ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign

Di Wu[1]
Faculty of Engineering and Natural Sciences,
Tampere University,
Tampere 33101, Finland
e-mail: dwa88@sfu.ca

Dmitry Sotnikov
Faculty of Information Technology and
Communication Sciences,
Tampere University,
Tampere 33101, Finland
e-mail: dmitry.sotnikov@psi.ch

G. Gary Wang
Product Design and Optimization Laboratory
(PDOL),
Simon Fraser University,
Surrey, BC V3T 0A3, Canada
e-mail: gary_wang@sfu.ca

Eric Coatanea
Faculty of Engineering and Natural Sciences,
Tampere University,
Tampere 33101, Finland
e-mail: eric.coatanea@tuni.fi

Mika Lyly
Faculty of Information Technology and
Communication Sciences,
Tampere University,
Tampere 33101, Finland
e-mail: mika.lyly@tuni.fi

Tiina Salmi
Faculty of Information Technology and
Communication Sciences,
Tampere University,
Tampere 33101, Finland
e-mail: tiina.salmi@tuni.fi

A Dimension Selection-Based Constrained Multi-Objective Optimization Algorithm Using a Combination of Artificial Intelligence Methods

The computational cost of modern simulation-based optimization tends to be prohibitive in practice. Complex design problems often involve expensive constraints evaluated through finite element analysis or other computationally intensive procedures. To speed up the optimization process and deal with expensive constraints, a new dimension selection-based constrained multi-objective optimization (MOO) algorithm is developed combining least absolute shrinkage and selection operator (LASSO) regression, artificial neural networks, and grey wolf optimizer, named L-ANN-GWO. Instead of considering all variables at each iteration during the optimization, the proposed algorithm only adaptively retains the variables that are highly influential on the objectives. The unselected variables are adjusted to satisfy the constraints through a local search. With numerical benchmark problems and a simulation-based engineering design problem, L-ANN-GWO outperforms state-of-the-art constrained MOO algorithms. The method is then applied to solve a highly complex optimization problem, the design of a high-temperature superconducting magnet. The optimal solution shows significant improvement as compared to the baseline design.
[DOI: 10.1115/1.4062548]

Keywords: multi-objective optimization, expensive constraints, LASSO, ANN, grey wolf optimizer, high-temperature superconducting magnet, AI, approximation-based optimal design, design optimization, machine learning, metamodeling, multi-objective optimization

1 Introduction

Two fundamental challenges exist for a complex simulation-based optimization problem: (1) long simulation time and (2) expensive constraints. Finite element analysis (FEA) and other numerical simulation methods are often employed to predict the performance of a complex system, where a single simulation can take hours or days to complete. Since optimization, especially multi-objective optimization (MOO), usually needs to call the simulation iteratively to search for optima, the computational cost for optimization often becomes prohibitive. On the other hand, constraints involved in complex engineering problems are usually obtained together with objective values as a result of simulations. Therefore, how to deal with expensive constraints becomes a question in improving the optimization efficiency for complex problems.

By reducing the number of expensive simulation calls during the optimization of such types of problems, surrogate models are generally employed. The common way to use a surrogate model in optimization is to construct and validate them before optimization. The structure of the model remains stable during optimization [1]. In the cases studied in this article, the accuracy of the surrogate model significantly influences the optimization results. If the approximation accuracy is low, the Pareto frontier obtained from the surrogate model will not be acceptable. Therefore, the state-of-art surrogate model-based optimization algorithms usually update the surrogate model adaptively during the optimization iterations to improve the accuracy in specific areas of interest. This is the case for the Pareto set pursuing method [2,3]. Employing a kernel

[1] Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received January 4, 2023; final manuscript received March 9, 2023; published online June 9, 2023. Assoc. Editor: Ikjin Lee.

function and a prior distribution, Bayesian optimization (BO) [4] focuses on using acquisition functions to guide new sample generation. In multi-objective BO, two kinds of operations are applied usually: one is to convert the multi-objective problems to single objective [5–8], and the other is to employ Pareto frontier-related acquisition functions, such as expected hypervolume improvement [9–11] and predicted entropy search [12]. In addition, adaptive surrogate model-assisted evolutionary algorithms can be employed as a design space exploration strategy, where surrogate models are updated when new samples are generated [13–15].

To further improve the efficiency of the optimization algorithm, the idea of focusing on only a subset of the dimensions at each iteration is proposed in single-objective optimization algorithms. In dynamic coordinate search (DYCORS) [16], the dimensions are selected randomly and only the selected dimensions are perturbated during the iteration. A more aggressive dimension selection method, based on the sensitivity of the variables, has been developed in methods such as mode-pursuing sampling using discriminative coordinate perturbation [17] and partial metamodel optimization [18].

With the development of artificial intelligence (AI), more and more AI-based MOO algorithms are proposed. Machine learning methods, such as extreme learning machines [19] and convolutional neural networks [20], are employed as surrogate models to deal with MOO problems. However, the networks used in both studies are static and a large number of initial samples are needed to ensure the accuracy of the surrogate. Additionally, compared to other surrogate models, such as radial basis function (RBF), deep neural networks usually need much more samples to train the model. On the other hand, a k-means clustering method is used in a large-scale MOO algorithm to divide the design variables into two groups [21]. Since the cluster needs samples by perturbing each design variable, extra computational costs are required before optimization.

