraints

$$k \leq w, \quad (2)$$

$$f(S) \geq \tau_f, \quad (3)$$

$$C(S) = \max_{x_i, x_j \in S, i \neq j} |r_{ij}| \leq \tau_r, \quad (4)$$

$$V(S) = \max_{x_j \in S} \text{VIF}(x_j) \leq \tau_{\text{VIF}}, \quad (5)$$

$$P(S) = \max_j (p_j) \leq \alpha. \quad (6)$$

Here, w is the maximum number of predictors allowed in the model, $r_{ij} = \text{corr}(x_i, x_j)$ is the pairwise correlation coefficient between predictors x_i and x_j , τ_f is the target-metric threshold, τ_r is the maximum allowed pairwise correlation among active predictors, τ_{VIF} is the VIF threshold, p_j is the p-value for coefficient j under the implemented coefficient-significance rule, and α is the significance level for coefficient tests. These constraints define the final reporting admissibility of stored models. During the search trajectory, however, final admissibility and movement feasibility are not identical. Correlation and coefficient-significance criteria are used as candidate-level feasibility screens in local neighborhood evaluations. The target metric guides local movement among screened candidates, while the target-metric threshold and VIF threshold determine whether evaluated subsets are stored and retained

as reportable models. Thus, EPR-C3 is both a neighborhood-search heuristic and an admissibility-aware model-selection strategy.

The number of non-null subsets evaluated by all-subsets regression up to size w is

$$N(d, w) = \sum_{k=1}^w \binom{d}{k}. \quad (7)$$

This quantity excludes the null model and grows rapidly as d and w increase. As a result, exhaustive enumeration can become impractical even when the final model is required to remain small. EPR-C3 addresses this issue by evaluating a structured fraction of the subset space while retaining direct OLS fitting and explicit model admissibility checks.

3 The EPR-C3 subset-search algorithm

3.1 Overview

EPR-C3 is a multi-start neighborhood-search heuristic for feature or predictor subset selection in MLR. Each trajectory begins with a small predictor subset used to initialize the active set S , on which a repeated sequence of local modifications is applied. Conditional on the supplied seed list, each search trajectory is deterministic. Seed generation may be randomized, but once the seed list and random state are fixed, all subsequent neighborhood decisions are deterministic and the full run is reproducible. The same ranked model list is expected under the same software and computational environment.

The acronym EPR-C3 summarizes the named components that constitute the method: E = Expand, P = Perturb, R = Reduce, and C3 = constraint-aware refinement comprising correlation cleanup, replacement recovery, and VIF pruning.

1. *Expand*: adds the inactive predictor that produces the largest improvement in the target metric among screened candidates, provided that the model-size limit is not exceeded.
2. *Perturb*: performs one-for-one exchanges between active and inactive predictors to escape locally optimal subsets.
3. *Reduce*: tests one-variable removals and accepts the removal when it maintains or improves the target score, thereby favoring parsimony when simplification is beneficial.
4. *C3 refinement*:
 - (a) *Correlation cleanup*: identify pairs of active predictors whose pairwise correlation exceeds the user-defined threshold and remove the predictor whose exclusion gives the better screened target score.
 - (b) *Replacement recovery*: revisit predictors removed during correlation cleanup and test screened replacement moves to recover or improve the target score.

- (c) *VIF pruning*: if the active subset exceeds the VIF threshold, iteratively remove high-VIF predictors whose exclusion yields the best screened score, and temporarily veto them within that trajectory; the veto is local to the current trajectory and is not applied globally across seeds.

EPR-C3 does not modify the OLS objective through shrinkage, as penalized regression does. Nor does it attempt to enumerate all predictor combinations, as in all-subsets regression. Its design differs from stochastic metaheuristics because it relies on seed-controlled trajectories followed by deterministic local decisions and ranking.

3.2 Admissibility function

During each trajectory, active predictor subsets that satisfy the final reporting criteria are stored in a model space denoted by $\mathcal{M}$. For any evaluated subset S to be stored as reportable in $\mathcal{M}$, it must satisfy the constraints introduced in Section 2. Let $A(S)$ denote the reporting admissibility function, with $A(S) = 1$ if all required constraints are satisfied and $A(S) = 0$ otherwise. A compact representation of this condition is

$$A(S) = \begin{cases} 1, & \text{if } k \leq w, f(S) \geq \tau_f, C(S) \leq \tau_r, \\ & V(S) \leq \tau_{\text{VIF}}, P(S) \leq \alpha, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

This function defines final reporting admissibility rather than the complete movement rule used at every neighborhood operation. In Expand, Perturb, and Reduce, newly proposed candidate subsets are evaluated by OLS and compared through the target metric $f(S)$ after correlation and coefficient-significance screening. Therefore, proposed moves are first screened using $C(S) \leq \tau_r$ and $P(S) \leq \alpha$, whereas $f(S)$ drives the local trajectory among the surviving candidates.

The C3 block can still operate on the active subset when correlation conflicts are inherited from the seed, retained from earlier trajectory states, or detected during refinement. The thresholds $f(S) \geq \tau_f$ and $V(S) \leq \tau_{\text{VIF}}$ determine whether evaluated subsets are stored as admissible reportable models, and VIF is additionally handled through the VIF-pruning and final filtering stages.

3.3 Algorithmic procedure

Algorithm 1 summarizes the EPR-C3 procedure. The algorithm returns a ranked set of admissible and non-duplicated MLR models. Algorithmic summaries of the additional benchmark methods are provided in the Supplementary Material.

Algorithm 1. EPR-C3 for constraint-aware MLR subset selection

Require: Candidate predictors $\mathcal{X}$, response y , maximum model size w , seed list $\mathcal{D}$, target metric f , thresholds τ_f , τ_r , τ_{VIF} , α .

Ensure: Ranked set of admissible predictor subsets $\mathcal{M}$.

