frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 13 February 2025
DOI 10.3389/fgene.2024.1462855

Check for updates

OPEN ACCESS

EDITED BY
Ruzong Fan,
Georgetown University Medical Center,
United States

REVIEWED BY
Zitong Li,
Commonwealth Scientific and Industrial
Research Organisation (CSIRO), Australia
Matias Bermann,
University of Georgia, United States

*CORRESPONDENCE
Alexandra Stadler,
✉ alexandra.stadler@jku.at
Werner G. Müller,
✉ werner.mueller@jku.at
Andreas Futschik,
✉ andreas.futschik@jku.at

RECEIVED 10 July 2024
ACCEPTED 20 December 2024
PUBLISHED 13 February 2025

CITATION
Stadler A, Müller WG and Futschik A (2025) A comparison of design algorithms for choosing the training population in genomic models. *Front. Genet.* 15:1462855. doi: 10.3389/fgene.2024.1462855

# A comparison of design algorithms for choosing the training population in genomic models

Alexandra Stadler*, Werner G. Müller* and Andreas Futschik*

Institute of Applied Statistics, Johannes Kepler University, Linz, Austria

In contemporary breeding programs, typically genomic best linear unbiased prediction (gBLUP) models are employed to drive decisions on artificial selection. Experiments are performed to obtain responses on the units in the breeding program. Due to restrictions on the size of the experiment, an efficient experimental design must usually be found in order to optimize the training population. Classical exchange-type algorithms from optimal design theory can be employed for this purpose. This article suggests several variants for the gBLUP model and compares them to brute-force approaches from the genomics literature for various design criteria. Particular emphasis is placed on evaluating the computational runtime of algorithms along with their respective efficiencies over different sample sizes. We find that adapting classical algorithms from optimal design of experiments can help to decrease runtime, while maintaining efficiency.

KEYWORDS

**GBLUP, genomic selection, optimal design, training population selection, design of experiments (DoE)**

## 1 Introduction

Plant breeding has been done by humans for centuries and has only grown in importance in recent history due to increasing global demand for food. Nowadays, it is possible to incorporate genetic information in breeding programs to improve decisions on artificial selection (Hickey et al., 2017). Researchers have investigated different statistical modeling methods for data analysis in this field. However, there are still a number of open questions w.r.t. model-oriented experimental design for phenotypic data collection in plant breeding programs. This problem is well-known in the field as optimization of the training set for model fit in genomic selection. Particularly, the search for optimum experimental designs under the genomic best linear unbiased prediction (gBLUP) model has not been studied extensively.

This article provides insights into the optimal design problem in the gBLUP model that is frequently applied in genomic selection. Section 2 is concerned with some preliminary information about the gBLUP model and a review of the generalized coefficient of determination, which is necessary for the introduction of the CDMin-criterion, an established criterion in breeding experiments. The optimal design problem is specified in Section 3 with emphasis on the particularities and differences to optimal design of experiments in classical linear models. Subsequently, different algorithms for the heuristic search of exact optimal designs are outlined in Section 4. A related discussion in a similar context is provided in Butler et al. (2021). The algorithms mentioned in Section 4 are applied to data and compared to algorithms provided in the `TrainSel` R package by

Akdemir et al. (2021). The setup for this comparison is described in Section 5 and results are stated w.r.t. criterion values achieved as well as runtime in Section 6. A discussion on extensions of this work is given in Section 7. We provide conclusions in the final Section 8.

## 2 The gBLUP model

The gBLUP model is a special case of a linear mixed model with individual-specific random effects. Consider the following linear mixed model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{y}$ is a $n \times 1$-vector of observed values, $\mathbf{X}$ is an $n \times p$ matrix, with corresponding $p \times 1$-vector $\boldsymbol{\beta}$ of fixed effects, and $\mathbf{Z} = \mathbf{I}_n$ being the identity matrix of dimension $n \times n$ with corresponding $n \times 1$-vector $\boldsymbol{\gamma}$ of individual-specific random effects. The $n \times 1$-vector $\boldsymbol{\varepsilon}$ denotes the random errors. The vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\varepsilon}$ are independent of each other and follow a normal distribution with

$$\begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\varepsilon} \end{pmatrix} \sim \mathbf{N}\left( \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \sigma_g^2 \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \sigma_e^2 \mathbf{I}_n \end{pmatrix} \right).$$

It is assumed that the genomic relationship matrix (GRM) $\mathbf{G}$ and the ratio $\lambda = \frac{\sigma_e^2}{\sigma_g^2}$ are known. Further, it is assumed that $\mathbf{G}$ is nonsingular. For convenience, define $\mathbf{V} = \sigma_g^2 \mathbf{Z}\mathbf{G}\mathbf{Z}' + \sigma_e^2 \mathbf{I}_n$ for subsequent formulas.

The GRM quantifies the degree of relatedness between plant lines in a breeding scheme. There are several ways to obtain such a matrix from genetic marker data (e.g., single-nucleotide polymorphism marker data). The calculation of the GRM is not the subject of this article, thus, for further information on this, the reader may want to refer to Clark and van der Werf (2013). For a detailed introduction to the gBLUP model and its relationship to other models in plant breeding, it is useful to refer to Habier et al. (2007) and VanRaden (2008).

The fixed and random effects in model (1) can be estimated/predicted via Henderson's mixed model equations, see, e.g., Henderson (1975) and Robinson (1991). The estimators/predictors can be written as

$$\begin{aligned} \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\gamma}} \end{pmatrix} &= \left( \mathbf{D}'\sigma_e^{-2}\mathbf{I}_n^{-1}\mathbf{D} + \sigma_g^{-2}\mathbf{B} \right)^{-1} \mathbf{D}'\sigma_e^{-2}\mathbf{I}_n^{-1}\mathbf{y} \\ &= \left( \sigma_e^{-2}\mathbf{D}'\mathbf{D} + \sigma_g^{-2}\mathbf{B} \right)^{-1} \sigma_e^{-2}\mathbf{D}'\mathbf{y}, \end{aligned}$$

with

$$\mathbf{D} = (\mathbf{X}\,\mathbf{Z}), \qquad \mathbf{B} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} \end{pmatrix}.$$

The covariance matrix of the estimators/predictors is

$$\begin{aligned} \mathrm{E}\left\{ \begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma} \end{pmatrix}' \right\} &= \left( \sigma_e^{-2}\mathbf{D}'\mathbf{D} + \sigma_g^{-2}\mathbf{B} \right)^{-1} \\ &\propto \left( \mathbf{D}'\mathbf{D} + \lambda\mathbf{B} \right)^{-1} = \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{pmatrix}, \end{aligned} \tag{2}$$

where $\mathbf{H}_{11}$ is of dimension $p \times p$ and it is the (proportional) covariance matrix of the fixed effects and correspondingly for the other submatrices.

The prediction variance of observed units is therefore

$$\mathrm{Var}(\hat{\mathbf{y}}) \propto \mathrm{diag}\left\{ \mathbf{D}\left( \mathbf{D}'\mathbf{D} + \lambda\mathbf{B} \right)^{-1} \mathbf{D}' \right\}.$$

Let the covariance between random effects of units $\mathbf{x}_i$ and $\mathbf{x}_j$ be denoted by $\sigma_g^2 \mathbf{G}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Cov}(\gamma_i, \gamma_j)$ and $\mathbf{G}(\mathbf{x}_i, \mathbf{x}_i) = \mathbf{G}(\mathbf{x}_i)$ with the appropriate extension to a covariance matrix of several individuals.

The prediction variance of an unobserved individual is

$$\mathrm{Var}(\hat{\gamma}_0) \propto \mathbf{d}_0 \left( \mathbf{D}'\mathbf{D} + \lambda\mathbf{B} \right)^{-1} \mathbf{d}_0'$$

where $\mathbf{d}_0 = (\mathbf{x}_0 \quad \mathbf{z}_0)$ is a row-vector of dimension $n + p$ with $\mathbf{z}_0 = \mathbf{G}(\mathbf{x}_0, \mathbf{X})\mathbf{G}(\mathbf{X})^{-1}$.

After model-fitting, breeders are mainly interested in the predicted random effects of all individuals potentially available for breeding. The predicted effects are called estimated genomic breeding values (GEBVs) and serve as a selection criterion, where the ranking of the breeding values is important (Jannink et al., 2010). The GEBVs of the individuals used for model fitting provide the predicted random effects $\hat{\boldsymbol{\gamma}}$. For individuals not included in the fitted model, the random effects must be rescaled to account for relationships between individuals. Let $\mathbf{z}_0 = \mathbf{G}(\mathbf{x}_0, \mathbf{X})\mathbf{G}(\mathbf{X})^{-1}$, where $\mathbf{x}_0$ denotes the vector of fixed effects for an unobserved individual. Then the GEBV of this individual is given by $\mathbf{z}_0\hat{\boldsymbol{\gamma}}$ (Clark and van der Werf, 2013).

Since breeders are mainly interested in the predictions of random effects, the goodness of a model shall be evaluated on the precision of the GEBVs in the sense that the ordering is most accurate. Arbitrary rescaling of the predicted random effects does not influence breeders' selection decisions, hence, it is not important. To utilize a measure that is most useful in this respect, Laloë (1993) has introduced the generalized coefficient of determination with the purpose of quantifying the precision of the prediction.

Let the coefficient of determination (CD) of the random effect of unit $i$ be defined by

$$\mathbf{CD}(\mathbf{x}_i|\mathbf{X}) = \frac{\mathrm{Var}(\hat{\gamma}_i)}{\mathrm{Var}(\gamma_i)} = 1 - \frac{\mathrm{Var}(\gamma_i|\hat{\gamma}_i)}{\mathrm{Var}(\gamma_i)}, \tag{3}$$

where the design matrix in the model fit to predict $\hat{\boldsymbol{\gamma}}$ is $\mathbf{X}$.

As can be seen in Equation 3, the CD of an individual effect is a function of the variance ratio before and after the experiment. Consequently, CD is in the range of $[0, 1]$. As described in Laloë (1993), the CD measures the squared correlation between the predicted and the realized random effect for an individual and thus measures the amount of information supplied by the data to obtain a prediction.

A matrix of CD values can easily be computed by

$$\mathbf{CD}(\mathbf{X}_0|\mathbf{X}) = \mathrm{diag}\left[ \left\{ \mathbf{G}(\mathbf{X}_0, \mathbf{X})\mathbf{Z}'\mathbf{P}\mathbf{Z}\mathbf{G}(\mathbf{X}, \mathbf{X}_0) \right\} \oslash \mathbf{G}(\mathbf{X}_0, \mathbf{X}_0) \right],$$

where $\oslash$ is the element-wise (Hadamard) division and $\mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}$ (Akdemir et al., 2021; Isidro y Sanchéz and Akdemir, 2021). Notice that $\mathbf{X}$ is the matrix of fixed effects for model fitting and $\mathbf{X}_0$ is a design matrix of individuals. The concept of CD can be extended by introducing contrasts between individuals. For a given contrast $\mathbf{c}$ with appropriate dimension and $|\mathbf{c}| = 0$, the respective CD values are given by

$$\mathrm{diag}\left[ \left\{ \mathbf{c}'\mathbf{G}(\mathbf{X}_0, \mathbf{X})\mathbf{Z}'\mathbf{P}\mathbf{Z}\mathbf{G}(\mathbf{X}, \mathbf{X}_0)\mathbf{c} \right\} \oslash \mathbf{c}'\mathbf{G}(\mathbf{X}_0, \mathbf{X}_0)\mathbf{c} \right].$$

## 3 Optimal design problem

The design problem in the gBLUP model can be viewed as a selection of $n$ experimental units out of $N$ candidate units without replicates. Denote the design space as $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, i.e., it is the set of all units that can potentially be included in the design. Note that the design matrix $\mathbf{Z}$ is not included in this definition, since it is the same regardless of the design for a given size $n$ of the experiment, and replication of units is not considered. However, it is the case that the covariance of units $\mathbf{x}_i$ in the model, with $i = 1, \ldots, n$, is different w.r.t. the design.

The exact design of size $n < N$ is then written as $\xi_n$ and fulfills $\xi_n \subset \mathcal{X}$, $|\xi_n| = n$. Define the remaining set as $\zeta = \{x_i \in \mathcal{X}\colon \mathbf{x}_i \notin \xi_n\}$ and notice that $\xi \cup \zeta = \mathcal{X}$ and $\xi \cap \zeta = \emptyset$.

Let the inverse of the (proportional) covariance matrix in Equation 2 be called the information matrix and denote it relative to a design $\xi_n$ by

$$\mathbf{M}(\xi_n) = \mathbf{D}'(\xi_n)\mathbf{D}(\xi_n) + \lambda \mathbf{B}(\xi_n).$$

Due to the dependence of the GRM $\mathbf{G}$ on the exact design $\xi_n$, the information matrix is not additive w.r.t. single experimental units. This makes the application of well-known experimental design theory on the classical linear model infeasible.

An optimal exact design $\xi_n^\star$ of size $n$ is typically defined as

$$\xi_n^\star = \underset{\xi_n}{\operatorname{argmax}}\, \Phi\{\mathbf{M}(\xi_n)\},$$

for some optimal design criterion given by a function $\Phi\colon \mathbb{R}^{q \times q} \mapsto \mathbb{R}$ on the information matrix, where $q = n + p$.

Since breeders are mainly interested in the prediction of random effects in the gBLUP model, we restrict ourselves to criteria on the part of the covariance matrix that corresponds to random effects, i.e., to $\mathbf{H}_{22}$. In accordance with the literature on optimal design of experiments (cf. Atkinson et al., 2007), let the D-criterion on the covariance of random effects be defined by

$$\Phi_1\{\mathbf{M}(\xi_n)\} = -\ln|\mathbf{H}_{22}(\xi_n)|. \tag{4}$$

The D-criterion is a standard criterion, which minimizes the generalized variance of the parameter estimates (i.e., predictions in this application). Hence, it measures the precision of the predictors. For a geometrical interpretation of the D-criterion, see, e.g., Atkinson et al. (2007), Fedorov and Leonov (2013).

Additionally, consider the so-called CDMin-criterion, defined as

$$\Phi_2\{\mathbf{M}(\xi_n)\} = \min\{\mathbf{CD}(\mathcal{X}|\xi_n)\}$$

$$\begin{aligned}\Phi_2\{\mathbf{M}(\xi_n)\} &= \min\{\mathbf{CD}(\mathcal{X}|\xi_n)\}\\ &= \min\left(\operatorname{diag}\left[\{\mathbf{G}(\mathcal{X},\xi_n)\mathbf{Z}'\mathbf{PZG}'(\mathcal{X},\xi_n)\} \oslash \mathbf{G}(\mathcal{X})\right]\right). \end{aligned} \tag{5}$$

The CDMin criterion is the default criterion in the R package `TrainSel` (see Akdemir et al., 2021). As previously mentioned in Section 2, the CD of an individual measures the amount of information supplied by an experiment to predict the random effect of an individual. The CDMin-criterion maximizes the minimum CD value over all individuals. Hence, it is driven by the intuition of controlling for the worst case. The generalization of the CD was initially proposed by Laloë (1993) and has since gained popularity in the field, see e.g., Rincent et al. (2012). There are other criteria related to the CD, e.g., the CDMean criterion, which instead optimizes for the average CD. Refer to Akdemir et al. (2021) for a discussion on the CDMin-criterion versus other criteria. The literature on optimization criteria for this task is very rich and current, see e.g., also Fernández-González et al. (2023), where the authors compare different criteria w.r.t. accuracy of the predicted traits. However, the main focus of this manuscript is not on the selection of a suitable criterion, but rather on the efficient optimization once a criterion has been chosen.