- 1: $\mathcal{M} \leftarrow \emptyset$
- 2: **for** each seed-defined initial subset $S_0 \in \mathcal{D}$ **do**
- 3: $S \leftarrow S_0$
- 4: $D_{\text{drop}} \leftarrow \emptyset$

```

5:    $D_{\text{VIF}} \leftarrow \emptyset$ 
6:   Evaluate  $S$  by OLS and store it in  $\mathcal{M}$  if  $A(S) = 1$ 
7:   while the trajectory has not stopped do
8:      $R \leftarrow \mathcal{X} \setminus (S \cup D_{\text{drop}} \cup D_{\text{VIF}})$ 
9:      $\mathcal{L}_{\text{swap}} \leftarrow \emptyset$ 
10:    Newly proposed neighborhood candidates are screened
11:    by  $C(S) \leq \tau_r$  and  $P(S) \leq \alpha$  before movement-level
12:    acceptance
13:    Expand: test  $S \cup \{x\}$  for  $x \in R$  when  $|S| < w$ 
14:    Accept the screened expansion with the largest
15:    improvement in  $f(S)$ , if any
16:    Perturb: test  $(S \setminus \{x_a\}) \cup \{x_i\}$  with  $x_a \in S, x_i \in R$ 
17:    Accept the screened exchange with the largest improve-
18:    ment in  $f(S)$ , if any
19:    Reduce: test  $S \setminus \{x_a\}$  for  $x_a \in S$ 
20:    Accept the screened removal when it maintains or
21:    improves  $f(S)$ , if any
22:    C3 constraint-aware refinement: apply correlation
23:    cleanup, replacement recovery, and VIF pruning
24:    Correlation cleanup: detect pairs  $(x_j, x_k) \in S$  with
25:     $|r_{jk}| > \tau_r$ 
26:    Remove the predictor whose exclusion gives the better
27:    screened target score
28:    Add removed predictors to  $D_{\text{drop}}$  and record conflicts in
29:     $\mathcal{L}_{\text{swap}}$ 
30:    Replacement recovery: test screened replacement moves recorded
31:    in  $\mathcal{L}_{\text{swap}}$ 
32:    Accept the screened replacement that recovers or improves
33:     $f(S)$ , if any
34:    VIF pruning: while  $\max \text{VIF}(S) > \tau_{\text{VIF}}$ 
35:      Evaluate one-variable removals among high-VIF
36:      predictors
37:      Remove the predictor whose exclusion gives the best
38:      screened score
39:      Add the removed predictor to  $D_{\text{drop}}$  and  $D_{\text{VIF}}$ 
40:      Store every non-duplicated evaluated candidate satisfy-
41:      ing  $A(S) = 1$  in  $\mathcal{M}$ 
42:      if the best active-subset score and the best stored score remain unchanged across
43:      two consecutive full EPR-C3 cycles then
44:        stop the trajectory
45:      end if
46:    end while
47:  end for
48: Rank  $\mathcal{M}$  according to  $f$ .

```

4 Benchmark design

The benchmark study consisted of three stages. First, all-subsets regression was implemented via brute-force enumeration to provide an exact reference for calibrating

EPR-C3 and quantifying its recovery of top-ranked models under increasing combinatorial complexity. Second, EPR-C3 was compared with representative subset selection baselines spanning complementary methodological families. Third, an applied regression-modeling case study was conducted to assess reproducible model recovery and runtime performance on real-world datasets. Across all three stages, R^2 was used as the target metric for model evaluation and ranking, while the maximum subset size w was fixed to enable comparison of recovery behavior under a common goodness-of-fit criterion without changing the OLS objective.

Benchmark experiments were performed on a 64-bit x86_64 system running Ubuntu 20.04.6 LTS. The hardware environment provided 44 CPU cores distributed across two sockets, with 22 cores per socket and one thread per core, a nominal CPU frequency of 2.29 GHz, and 91 GiB of system memory.

4.1 Calibration against exhaustive enumeration

EPR-C3 was first benchmarked against an all-subsets regression procedure to address two questions: (i) what seed budget provides a favorable tradeoff between recovery of top-ranked models and computational cost and (ii) when EPR-C3 becomes computationally advantageous.

Recovery was defined as the percentage of the top- K reference subsets that were exactly recovered by EPR-C3, where the reference subsets were obtained from all-subsets regression.

CPU time was compared between EPR-C3 and exhaustive enumeration under identical admissibility rules to determine the number of subsets at which EPR-C3 starts to outperform the all-subsets regression.

Because EPR-C3 is intended for combinatorially demanding settings, the benchmark explicitly included regimes in which the number of possible subsets approached hundreds of millions. The benchmark covers four increasingly demanding regimes: $d = 60, w = 4$; $d = 60, w = 6$; $d = 45, w = 8$; and $d = 35, w = 10$. These settings correspond to 523,685; 56,049,057; 270,463,854; and 286,454,523 possible subsets, respectively. The default seed size was defined as half of the maximum permitted subset size w . This pragmatic initialization places the starting subset near the midpoint of the feasible model-size range, thereby allowing both expansion and reduction moves from the earliest search iterations. In this way, trajectories are not systematically biased toward either overly sparse or near-saturated initial subsets.

4.2 Comparative benchmarking against alternative methods

After calibration against exhaustive enumeration, EPR-C3 was compared with representative baseline methods selected to cover complementary families of subset-selection procedures. For interpretive consistency, Forward and Backward Stepwise were treated as stepwise procedures; Ridge, Elastic Net, and Lasso as penalized-regression screening methods followed by exhaustive OLS subset refinement; Correlation Screening and Redundancy Filtering before All-Subsets Search (CSR-AS) as a filter-wrapper screening method; Leaps as a branch-and-bound best-subset search implementation [30, 36]; the genetic algorithm baseline as a stochastic metaheuristic [35]; and EPR-C3 as the

proposed deterministic neighborhood-search heuristic. Each candidate subset generated by these methods was refitted by OLS and subjected to the same admissibility criteria.

The penalized-regression baselines were implemented in two stages. First, each penalized model was fitted on the standardized predictor matrix and used only to rank predictors according to the absolute magnitude of the fitted coefficients. Lasso, Elastic Net, and Ridge regression used cross-validated regularization-parameter selection, using the `LassoCV`, `ElasticNetCV`, and `RidgeCV` estimators in scikit-learn, respectively. For Elastic Net, ℓ_1 -mixing ratios of 0.2, 0.5, and 0.8 were evaluated using 5-fold cross-validation. In the second stage, the top 20 predictors from each penalized-regression ranking were retained as a reduced candidate pool. All OLS subsets of size 1 to w were then enumerated from this pool, where w is the maximum allowed model size.

CSR-AS was also implemented as a two-stage filter-wrapper procedure. Predictors were first ranked by the absolute Pearson correlation with the response variable y . The algorithm then built a reduced pool of 20 predictors by iteratively selecting the highest-ranked predictors while excluding candidates that exceeded the redundancy threshold with any predictor already selected. All OLS subsets of size 1 to w were subsequently enumerated from this nonredundant pool.

The genetic algorithm baseline used a binary chromosome representation in which each gene indicated whether a predictor was included in the subset. The initial population contained 1000 randomly generated subsets with sizes between 1 and w , matching the seed count used by EPR-C3. Evolution was performed for 20,000 generations using elitist selection, single-point crossover with probability 0.8, and bit-flip mutation with probability 0.08; this generation budget was selected to provide a computationally demanding comparator of similar practical intent rather than a minimal genetic-algorithm run. Whenever mutation or crossover produced subsets larger than w , excess active predictors were randomly removed to enforce the model-size constraint.

Leaps was included as a classical branch-and-bound implementation for best-subset regression, using the R package `leaps` [36]. The method was run through `regsubsets` using `method = "exhaustive"`, `really.big = TRUE`, `nvmax = w`, and `nbest = 35000`. The `nbest` value was set as a deliberately broad retained-candidate budget intended to make Leaps comparable to EPR-C3 in practical computational cost while increasing the likelihood of retaining admissible subsets after post-processing. The resulting candidate subsets were exported and then evaluated in the same post-processing pipeline used for the other methods.

Forward Stepwise started from the empty model and iteratively added the predictor that produced the best admissible OLS subset at each step, stopping when no admissible candidate remained or when the maximum size w was reached. Backward Stepwise started from the best admissible w -predictor subset identified within an initial candidate pool containing, at most, twice the maximum number of predictors allowed per model, selected from the first available predictor columns. It then iteratively removed one predictor at a time when the resulting reduced subset provided the best admissible model.

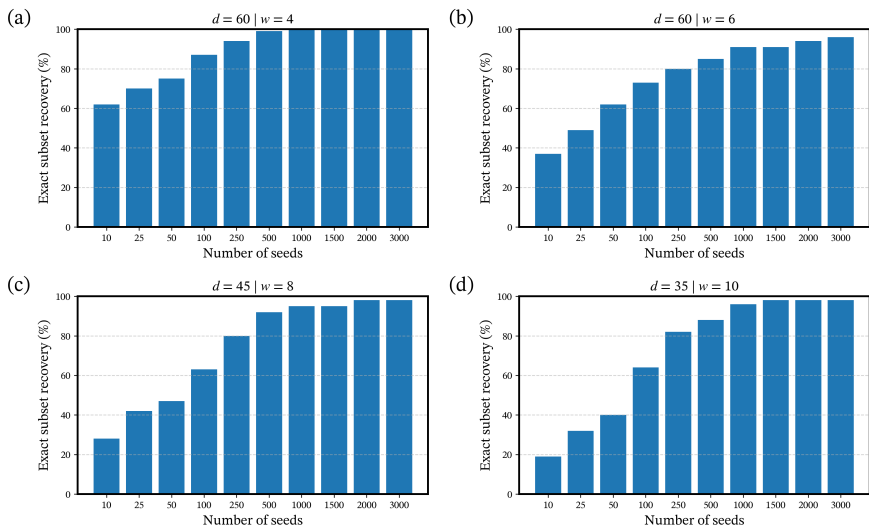


Fig. 1 Exact recovery of the top 100 reference models from all-subsets regression by EPR-C3 as a function of the number of seeds. Panels: (a) $d = 60, w = 4$; (b) $d = 60, w = 6$; (c) $d = 45, w = 8$; and (d) $d = 35, w = 10$, where d is the total number of candidate predictors and w is the maximum number of predictors per model.

This design ensured that the comparison focused on the search strategy rather than on differences in model fitting or acceptance criteria. Across all methods, candidate subsets were ultimately refitted as OLS models, evaluated using the same admissibility filters, screened to remove duplicate predictor combinations, and ranked using the same performance criterion. Algorithmic summaries of all comparative baselines are provided in the Supplementary Material to make the benchmark protocol explicit.

4.3 Case study

A published pK_a modeling dataset with 53 candidate predictors was used as an applied test case [37]. The purpose of this case study is not to introduce a new chemical model, but to verify whether EPR-C3 can recover a previously reported linear model in a real-world modeling scenario involving approximately 2.6×10^7 possible subsets and to compare its runtime with the all-subsets regression procedure used in the original study. Successful recovery was defined as exact recovery of the published predictor set and equation after OLS refitting under the same admissibility rules.

5 Results and discussion

5.1 Seed calibration and utility threshold

5.1.1 Seed budget calibration

Figure 1 summarizes the recovery behavior of EPR-C3 relative to all-subsets regression for the top 100 ranked models.

The results indicate that recovery improves with the number of seeds. In the moderate combinatorial regime with $d = 60$ and $w = 4$ (Figure 1a), recovery exceeded 80% with approximately 250 seeds, while complete agreement with the all-subsets top 100 models is observed at 1000 seeds. When the maximum model size is increased to $w = 6$ (Figure 1b), the number of candidate subsets increases by more than two orders of magnitude, yet strong recovery is still obtained from 1000 seeds onward. Similar behavior is observed for $d = 45$ and $w = 8$ (Figure 1c), and $d = 35$ and $w = 10$ (Figure 1d), where the total number of possible subsets reaches approximately 2.7×10^8 and 2.9×10^8 , respectively.

Across the evaluated scenarios, 1000 seeds provided a practical baseline associated with recovery levels above 95% under the evaluated configurations. This does not imply that 1000 seeds is universally optimal, but it provides a defensible default when the number of possible subsets makes exhaustive benchmarking no longer feasible. While additional seeds may be required for substantially larger subset spaces, direct recovery assessment becomes impractical because all-subsets regression is itself computationally inaccessible.

5.1.2 Computational utility and utility threshold

Figure 2 compares the CPU-time scaling of EPR-C3 with 1000 seeds against all-subsets regression. In low-combinatorial regimes, exhaustive search can remain faster because the computational overhead associated with launching and evaluating 1000 seed trajectories may exceed the cost of directly enumerating the complete subset space. Although reducing the number of seed trajectories decreases this overhead, the 100- and 250-seed comparisons reported in the Supplementary Material (Figure S1) indicate that smaller seed budgets offer limited practical computational advantages in these low-dimensional regimes.