In the following, we will concentrate on the two criteria stated above for clarity, but our methodology and discussion may extend well beyond those.

## 4 Heuristic search for an exact optimal design

Experimental design optimization in the gBLUP model is challenging in the sense that the covariance matrix of the random effects is directly dependent on the design. In particular, the information matrix is not a weighted sum of unit-specific matrices, which does not allow for standard convex design theory from the classical linear model to be applied directly. Due to this fact, it seems most reasonable to consider only optimization methods for exact designs rather than approximate designs (cf. Prus and Filová, 2024).

Even though the information matrix in the gBLUP model is quite different from the standard case, it may be sensible to adopt some ideas on algorithms from the classical linear model. This section reviews and alters algorithms for the application at hand. Furthermore, a comparison to the algorithm and R package `TrainSel` by Akdemir et al. (2021) will be made in the subsequent section.

The optimization strategy proposed by Akdemir et al. (2021) (implemented in `TrainSel`) combines a genetic algorithm with simulated annealing to identify local maxima in a given design region. It does so in a brute-force way without relying on theoretical considerations w.r.t. the design of experiments. The `TrainSel` algorithm provides flexibility by allowing for user-specified criteria, which comes at the cost of perhaps more efficient optimization strategies for specific criteria. The implemented function `TrainSel` offers several parameters that can be modified for a given situation; most relevant to this article are the maximum number of iterations for the algorithm, as well as the number of iterations without (significant) improvement until the algorithm is considered to have converged.

The published article Akdemir et al. (2021) refers to the author's publicly available R package on Github. Since then, the authors have made a major update to the repository in May 2024 (see Javier, 2024). The full functionality of the package is now under license, with the possibility of obtaining a free license for public bodies. A substantial part of the computations in this paper have been performed on `TrainSel v2.0`, the previously publicly available version on Github. To the best of our knowledge, this version is no longer available for download. We have made comparisons with the licensed version `Trainsel v3.0`, and have observed some changes, which will be discussed in subsequent sections. Most notably, the newest version permits

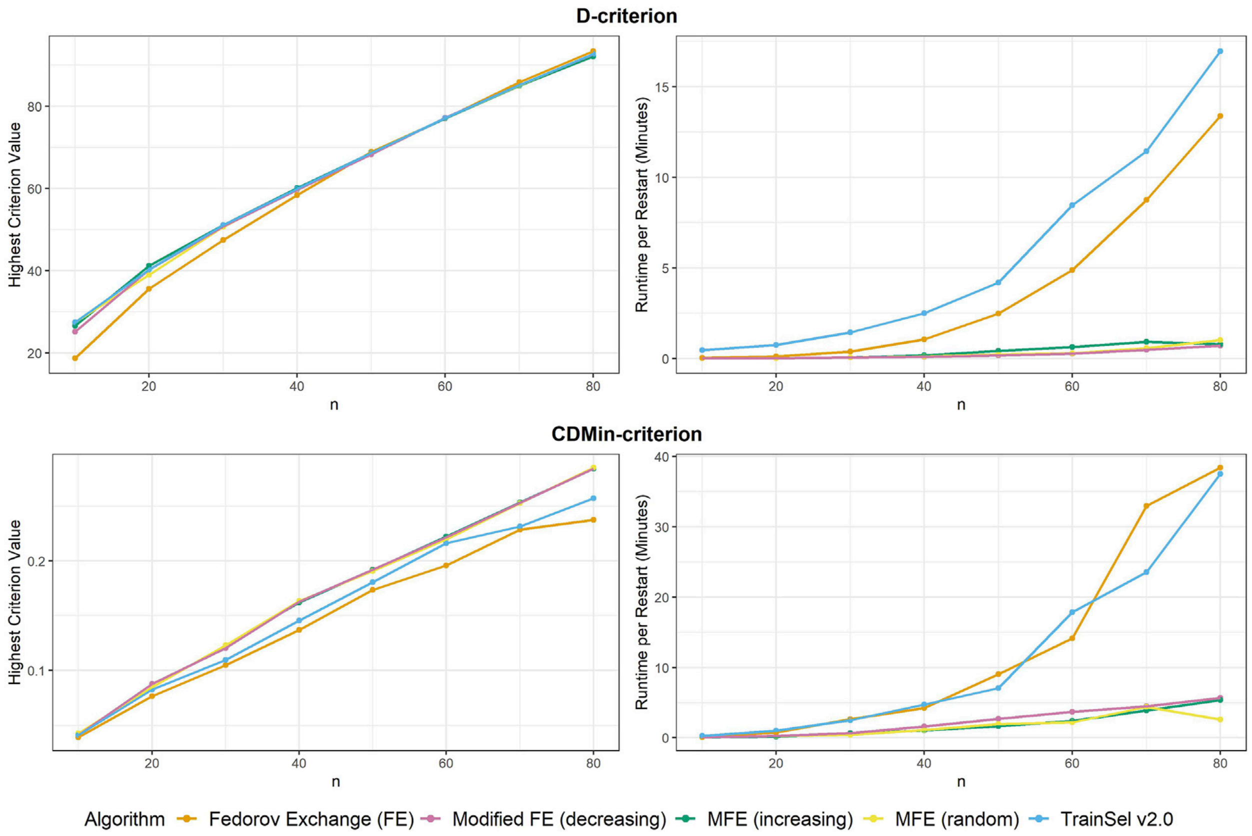

FIGURE 1
Highest criterion value and runtime per restart of different algorithms.

the manipulation of even more hyperparameters in the optimization algorithms and provides more user-friendliness in the specification of these parameters by introducing settings related to the size and complexity of the optimization problem. The optimization algorithm itself seems to have improved as well.

In contrast to this approach, the literature on experimental design typically proposes exchange-type algorithms in the search for exact designs (cf. Atkinson et al., 2007). Exchange-type algorithms start with an initial design $\xi_n$ of size $n$ (e.g., randomly sampled) and iteratively improve the design by exchanging units from the support and candidate set. In this case, the support and candidate set are mutually exclusive, and their union is the whole design space $\mathcal{X}$. The units are exchanged either according to the best improvement in the criterion (Fedorov exchange, see Fedorov, 1972) or according to a greedy strategy (modified Fedorov exchange, see Cook and Nachtsheim, 1980). In the classical Fedorov exchange algorithm, the criterion value for all possible exchanges between the candidate and the support set is calculated at each iteration. This guarantees that the most beneficial exchange is made in each step, which leads to quick ascension. The disadvantage is that the number of computations in each iteration can be large depending on the size of the support and candidate set. In the modified Fedorov exchange, the exchange is performed as soon as an improvement is identified, hence, the ordering of the support and candidate set has an impact on the search. The advantage of the modified Fedorov exchange over the classical one is the reduced computational effort in each iteration if a beneficial exchange can be found quickly. The final exact designs are dependent on the initial design, i.e., for randomly sampled starting designs, the final designs are also subject to randomness.

Exchange algorithms have been applied in the field of plant breeding, e.g., in Rincent et al. (2012) and Berro et al. (2019). However, these articles have not included any guidance on selecting potential instances for exchange. Rather, they have randomly performed an exchange and recalculated the criterion value. If an improvement could be observed, the exchange was kept. We believe that exchange algorithms can be improved in two ways, either the known best exchange can be made at each step, or the chance of making a good exchange can be increased by ordering the support and candidate set by some heuristic. These two approaches will be discussed below.

Unfortunately, none of the algorithms can guarantee convergence to the global optimum. Thus, it is reasonable to restart the algorithms at different random seeds to obtain multiple solutions, as is typically advised in the literature (see, e.g., Atkinson et al., 2007). The design with the highest criterion value over all random restarts should ultimately be chosen.

In the classical linear model, the most beneficial exchanges under the D-criterion are performed by exchanging the unit with minimum prediction variance in the support set with the unit with maximum prediction variance from the candidate set (cf. Atkinson

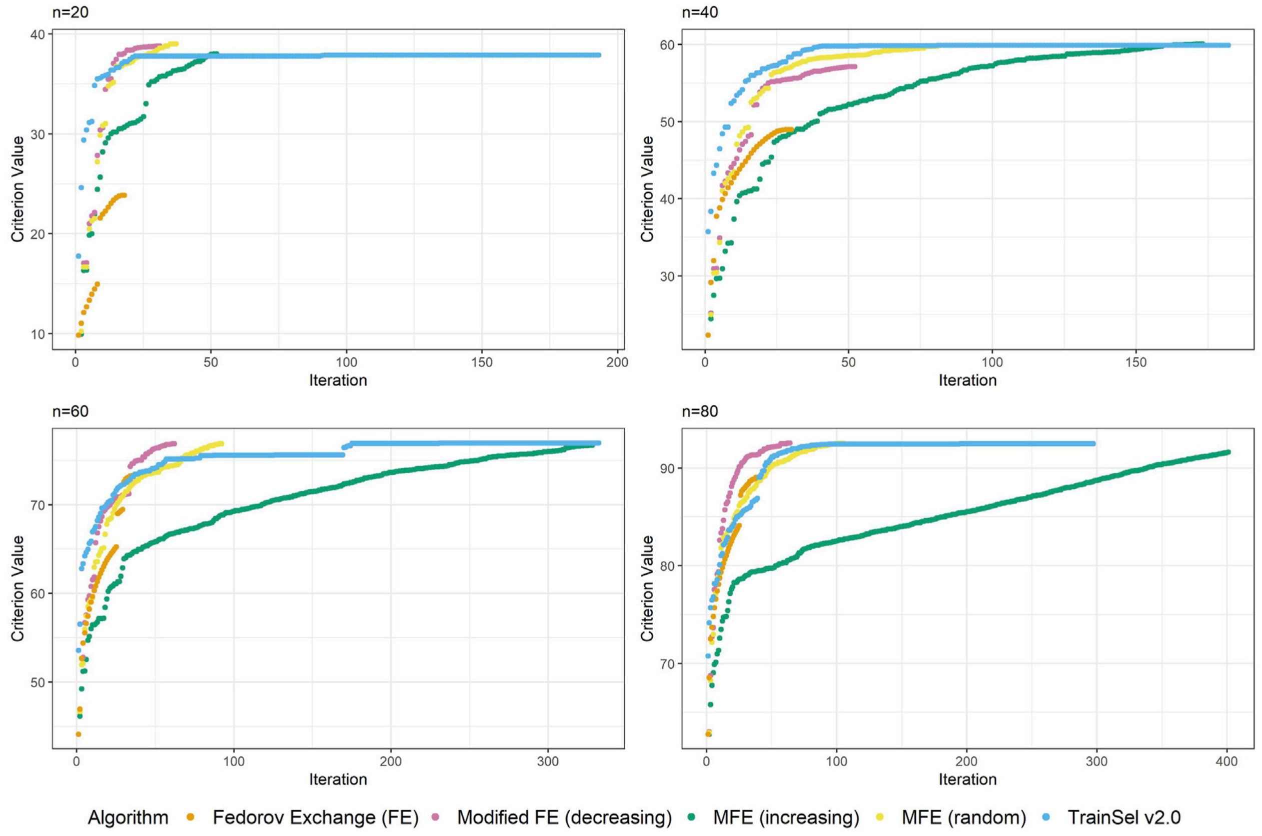

FIGURE 2
Convergence of algorithms over different sample size at one random seed (D-criterion).

et al., 2007). The reasoning is not as clear-cut in the gBLUP model as will be argued below. A modified Fedorov exchange can be performed with different orderings of the candidate and support set, which may lead to different resulting designs as well as different runtimes.

As stated, the candidate set in the modified Fedorov exchange is usually ordered w.r.t. decreasing prediction variance for the D-criterion, i.e., candidates with a high prediction variance are considered first for inclusion in the support set. Since the gBLUP model relies on individual-specific random effects, the removal of a unit from the support set has non-trivial consequences on the variance of other units. This makes it relatively difficult to specify a sensible ordering of the support set for exchanges. Since the D-criterion minimizes the generalized variance of the predicted random effects in this application, it seems reasonable to remove units (and thereby also random effects in the model) that have a high prediction variance. As this is completely converse to the reasoning in the classical linear (or mixed effect) model, we will here vary the ordering of the support set to increasing, decreasing, and random for the subsequent examples.

For the CDMin-criterion, following a similar thought process, it seems reasonable to add units with a small CD from the candidate set in the modified Fedorov exchange. In turn, units with a high CD could be sensible candidates for removal from the support set. In an effort to compare the ordering strategies, the ordering of the support set is also performed in increasing, decreasing, and random order as for the D-criterion in our examples.

The Fedorov exchange algorithm has a natural ending, i.e., when no further improvement can be found for any exchange of a unit from the candidate set with a unit from the support set, the best solution obtained is returned. For the modified Fedorov exchange, the same stopping rule is given, but the path and solution obtained are not necessarily the same. On the other hand the `TrainSel` algorithm does not have such a natural ending. Instead, the user must specify a maximum number of iterations for the algorithm and a number of iterations without a (significant) improvement on the criterion value, which will terminate the algorithm. The threshold for minimum improvement must also be specified by the user. Termination of this algorithm is defined as early stopping when no significant improvement on the criterion value was attained for a specified number of iterations.

## 5 Comparison of different algorithms

In the previous section, three algorithms have been introduced: `TrainSel`, Fedorov exchange and modified Fedorov exchange, where in the modified Fedorov exchange the support set can be ordered in different ways.