The point at which EPR-C3 becomes computationally preferable depends primarily on the size of the combinatorial subset space. As both the predictor dimensionality and the maximum subset size increase, the number of candidate subsets grows rapidly. Under these conditions, the fixed overhead of the seed-based trajectories becomes progressively less important relative to the cost of exhaustive enumeration, and EPR-C3 enters its favorable operating regime.

For $w = 4$, EPR-C3 with 1000 seeds becomes computationally advantageous only when the search space exceeds approximately 2×10^6 candidate subsets. As shown in Figure 2a, below this empirical threshold, the total CPU time of EPR-C3 is greater than that of all-subsets regression because the cost of the seed trajectories is not yet compensated by the reduction in evaluated subsets. Around $d = 90$, corresponding to 2,676,765 possible subsets, the EPR-C3 and all-subsets curves intersect. This crossover marks the beginning of the regime in which EPR-C3 becomes computationally advantageous for the $w = 4$ and 1000-seed configuration.

The crossover shifts as the maximum subset size increases. For $w = 6$, EPR-C3 becomes faster than exhaustive search at approximately $d = 50$ (Figure 2b), corresponding to 18,260,635 candidate subsets according to Equation 7. For $w = 8$, the transition occurs around $d = 35$ (Figure 2c), corresponding to 32,267,667 subsets. For $w = 10$, EPR-C3 becomes clearly advantageous at approximately $d = 30$, where

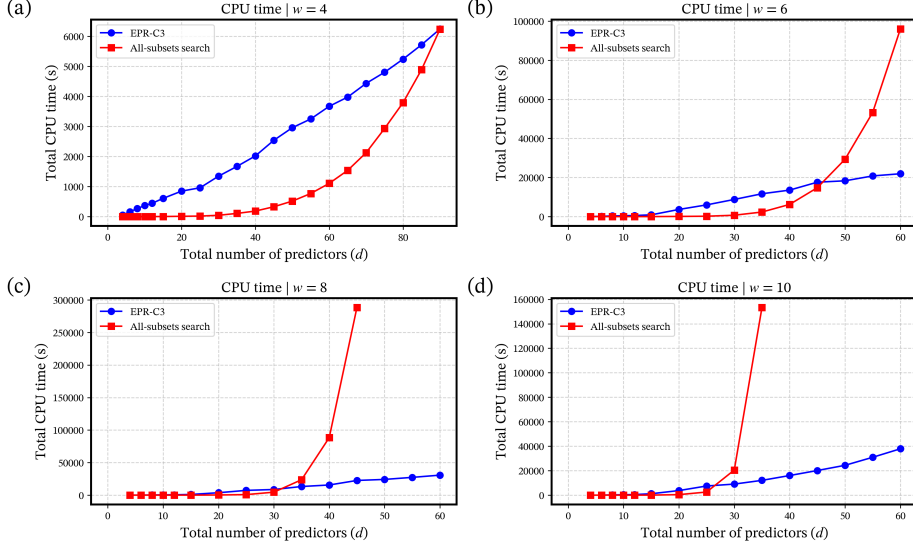


Fig. 2 CPU-time scaling of EPR-C3 and all-subsets regression. Panels (a–d) show total CPU time as a function of the number of candidate predictors (d) for $w = 4, 6, 8$, and 10 , respectively. EPR-C3 was evaluated using 1000 seeds, whereas all-subsets regression enumerated all predictor combinations up to each maximum model size.

the subset space contains 53,009,101 candidate subsets (Figure 2d). At $d = 35$ and $w = 10$, EPR-C3 is approximately 12.7 times faster than exhaustive all-subsets search while maintaining strong recovery of the highest-ranked solutions. This behavior confirms that the computational benefit of EPR-C3 increases as the combinatorial burden grows.

Based on these empirical crossover points, the utility threshold U above which EPR-C3 becomes computationally advantageous under the benchmarked implementation and hardware was empirically approximated by the linear relation

$$U = (8.2w - 31) \times 10^6, \quad w \geq 4. \quad (9)$$

Here, w is the maximum number of predictors per model. If $N(d, w)$ denotes the total number of candidate subsets defined in Equation 7, EPR-C3 is expected to operate in its favorable computational regime when $N(d, w) > U$ under the benchmarked implementation, seed budget, and hardware configuration.

For $w < 4$, the threshold should be fixed at the $w = 4$ reference value rather than extrapolated from Equation 9. This recommendation avoids overestimating the usefulness of EPR-C3 in small subset spaces, where exhaustive enumeration is usually inexpensive and fully exact. Therefore, Equation 9 should not be interpreted as a theoretical complexity bound, but as an empirical operational guideline derived from the benchmark conditions. Its purpose is practical: to prevent unnecessary use of the

heuristic in low-combinatorial regimes and to recommend EPR-C3 only when the subset space is large enough for its structured search strategy to provide a computational advantage.

Within this framework, the utility threshold can also be interpreted as an operational boundary between low- and high-combinatorial MLR regimes. This distinction is more informative than predictor dimensionality alone because the computational burden depends jointly on the total number of candidate predictors d and the maximum target subset size w . Once $N(d, w)$ exceeds U , the number of candidate subsets increases sharply as d grows, with the rate of expansion being directly controlled by w . For the 1000-seed configuration evaluated here, EPR-C3 shows favorable computational performance above this boundary, as illustrated in Figure 2, while maintaining strong recovery of high-ranking exhaustive solutions in the evaluated Top- K analyses.

However, Equation 9 is specific to the benchmarked 1000-seed configuration and should not be used as a seed-independent guarantee. If a substantially larger number of seeds is used when $N(d, w)$ is only slightly greater than U , the computational advantage of EPR-C3 over all-subsets regression is not necessarily guaranteed, since additional seed trajectories increase the fixed search overhead. In contrast, when $N(d, w) \gg U$, all-subsets regression rapidly becomes impractical, and EPR-C3 becomes increasingly justified as a necessary and computationally more efficient approximation, even when larger seed budgets are used to improve recovery robustness. The linear-fit guidance supporting Equation 9 is provided in the Supplementary Material (Figure S2).

5.2 Comparison with alternative subset-search methods

To further contextualize EPR-C3 relative to representative subset-selection paradigms, Figure 3 compares the recovery of reference subsets and the corresponding computational cost across benchmark methods. CPU times are reported relative to Forward Stepwise, which is assigned a reference value of 1.0.

For interpretive clarity, the benchmarked methods were organized into methodological families: filter-wrapper screening (CSR-AS), stepwise procedures (Forward and Backward Stepwise), penalized-regression preselection followed by OLS subset refinement (Elastic Net, Ridge, and Lasso), stochastic metaheuristic search (genetic algorithm baseline), branch-and-bound search (Leaps), and the proposed deterministic neighborhood-search heuristic EPR-C3.

The weakest recovery was observed for methods that restrict the search space early or follow a single greedy trajectory. CSR-AS failed to recover any subset from the top 100 reference subsets, despite requiring 52.1 times the CPU time of Forward Stepwise. This result indicates that aggressive marginal screening and redundancy filtering can remove predictors that are not individually dominant but become relevant within high-ranking multivariate subsets. Similarly, Forward Stepwise recovered no reference subset, whereas Backward Stepwise recovered only 4%. Although these procedures were computationally inexpensive, with relative CPU times of 1.0 and 3.8, respectively, their limited recovery confirms that greedy sequential inclusion or elimination is poorly suited to reconstructing the broader admissible subset landscape.

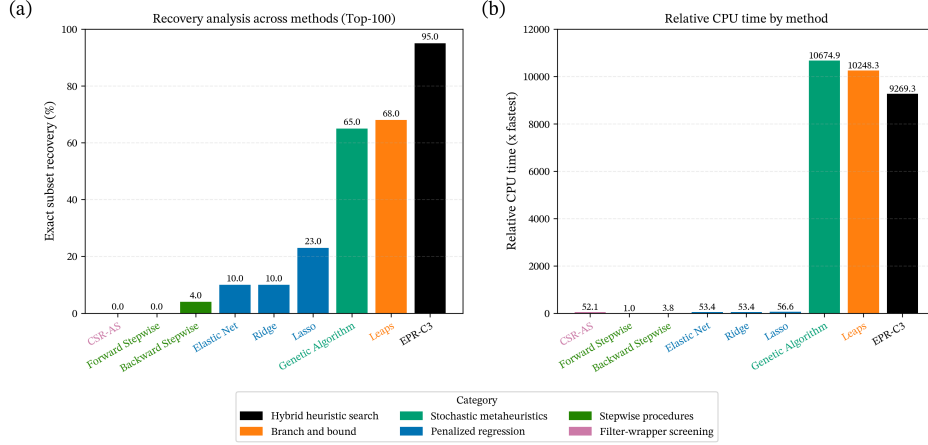


Fig. 3 Comparative subset recovery and computational efficiency of representative subset-search methods. Panel (a) reports exact recovery of the top 100 reference subsets; panel (b) reports CPU time relative to Forward Stepwise. Methods are grouped as stepwise procedures, penalized-regression preselection followed by OLS refinement, filter-wrapper screening, branch-and-bound subset search, stochastic metaheuristic search, and the proposed deterministic neighborhood-search heuristic.

Penalized-regression preselection improved recovery modestly but remained far from reference behavior. Elastic Net and Ridge each recovered 10% of the top 100 reference subsets, whereas Lasso recovered 23%. Their computational costs were moderate and similar, with relative CPU times of 53.4, 53.4, and 56.6, respectively. These results show that shrinkage-based ranking can provide a useful reduced candidate pool, particularly for Lasso, but does not reliably recover the high-ranking admissible OLS subset space. This limitation is expected because penalized models optimize a regularized estimation objective, whereas the benchmark evaluates explicit OLS subsets under statistical and multicollinearity admissibility constraints.

Search-driven methods showed substantially stronger recovery, but with different recovery–efficiency profiles. The genetic algorithm baseline recovered 65% of the top 100 reference subsets, confirming that broader combinatorial exploration is more effective than greedy or screening-based procedures. However, this improvement required a high computational cost, reaching 10,674.9 times the CPU time of Forward Stepwise. Leaps recovered 68% of the reference subsets with a comparable relative CPU time of 10,248.3. Although Leaps is designed to identify highly ranked subsets efficiently through branch-and-bound search, in the evaluated workflow the key admissibility constraints were applied after subset generation rather than during the branch-and-bound search. Therefore, many highly ranked subsets identified by Leaps can be discarded during post-processing because they violate correlation, VIF, significance, or other admissibility criteria. Nevertheless, recovery reached only 68%, despite the use of a broad retained-candidate budget intended to increase the probability of recovering 100 admissible top-ranked subsets after filtering ($n_{\text{best}} = 35000$). This behavior highlights a central limitation of applying branch-and-bound subset ranking to constraint-rich

MLR workflows: high ranking under the internal search criterion does not guarantee admissibility under the final modeling requirements.

Under this benchmark configuration, EPR-C3 achieved the highest recovery among all evaluated methods, recovering 95% of the top 100 reference subsets. Its relative CPU time was 9,269.3, lower than both the genetic algorithm baseline and Leaps under the benchmark conditions. The advantage of EPR-C3 therefore lies not only in broader exploration but also in coupling structured neighborhood moves with constraint-aware acceptance and refinement. By incorporating correlation control, VIF pruning, significance filtering, and admissibility checks into the search workflow, EPR-C3 directs computational effort toward subsets that are more likely to survive final post-processing.

The joint interpretation of Panels (a) and (b) in Figure 3 reinforces the methodological premise of EPR-C3. Fast methods generally failed to recover the admissible high-ranking subset space, whereas broader search methods improved recovery at substantially greater computational cost. However, Leaps and the genetic algorithm baseline demonstrate that broader exploration alone is insufficient to guarantee near-exhaustive admissible recovery. Leaps identifies many highly ranked subsets but applies the key constraints only after search, while the genetic algorithm baseline explores more broadly but remains computationally expensive and stochastic. In contrast, EPR-C3 provides a more favorable recovery–efficiency tradeoff by embedding constraint-aware decisions within a deterministic multi-start neighborhood-search framework.