A comparison between these algorithms is performed on a dataset of $N = 200$ wheat lines given in the `TrainSel` R package (see Akdemir et al., 2021). The data consists of

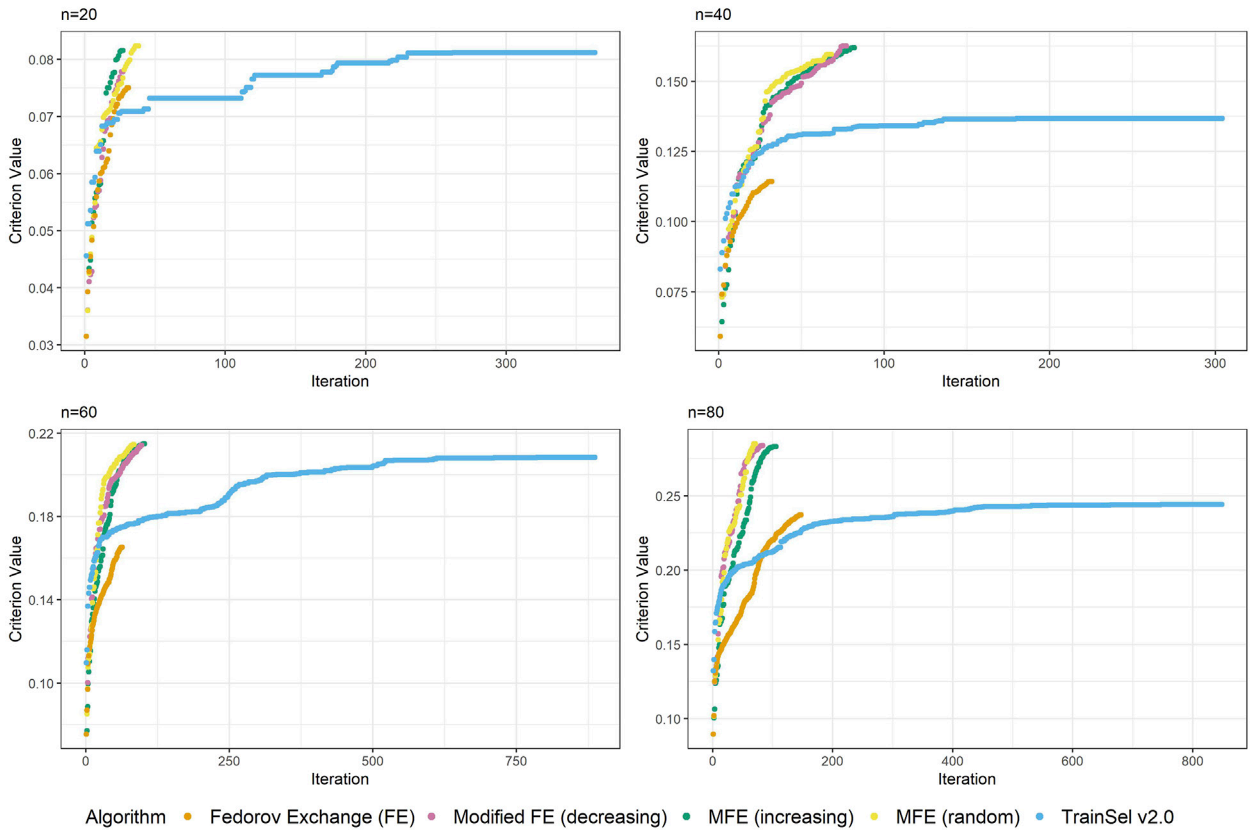

FIGURE 3
Convergence of algorithms over different sample size at one random seed (CDMin-criterion).

genotypes at 4,670 markers or as a precalculated relationship matrix of the 200 available individuals. Akdemir et al. (2021) name https://triticeaetoolbox.org as the original source of the data, which has undergone some additional preprocessing by the authors. We proceed with the GRM **G** over the 200 individuals as it is provided since the calculation is not the subject of this article. In this simulated design problem, we set the ratio of variances $\lambda = 1$ and $\mathbf{R} = \mathbf{I}_n$ for a given sample size $n$. We further include an intercept in the model, i.e., $\mathbf{X} = \mathbf{1}$.

The comparison is performed over ten restarts with different random seeds over the D-criterion in Equation 4 and the CDMin-criterion in Equation 5. Overall, the algorithms are run at sample sizes of $n \in \{10, 20, \ldots, 80\}$. We eventually compare five different search strategies w.r.t. the criterion value obtained and the runtime of the algorithms.

Since there is no natural convergence of `TrainSel`, some arbitrary parameters must be selected for comparison to the other algorithms. There are conflicting goals in comparing criterion values and runtimes between algorithms, so the parameters were chosen such that `TrainSel` was not run unnecessarily long, but would converge to an appropriate solution. In particular, the maximum number of iterations was chosen to be $30 \cdot n$, and the number of iterations without (sufficiently large) improvement before convergence was set to 100. All other parameters were set to the default values.

The results in the subsequent section refer to `Trainsel v2.0`, which is the algorithm described in the article Akdemir et al. (2021). Since the corresponding R package has since undergone major updates to version `3.0`, we reproduce the plots in this section for the newer version as well. The set of hyperparameters in the newest version has increased in comparison to `v2.0`. First, we have strived to set parameters as similarly to version `v2.0` as possible. Second, we have reproduced the design problems with the suggested hyperparameters in `TrainSel v3.0` for large high-complexity designs. Only the maximum number of iterations and the number of iterations without improvement before convergence have been adjusted as in the computations with `TrainSel v2.0`. Since results were better in the second configuration, we only discuss this case in Section 6.1.

# 6 Results

Firstly, the criterion values of different algorithms are compared to one another, secondly, runtimes are investigated and lastly, some convergence plots are examined. We start with comparisons for the D-criterion. The final criterion values and runtime per random restart can be seen in Figure 1. It is most notable that the Fedorov exchange algorithm does not perform well in terms of criterion value in small sample sizes, while it does obtain a slightly larger criterion value at $n = 80$ than the other algorithms. In general, for sample sizes $n \geq 30$, the criterion values for the `TrainSel` and (modified) Fedorov exchange algorithms are similar, while the runtime for

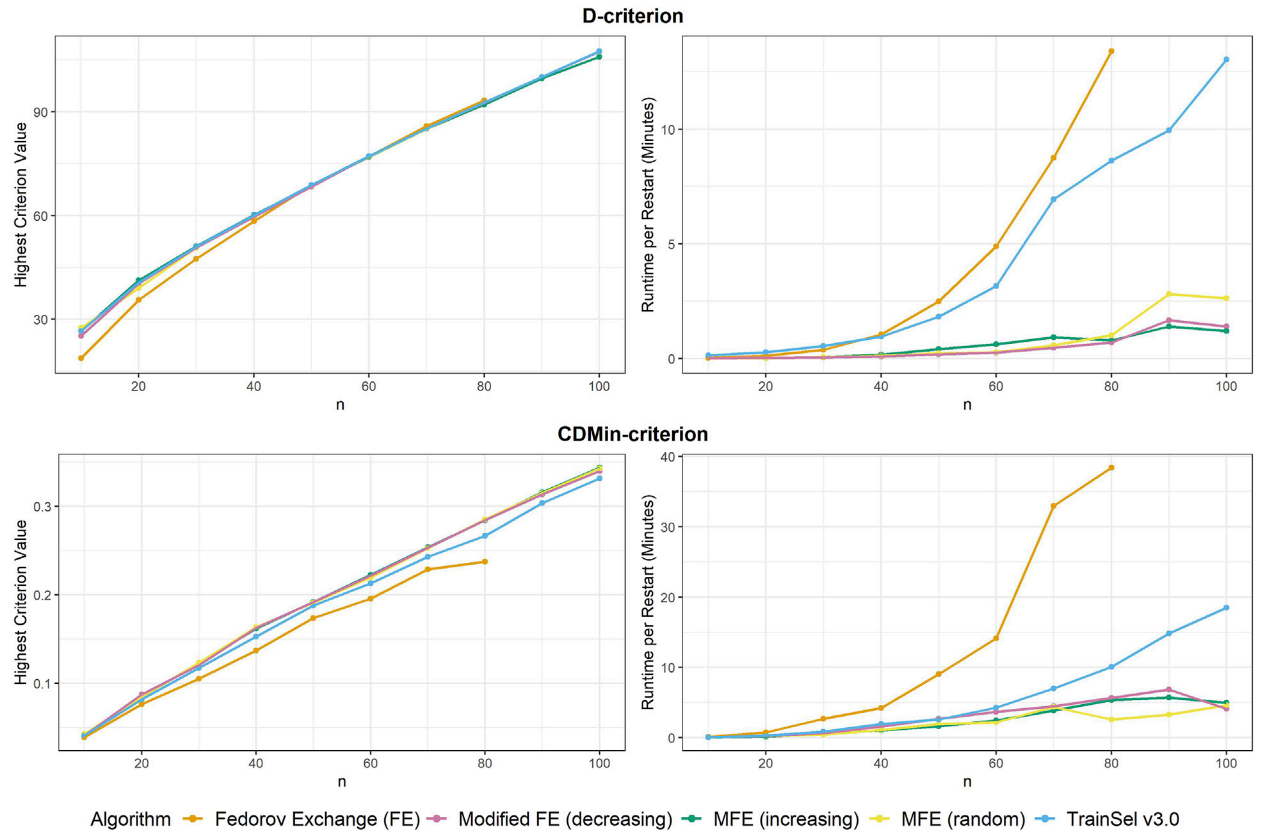

FIGURE 4
Highest criterion value and runtime per restart of different algorithms including `TrainSel v3.0`

the `TrainSel` algorithm as well as the Fedorov exchange exhibits rapid growth over increasing sample size as opposed to the modified Fedorov exchange.

Convergence was attained for all algorithms at all random restarts and sample sizes.

Although the convergence plots of different algorithms do not directly imply anything related to runtime, it is still interesting to compare the ascension to criterion values, especially w.r.t. the different orderings of the support set in the modified Fedorov exchange algorithm. Therefore, Figure 2 displays criterion values per iteration for the different algorithms.

Figure 2 depicts the criterion values over iterations for sample sizes of $n \in \{20, 40, 60, 80\}$ at one random seed. The reader should be made aware that these graphs can look different at other random seeds, in particular for small sample sizes, but we deem this plot representative for showing some general properties. For a more comprehensive understanding, some more convergence plots at all random seeds are provided in the accompanying Supplementary material for this article.

One thing that is particular about the graphs is that the modified Fedorov exchange with ordering according to increasing variance (as is traditionally used in the classical linear model) displays a very slow ascension of the criterion value. Ultimately, the resulting criterion value at convergence is similar to the other algorithms, but if the algorithm were to be stopped early, even just a random ordering of the support set might result in higher criterion values. This may seem counterintuitive, but as has been mentioned in Section 5, by removing a unit from the support set, the whole individual-specific effect of this individual disappears from the model, hence units with high variance will be interesting candidates for removal. Obviously, there is a trade-off w.r.t. relationships (i.e., covariances) to other individuals, thereby making this reasoning non-trivial.

Another interesting fact is that at this random seed, `TrainSel` ascends rather quickly, but a plateau on the criterion value is retained for several iterations before convergence. This is due to the specified settings. Other settings would have resulted in different behavior (e.g., earlier stopping). It is not straightforward to choose settings *a priori* that will result in reasonable convergence behavior.

Now moving on to the CDMin-criterion, similar plots of criterion values and runtime are provided in Figure 1.

The results in the CDMin-criterion are different compared to the D-criterion in a number of ways. Firstly, it is very clear from Figure 1 that the Fedorov exchange is not well-suited for this problem, since it results in criterion values below the ones of all other algorithms and it has bad runtime properties. The same applies to the `TrainSel` algorithm, although it is still better than Fedorov exchange w.r.t. criterion values. One may observe that the modified Fedorov exchange yields similar criterion values regardless of the ordering of the support set, but a decreasing ordering of the support set w.r.t. CD results in a longer runtime. In general, the modified Fedorov exchange requires a longer runtime for the CDMin-criterion than for the D-criterion.

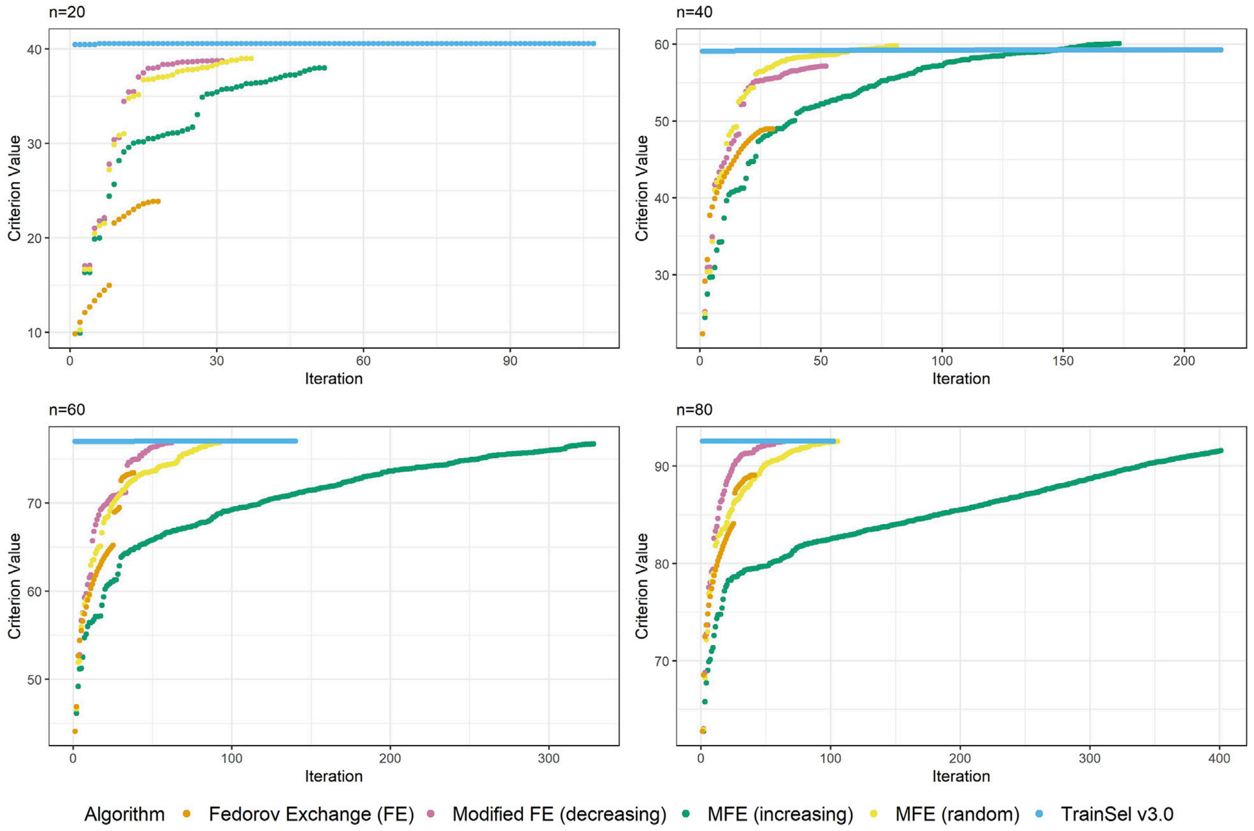

FIGURE 5
Convergence of algorithms over different sample size at one random seed (D-criterion) including `TrainSel v3.0`

Convergence occurred for all algorithms at all random seeds and sample sizes, except for `TrainSel`, where 3 out of 10 random restarts did not converge for a sample size of $n = 10$, otherwise, convergence was always attained.