These results should therefore be interpreted not as a comparison of predictive modeling strategies in general, but as a comparison of their ability to reconstruct the admissible high-ranking OLS subset space. From this perspective, Figure 3 supports the central contribution of EPR-C3: targeted, deterministic, and constraint-aware subset exploration can approximate reference recovery while avoiding the full combinatorial burden of exhaustive enumeration and the post hoc admissibility losses observed in branch-and-bound subset ranking.

5.3 Scope of EPR-C3

The comparative results indicate that EPR-C3 can recover high-ranking admissible MLR subsets more effectively than the alternative search strategies evaluated in this study. Nevertheless, the method should not be interpreted as a replacement for exhaustive enumeration whenever all-subsets regression is computationally feasible. Exhaustive search remains the reference procedure because it evaluates the complete subset space and provides an exact ranking under the specified objective and constraints. The practical role of EPR-C3 is therefore complementary: it is intended for regimes with large candidate feature or predictor spaces in which exhaustive enumeration becomes computationally prohibitive, but where model transparency, OLS interpretability, reproducibility, and statistical admissibility remain essential.

Within this scope, EPR-C3 occupies an intermediate methodological position between all-subsets regression and more flexible stochastic metaheuristics. Unlike greedy stepwise procedures, it is not restricted to a single sequential inclusion or

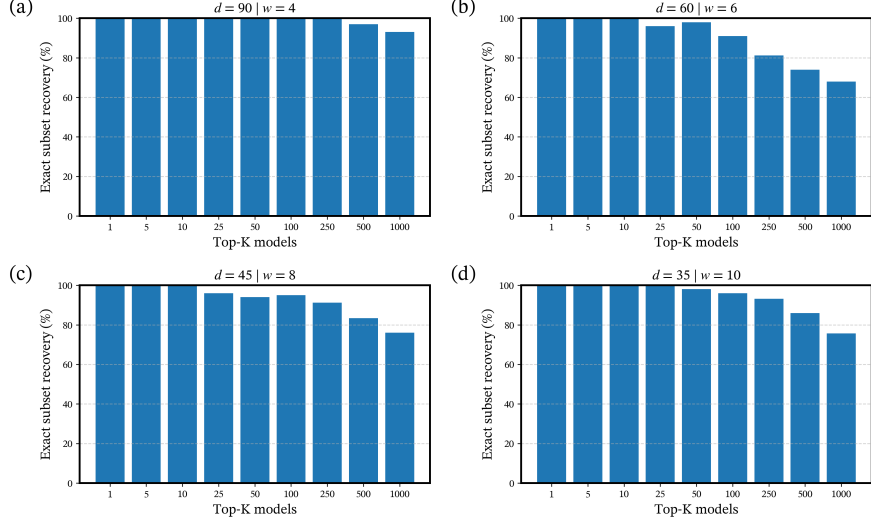


Fig. 4 Top- K exact recovery of all-subsets solutions by EPR-C3. Panels (a–d) report the percentage of exhaustive Top- K subsets recovered exactly by EPR-C3 ($K = 1$ –1000). All EPR-C3 searches were conducted using 1000 seeds. The analysis covers four predictor-space configurations with increasing maximum subset size: (a) $d = 90$, $w = 4$; (b) $d = 60$, $w = 6$; (c) $d = 45$, $w = 8$; and (d) $d = 35$, $w = 10$, where d denotes the total number of candidate predictors and w the maximum number of predictors per model.

elimination path. Unlike general-purpose stochastic metaheuristics, it follows a deterministic multi-start neighborhood-search logic under fixed seeds. Unlike post hoc filtering workflows, it incorporates admissibility constraints into the search and refinement process. This combination is central to its intended use: recovering competitive, statistically admissible OLS subsets without requiring complete enumeration of the combinatorial space.

This positioning is particularly relevant to data-science and statistical-computing workflows in which the objective is not only to maximize a predictive score but also to obtain solutions that remain interpretable, reproducible, and statistically defensible. EPR-C3 addresses this requirement by embedding subset selection within a constraint-aware admissibility framework, where correlation control, multicollinearity reduction, significance filtering, and model-size restrictions are treated as part of the computational search process rather than as purely external post-processing steps.

To further define the practical scope of the method, we examined whether the computational advantages of EPR-C3 are accompanied by reliable recovery of the highest-ranked all-subsets solutions. Figure 4 reports the percentage of exhaustive Top- K subsets that were exactly recovered by EPR-C3 across four predictor-space configurations with increasing maximum subset size.

Across all evaluated configurations, EPR-C3 achieved very high recovery of the highest-ranked exhaustive all-subsets models, with complete or nearly complete agreement for the smallest Top- K values. This result indicates that the structured search

procedure reliably identifies the most competitive subsets, which are typically the models of greatest practical relevance in MLR-based model selection.

The recovery profile was strongest for the $d = 90$, $w = 4$ configuration, where EPR-C3 remained close to complete agreement across the full Top- K range. As the maximum subset size increased from $w = 6$ to $w = 10$, recovery remained high for the best-ranked solutions but decreased progressively when broader ranked lists were considered. This trend is expected because increasing w expands the combinatorial search space and increases the number of statistically similar or near-equivalent admissible subsets. Under these conditions, exact agreement with the exhaustive ranking becomes more difficult, especially for lower-ranked models within large Top- K lists.

Importantly, the observed decrease in exact recovery is concentrated mainly in broader Top- K ranges rather than among the leading solutions. This distinction is central to the intended use of EPR-C3. In practical analytics and model-selection workflows, the highest-ranked admissible subsets usually carry the greatest interpretive and decision-making value, whereas exact reconstruction of every lower-ranked alternative is less critical. Therefore, the results in Figure 4 support the use of EPR-C3 as a computationally efficient approximation to all-subsets regression in large-subset-space or combinatorially demanding settings.

Taken together, these findings define the operational scope of EPR-C3. When exhaustive enumeration becomes impractical, EPR-C3 provides a deterministic and constraint-aware alternative that preserves the most relevant model-selection outcomes while substantially reducing the computational burden associated with complete subset enumeration.

5.4 Applied case study

Based on the findings from the benchmark, EPR-C3 was run on the 53-predictor pK_a dataset using a configuration of 1000 seeds and a maximum model size of six predictors, with the same admissibility rules applied in the benchmark. Under this configuration, EPR-C3 was approximately 2.5 times faster than all-subsets regression while recovering the same predictor set and OLS-refitted training equation previously reported:

$$\begin{aligned} pK_a = & 0.1074 \Delta E - 0.1422 \Delta HL_{\text{Gap}} \\ & - 1.4887 \Delta ALIE_N - 0.9132 \chi_M \\ & + 0.0151 \%NPSA + 3.0608 \text{BaseT} \\ & - 30.7140. \end{aligned} \tag{10}$$

The case study illustrates three practical aspects of the method. First, EPR-C3 can recover an interpretable published MLR equation while enforcing explicit statistical and multicollinearity constraints. Second, the method can reduce runtime when the candidate subset space is sufficiently large to justify heuristic search. Third, the method remains compatible with downstream validation and diagnostic workflows because every candidate solution is an ordinary least squares model with an explicit predictor subset.

6 Limitations and future work

EPR-C3 should be interpreted as a search heuristic for large subset spaces, not as an exact optimizer. Its recovery behavior was validated only in regimes where all-subsets regression remained computationally feasible. Beyond those regimes, especially above 3×10^8 candidate subsets, larger seed budgets may be required to maintain robust recovery. The empirical utility threshold in Eq. (9) is likewise operational rather than theoretical, because it depends on implementation efficiency, seed budget, hardware configuration, parallelization strategy, and admissibility settings.

Future work should examine adaptive seed generation, convergence diagnostics, and extensions to generalized linear models, nonlinear regression, and hybrid OLS-machine-learning workflows for broader data-science applications. These directions are promising but require separate methodological validation.

7 Conclusions

This work presents EPR-C3 as a deterministic, multi-start, constraint-aware neighborhood-search heuristic for feature or predictor subset selection in high-dimensional data within an OLS-based statistical-modeling framework. The method is not intended to replace all-subsets regression when complete enumeration is feasible; rather, it should be used selectively once the candidate subset space exceeds the empirical utility threshold at which exhaustive search becomes inefficient.

Benchmark results show that 1000 seed trajectories provide a practical baseline for robust recovery under the evaluated conditions and that EPR-C3 becomes advantageous in highly combinatorial problems. Comparative results against step-wise procedures, penalized-regression preselection, filter-wrapper screening, the genetic algorithm baseline, and branch-and-bound support its favorable recovery-efficiency tradeoff. This comparison highlights the importance of embedding admissibility constraints within the search workflow rather than applying them only as post hoc filters. EPR-C3 mitigates post-filtering losses by coupling structured neighborhood exploration with constraint-aware acceptance and refinement.

An applied case study further demonstrates reproducible model recovery and a notable runtime reduction relative to exhaustive search. Overall, EPR-C3 is intended as a practical constraint-aware subset-search strategy for high-dimensional MLR problems where exhaustive enumeration becomes operationally impractical and interpretable, reproducible, statistically admissible linear models remain important. By combining deterministic heuristic search with explicit OLS evaluation, statistical admissibility rules, and transparent equation-based model representation, EPR-C3 provides an interpretable data-science and statistical-computing approach for high-dimensional model selection.

Supplementary information. The algorithmic summaries of the benchmark methods, and supplementary figure descriptions are provided as Supplementary Material.

Acknowledgements. The author gratefully acknowledges ANID and the Instituto de Ciencias e Innovación en Medicina (ICIM) at Universidad del Desarrollo for

their support and for providing access to computational resources acquired through FONDEQUIP project EQM150093. The author also thanks Alessandra Misad for her thorough proofreading of the manuscript.

Declarations

Funding. This research was supported by ANID-FONDECYT grant No. 3240378.

Competing interests. The author declares no competing interests.

Data availability. All data supporting the findings of this study are available within the paper, and its Supplementary Information.

Code availability. The EPR-C3 method is implemented in the author-maintained MLR-X project and is available at: <https://jacksonalcazar.github.io/MLR-X/>.

Author contribution. J.A. conceived the study, designed the methodology, analyzed the data, and wrote the entire manuscript.

References

- [1] Andersen, P.K., Skovgaard, L.T.: 8. Multiple regression, the linear predictor, pp. 231–302. Springer, New York, NY (2010). <https://doi.org/10.1007/978-1-4419-7170-8>
- [2] Uyanık, G.K., Güler, N.: A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences* **106**, 234–240 (2013) <https://doi.org/10.1016/j.sbspro.2013.12.027>
- [3] Hope, T.M.: 4. Linear regression, pp. 67–81. Elsevier, Cambridge, MA, USA (2020)
- [4] Sun, Y., Wang, X., Zhang, C., Zuo, M.: Multiple regression: Methodology and applications. *Highlights in Science, Engineering and Technology* (2023) <https://doi.org/10.54097/hset.v49i.8611>
- [5] Ruan, Y.: Exploring multiple regression models: Key concepts and applications. *Science and Technology of Engineering, Chemistry and Environmental Protection* (2024) <https://doi.org/10.61173/yjpt3s59>
- [6] Tenekedjiev, K., Abdussamie, N., An, H.-J., Nikolova, N.: Regression diagnostics with predicted residuals of linear model with improved singular value classification applied to forecast the hydrodynamic efficiency of wave energy converters. *Applied Sciences* (2021) <https://doi.org/10.3390/app11072990>
- [7] James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J.: 3. Linear Regression, pp. 69–134. Springer, Cham, Switzerland (2023). <https://doi.org/10.1007/978-3-031-38747-0.3>