Again, Figure 3 looks quite different from Figure 2, i.e, the case of the D-criterion. In particular, the `TrainSel` algorithm does not ascend as quickly and there are still large jumps in the criterion value after a considerable number of iterations. It can also be seen from the second graph in this figure that `TrainSel` can be stuck in a local minimum that is not optimal and converges too early. Interestingly, the modified Fedorov exchange with the random ordering of the support set seems to have good ascension properties. As for the D-criterion, refer to the Supplementary Material to find convergence graphs for all random seeds at different sample sizes.

## 6.1 Comparison to `TrainSel v3.0`

We have observed improvements of the optimization algorithm in `TrainSel` in the last major update. Particularly, the optimization over the D-criterion is considerably improved. We reproduce Figures 1–3 subsequently to showcase the improvements.

We have included further computations for samples sizes $n \in \{90, 100\}$ for all algorithms except for the classical Fedorov exchange since this algorithm is not competitive with the other algorithms for large sample sizes.

From Figure 4 it seems that the runtime has improved substantially, while the criterion value is still equally high as in `TrainSel v2.0` for the D-criterion. Looking at Figure 5, it is clear that `TrainSel v3.0`’s best solution is attained very quickly and the subsequent iterations are not necessary. While it is typically not possible to know this before optimization of a design problem, we can see *a posteriori* that the number of iterations until termination of the algorithm could have been chosen much smaller, which would improve runtime even further.

Seemingly, `TrainSel v3.0` performs better than the previous version on the CDMin-criterion as well, both in runtime and criterion value attained. Nonetheless, even with these improved properties, the criterion values for larger sample sizes are still short of the ones attained with the modified Fedorov exchange algorithms. Also, the runtime still increases rapidly with the sample size, i.e., complexity of the design problem.

Comparing Figures 3, 6, one can conclude that the ascension to the criterion value is quicker in the beginning of the optimization in `TrainSel v3.0`, but the problem of getting stuck in a local optimum seems to remain. As emphasized previously, the tuning parameters could be adjusted to increase computational efforts put into the optimization, but it seems unclear how much improvement this can provide in general.

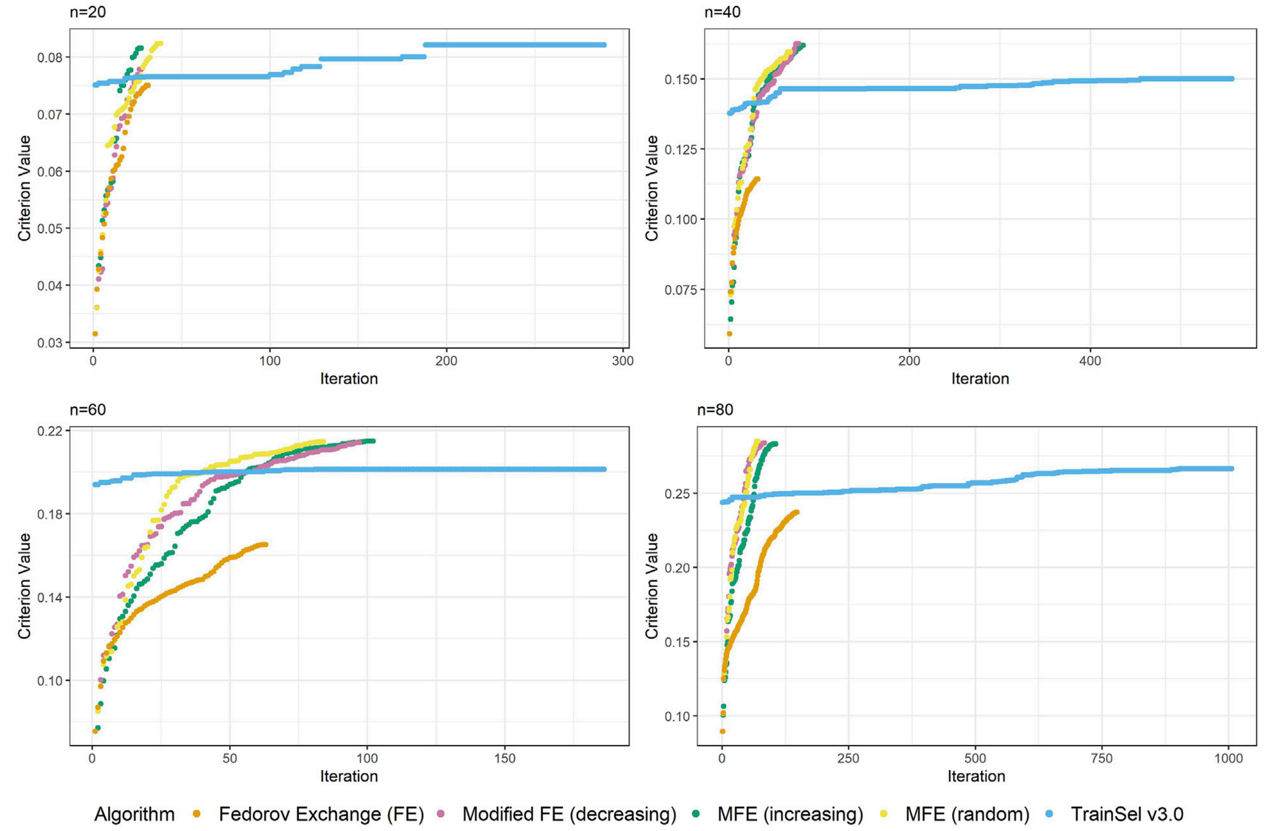

FIGURE 6
Convergence of algorithms over different sample size at one random seed (CDMin-criterion) including `TrainSel v3.0`

The reason for the performance improvement in `TrainSel` is difficult to explain. As the latest version of it is under license, it is not possible to directly access the base code to compare the changes that have been made to the optimization algorithm. The package is hosted in a GitHub project, which means that some changes to the top level documents of this package are visible between version 2.0 and 3.0. For example, the new version of `TrainSel` includes new dependencies on the R packages `foreach` and `doParallel`. As the names suggest, these packages are used for parallel computing. Even though the package `parallel` was previously used by `TrainSel v2.0`, some changes in parallel computing could have led to increased efficiency in the latest version. It is also possible that refinement to the optimization algorithm itself have led to this increased performance.

## 7 Discussion

This manuscript has showcased the application of different algorithms to a design problem in a single-trait gBLUP model in one environment. Naturally, in applications in plant breeding there may be interest in extending the model in a number of ways. For example, plant breeders are frequently intersted in performing experiments in multiple environments. This extension can be taken into account in the model, e.g., by including additional variables in the fixed effects term $\mathbf{X}\boldsymbol{\beta}$ in the model of Equation 1. Additionally, there is special interest in modeling the interaction of genomic effects and environmental effects (G x E effects). This is important information for breeders, since it helps to evaluate the quality of particular plant lines in different environments. E.g., some may be very good in one particular environment, but perform badly when exposed to a “stressful” environment, like one with prolonged droughts.

There is a large interest in these effects, a motivational article on modeling G x E effects is e.g., van Eeuwijk et al. (2016). Again, the model can be extended to include this interaction term, which would correspond to another random effect in the model of Equation 1. The additional effect could be denoted $\mathbf{Z}_2\boldsymbol{\gamma}_2$, where $\mathbf{Z}_2$ is the matrix of interactions and $\boldsymbol{\gamma}_2$ is the vector of random interaction effects. The random vector shall follow a normal distribution with mean 0 and some appropriate covariance matrix, e.g., $\Sigma_{\mathrm{GxE}} = \Sigma_{\mathrm{E}} \otimes \mathbf{G}$, where $\Sigma_{\mathrm{E}}$ could be a diagonal matrix with variances corresponding to the different environments and $\otimes$ is the Kronecker product.

Another possible extension could be the joint modeling of multiple traits that are of interest, either in one or multiple environments. For an exemplary article with stipulation and application of such a model to simulated and real data, see e.g., Montesinos-López et al. (2016).

The present manuscript was written such that the particularities of experimental design could most easily be understood. Therefore, extensions of this kind were omitted and focus was centered on the algorithmic optimization in such problems.

## 8 Conclusion

This article reviewed the optimal design problem in the gBLUP model and the main differences to the classical linear model. Some strategies for the search of exact optimal designs were discussed and put to use in a fictitious optimal experimental design problem with data from the TrainSel R package. A comparison w.r.t. criterion values and runtime was performed on the D- and CDMin-criteria over several sample sizes.

The results suggest that while TrainSel provides a lot of flexibility for users to optimize various design problems, it might be more efficient to rely on theoretical considerations and more classical algorithms in at least some applications on the gBLUP model. In particular, for the D-criterion, all algorithms resulted in similarly efficient designs for a sample size of $n \geq 40$, but a much larger runtime was necessary for TrainSel 2.0. The newest licensed version of TrainSel has improved on runtime properties and optimization results. In general, for increasing sample size, TrainSel's runtime grows much quicker than the ones for the modified Fedorov exchange.

It is quite interesting to note that the ordering of the support set in the modified Fedorov exchange according to decreasing variance seems to be beneficial and leads to quick ascension, as opposed to the traditional ordering by increasing variance.

For the CDMin-criterion (the default criterion in TrainSel), the Fedorov exchange is not suitable, since it provides smaller criterion values and has bad runtime properties. The TrainSel algorithm results in smaller criterion values as well and runtime increases rapidly and is much larger than for the modified Fedorov exchange. In the modified Fedorov exchange, it seems that all sortings of the support set provide similarly efficient designs, but sorting by decreasing CD still takes slightly longer. It is interesting to see that even random sorting of the support set for exchange seems to work quite well.

Overall it was demonstrated that classical exchange-type algorithms can typically obtain similarly efficient designs as TrainSel while substantially saving computational runtime. We expect this advantage to scale with respect to the increasing complexity of the respective problem.

## Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: The data underlying this article are available in the R package TrainSel (see Akdemir et al., 2021). The authors name https://triticeaetoolbox.org as the original source of the data, which has undergone some additional preprocessing.

## Author contributions

AS: Formal Analysis, Investigation, Methodology, Software, Writing–original draft, Writing–review and editing. WM: Conceptualization, Formal Analysis, Methodology, Supervision, Writing–review and editing. AF: Supervision, Validation, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This publication is supported by Johannes Kepler University Open Access Publishing Fund and the Federal State Upper Austria.

## Acknowledgments

We are grateful to the referees for their remarks, which led to a considerable improvement of the paper.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2024.1462855/full#supplementary-material

## References

Akdemir, D., Rio, S., and Isidro y Sánchez, J. (2021). TrainSel: an R package for selection of training populations. *Front. Genet.* 12, 655287. doi:10.3389/fgene.2021.655287

Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007). *Optimum experimental designs, with SAS*. Oxford University Press.

Berro, I., Lado, B., Nalin, R. S., Quincke, M., and Gutiérrez, L. (2019). Training population optimization for genomic selection. *Plant Genome* 12, 1–14. doi:10.3835/plantgenome2019.04.0028

Butler, D., Cullis, B., and Taylor, J. (2021). Extensions in linear mixed models and design of experiments.

Clark, S. A., and van der Werf, J. (2013). Genomic best linear unbiased prediction (gBLUP) for the estimation of genomic breeding values. *Genome-wide Assoc. Stud. genomic Predict.* 1019, 321–330. doi:10.1007/978-1-62703-447-0_13

Cook, R. D., and Nachtsheim, C. J. (1980). A comparison of algorithms for constructing exact D-Optimal designs. *Technometrics* 22, 315. doi:10.2307/1268315

Fedorov, V. (1972). *Theory of optimal experiments designs*. New York: Academic Press.

Fedorov, V. V., and Leonov, S. L. (2013). *Optimal design for nonlinear response models*. Boca Raton, FL: CRC Press.

Fernández-González, J., Akdemir, D., and Sánchez, J. I. Y. (2023). A comparison of methods for training population optimization in genomic selection. *Theor. Appl. Genet.* 136, 30. doi:10.1007/s00122-023-04265-6

Habier, D., Fernando, R. L., and Dekkers, J. C. M. (2007). The impact of genetic relationship information on genome-assisted breeding values. *Genetics* 177, 2389–2397. doi:10.1534/genetics.107.081190

Henderson, C. (1975). Best linear unbiased estimation and prediction under a selection model. *Biometrics* 31, 423–447. doi:10.2307/2529430

Hickey, J. M., Chiurugwi, T., Mackay, I., Powell, W., and Implementing Genomic Selection in CGIAR Breeding Programs Workshop Participants (2017). Genomic prediction unifies animal and plant breeding programs to form platforms for biological discovery. *Nat. Genet. Print.* 49, 1297–1303. doi:10.1038/ng.3920

Isidro y Sanchéz, J., and Akdemir, D. (2021). Training set optimization for sparse phenotyping in genomic selection: a conceptual overview. *Front. Plant Sci.* 12, 715910. doi:10.3389/fpls.2021.715910

Jannink, J., Lorenz, A. J., and Iwata, H. (2010). Genomic selection in plant breeding: from theory to practice. *Briefings Funct. Genomics* 9, 166–177. doi:10.1093/bfgp/elq001

Javier, F. G. (2024). TrainSel v3.0. Available at: https://github.com/TheRocinante-lab/TrainSel (Accessed June 10, 2024).

Laloë, D. (1993). Precision and information in linear models of genetic evaluation. *Genet. Sel. Evol.* 25, 557–576. doi:10.1186/1297-9686-25-6-557

Montesinos-López, O. A., Montesinos-López, A., Crossa, J., Toledo, F. H., Pérez-Hernández, O., Eskridge, K. M., et al. (2016). A genomic Bayesian multi-trait and multi-environment model. *G3 Genes Genomes Genet.* 6, 2725–2744. doi:10.1534/g3.116.032359

Prus, M., and Filová, L. (2024). Computational aspects of experimental designs in multiple-group mixed models. *Stat. Pap.* 65, 865–886. doi:10.1007/s00362-023-01416-1

Rincent, R., Laloë, D., Nicolas, S., Altmann, T., Brunel, D., Revilla, P., et al. (2012). Maximizing the reliability of genomic selection by optimizing the calibration set of reference individuals: comparison of methods in two diverse groups of maize inbreds (Zea mays L.). *Genetics* 192, 715–728. doi:10.1534/genetics.112.141473

Robinson, G. K. (1991). That BLUP is a good thing: the estimation of random effects. *Stat. Sci.* 6, 15–32. doi:10.1214/ss/1177011926

van Eeuwijk, F. A., Bustos-Korts, D. V., and Malosetti, M. (2016). What should students in plant breeding know about the statistical aspects of genotype × environment interactions? *Crop Sci.* 56, 2119–2140. doi:10.2135/cropsci2015.06.0375

VanRaden, P. M. (2008). Efficient methods to compute genomic predictions. *J. Dairy Sci.* 91, 4414–4423. doi:10.3168/jds.2007-0980