- [8] Roustaei, N.: Application and interpretation of linear-regression analysis. *Medical Hypothesis, Discovery and Innovation in Ophthalmology* **13**, 151–159 (2024) <https://doi.org/10.51329/mehdiophthal1506>
- [9] Lenz, A., Haktanir, A., Flores, D.: Statistical power, reliability, and model assumptions in multiple linear regression: Assessing and addressing common issues. *Measurement and Evaluation in Counseling and Development* **59**, 94–107 (2026) <https://doi.org/10.1080/07481756.2025.2589207>
- [10] Brooks, G.P., Ruengvirayudh, P.: Best-subset selection criteria for multiple linear regression. *General Linear Model Journal* **42**(2), 14–25 (2016)
- [11] Heinze, G., Wallisch, C., Dunkler, D.: Variable selection – a review and recommendations for the practicing statistician. *Biometrical Journal* **60**, 431–449 (2018) <https://doi.org/10.1002/bimj.201700067>
- [12] Khan, D.M., Yaqoob, A., Iqbal, N., Wahid, A., Khalil, U., Khan, M., Abd Rahman, M.A., Mustafa, M.S., Khan, Z.: Variable selection via scad-penalized quantile regression for high-dimensional count data. *IEEE Access* **7**, 153205–153216 (2019) <https://doi.org/10.1109/ACCESS.2019.2948278>
- [13] Ullmann, T., Heinze, G., Hafermann, L., Schilhart-Wallisch, C., Dunkler, D.: Evaluating variable selection methods for multivariable regression models: A simulation study protocol. *PLOS ONE* **19** (2024) <https://doi.org/10.1371/journal.pone.0308543>
- [14] Mohamad, M., Selamat, A., Krejcar, O., Crespo, R.G., Herrera-Viedma, E., Fujita, H.: Enhancing big data feature selection using a hybrid correlation-based feature selection. *Electronics* **10**(23), 2984 (2021) <https://doi.org/10.3390/electronics10232984>
- [15] Liu, F., Panagiotakos, D.: Real-world data: a brief review of the methods, applications, challenges and opportunities. *BMC Medical Research Methodology* **22**(1), 287 (2022) <https://doi.org/10.1186/s12874-022-01768-6>
- [16] Brown, J.D.: 8. Model Selection and Biased Estimation, pp. 253–288. Springer, Cham, Switzerland (2018). https://doi.org/10.1007/978-3-319-93549-2_8
- [17] Kim, J.H.: Multicollinearity and misleading statistical results. *Korean Journal of Anesthesiology* **72**, 558–569 (2019) <https://doi.org/10.4097/kja.19087>
- [18] Mizumoto, A.: Calculating the relative importance of multiple regression predictor variables using dominance analysis and random forests. *Language Learning* (2022) <https://doi.org/10.1111/lang.12518>
- [19] Desboulets, L.D.D.: A review on variable selection in regression analysis. *Econometrics* **6**(4), 45 (2018) <https://doi.org/10.3390/econometrics6040045>

- [20] Fan, J., Lv, J.: Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **70**(5), 849–911 (2008) <https://doi.org/10.1111/j.1467-9868.2008.00674.x>
- [21] Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **58**(1), 267–288 (1996) <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- [22] Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**(1), 55–67 (1970) <https://doi.org/10.1080/00401706.1970.10488634>
- [23] Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **67**(2), 301–320 (2005) <https://doi.org/10.1111/j.1467-9868.2005.00503.x>
- [24] Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**(456), 1348–1360 (2001) <https://doi.org/10.1198/016214501753382273>
- [25] Zhang, C.-H.: Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* **38**(2), 894–942 (2010) <https://doi.org/10.1214/09-AOS729>
- [26] Greenwood, C.J., Youssef, G.J., Letcher, P., Macdonald, J.A., Hagg, L.J., Sanson, A., McIntosh, J., Hutchinson, D.M., Toumbourou, J.W., Fuller-Tyszkiewicz, M., *et al.*: A comparison of penalised regression methods for informing the selection of predictive markers. *PloS One* **15**(11), 0242730 (2020) <https://doi.org/10.1371/journal.pone.0242730>
- [27] Efroymson, M.A.: Multiple regression analysis. *Mathematical methods for digital computers*, 191–203 (1960)
- [28] Ruengvirayudh, P., Brooks, G.P.: Comparing stepwise regression models to the best-subsets models, or, the art of stepwise. *General Linear Model Journal* **42**(1), 1–14 (2016)
- [29] Olusegun, A.M., Dikko, H.G., Gulumbe, S.U.: Identifying the limitation of stepwise selection for variable selection in regression analysis. *American Journal of Theoretical and Applied Statistics* **4**(5), 414–419 (2015) <https://doi.org/10.11648/j.ajtas.20150405.22>
- [30] Furnival, G.M., Wilson, R.W.: Regressions by leaps and bounds. *Technometrics* **42**(1), 69–79 (2000) <https://doi.org/10.1080/00401706.2000.10485982>
- [31] Broadhurst, D., Goodacre, R., Jones, A., Rowland, J., Kell, D.: Genetic algorithms as a method for variable selection in multiple linear regression and partial

- least squares regression, with applications to pyrolysis mass spectrometry. *Analytica Chimica Acta* **348**, 71–86 (1997) [https://doi.org/10.1016/S0003-2670\(97\)00065-2](https://doi.org/10.1016/S0003-2670(97)00065-2)
- [32] Örkücü, H.H.: Subset selection in multiple linear regression models: A hybrid of genetic and simulated annealing algorithms. *Applied Mathematics and Computation* **219**, 11018–11028 (2013) <https://doi.org/10.1016/j.amc.2013.05.016>
 - [33] Trejos, J., Villalobos-Arias, M.A., Espinoza, J.L.: 5. Variable selection in multiple linear regression using a genetic algorithm, pp. 133–159. IGI Global Scientific Publishing, Hershey, PA, USA (2016). <https://doi.org/10.4018/978-1-4666-9644-0.ch005>
 - [34] Yalçinkaya, A., Gedik Balay, Şenoğlu, B.: A new approach using the genetic algorithm for parameter estimation in multiple linear regression with long-tailed symmetric distributed error terms: An application to the covid-19 data. *Chemometrics and Intelligent Laboratory Systems* **216**, 104372 (2021) <https://doi.org/10.1016/j.chemolab.2021.104372>
 - [35] Alhijawi, B., Awajan, A.: Genetic algorithms: theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*, 1–12 (2023) <https://doi.org/10.1007/s12065-023-00822-6>
 - [36] Lumley, T.: Leaps: Regression Subset Selection. (2024). R package version 3.2. <https://CRAN.R-project.org/package=leaps>
 - [37] Alcázar, J.J., Misad Saide, A.C., Campodónico, P.R.: Reliable and accurate prediction of basic pka values in nitrogen compounds: The pka shift in supramolecular systems as a case study. *Journal of Cheminformatics* **15**(1), 90 (2023) <https://doi.org/10.1186/s13321-023-00763-3>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Sl.zip](#)