How to deal with expensive constraints is another research direction. Constraint handling strategies can be classified into four categories: using a penalty function, separating objectives and constraints, converting constraints into an extra objective, or hybrid methods [22]. Different penalty types are utilized in the category of using penalty functions. The main issue with using a penalty function is the difficulty to determine the penalty factors. The adaptive penalty factor is the most attractive in this category [23]. In the category of separating objectives and constraint methods, one of the common approaches is considering constraints in the dominance principles, such as constraint dominance principle [24] and epsilon constraint handling [25]. Other constraint handling approaches are developed recently. In the push and pull search framework [26], the optimization process is divided into the pull and push stages. Only the objectives are considered in the pull stage, and all objectives and constraints are involved in the pushing stage. In Ref. [27], a coevolutionary framework is constructed by optimizing two separated groups of the population concerning two problems: one is the original problem with constraints and the other is the original problem without constraints. During optimization, the two population groups are connected to share information to converge to the feasible Pareto solutions. The method of converting constraints into an extra objective is proposed in Ref. [28], where the degree of constraint violation is calculated and treated as an extra objective to be optimized to find feasible solutions located at the constraint boundaries. But in MOO, the Pareto solutions do not always locate on the boundaries. Therefore, hybrid methods are developed, such as infeasibility-driven evolutionary algorithms (IDEA) [29]. In IDEA, two groups of the population are generated: one is solved for the original constrained MOO problem and the other smaller group is solved by converting the constraint violation as an extra objective to focus on the boundaries of the constraints. However, the number of function evaluations for this kind of method is usually higher than for other kinds of methods.

Meanwhile, surrogate models are employed to handle expensive constraints to reduce the computational cost. One way is to employ surrogate models in the constrained MOO algorithms. For example, a surrogate model-based evolution strategy uses the prediction values from RBF models of objectives and constraints to estimate the constraint dominance scores for each offspring in a framework of non dominated sorting genetic algorithm-II (NSGA-II) [30]. By directly replacing expensive simulations with surrogate models, it becomes harder to find feasible solutions when the accuracy of the surrogate model is low. Another way is to adjust the surrogate model-based constraint handling strategies in single-objective optimization to solve multi-objective problems. In Refs. [31,32], a surrogate-based constraint handling process, named constrained optimization by radial basis function approximation [33], is employed to deal with the expensive constraints by optimizing a problem based on surrogate models. In both cases, an optimization problem based on surrogate models is solved at each iteration to produce a new sample for a MOO problem. As a result, the extra optimization process will increase the complexity when dealing with high-dimensional problems. In multi-objective BO, one way to deal with constraints is to add constraints to the acquisition function to solve a constrained suboptimization problem to find feasible new samples [34]. Another way is to include constraint satisfaction in the acquisition function [9]. However, multi-objective BO needs to solve a suboptimization problem on the acquisition function, and it is an inefficient approach when high-dimensional problems are considered.

In this article, we focus on the strategy of separating the objectives and constraints to avoid involving unnecessary complexity in the original problem. To inherit the high efficiency of a partial model and fully utilize each dimension to deal with constrained optimization problems, an AI technique-based dimension selection strategy and related constraints handling processes are developed in this article. A novel sensitivity-based dimension selection method using the least absolute shrinkage and selection operator (LASSO) [35] regression is proposed to separate the variables into two subsets, one for objectives and one for constraints. The subset of variables related to objectives is updated by a grey wolf optimizer (GWO) only considering the objectives, while the other subset of variables is adjusted to find feasible solutions through a random local search process. To reduce the number of expensive simulations evaluated during optimization, artificial neural networks (ANNs) for objectives and constraints are trained based on the existing samples and updated when new samples are generated along with optimization.

Superconducting magnetic field optimization is an excellent example of a complex multi-objective optimization problem with expensive constraints. Superconducting magnet has highly nonlinear material properties. FEA simulations are needed to obtain the magnet field properties. However, few studies have been developed using optimization algorithms to deal with superconducting magnetic field optimization [23,24]. In this article, the proposed MOO algorithm is employed to solve a superconducting magnetic field design problem to minimize the amount of required superconductor material and to obtain the largest magnetic field under the requirements of the field qualities.

This article is organized as follows. Section 2 details the proposed method. To validate the performance of the proposed algorithm, comparisons with ANN-GWO and well-known state-of-the-art optimization algorithms are performed in Sec. 3. Then, the proposed method is applied to solve solenoid magnetic design problems using Cu tapes and second-generation high-temperature superconductor (2G HTS) tapes in Sec. 4. Section 5 concludes the article by summarizing the properties of the algorithm.

2 Proposed Optimization Algorithm

The proposed constrained multi-objective optimization algorithm is introduced in this section. At the beginning of each iteration, dimensions are divided into two subsets, one subset is for objectives and the other is for constraints. The GWO updating mechanism updates the values of the samples focusing on the subset of dimensions for objectives. Constraint handling operations are performed

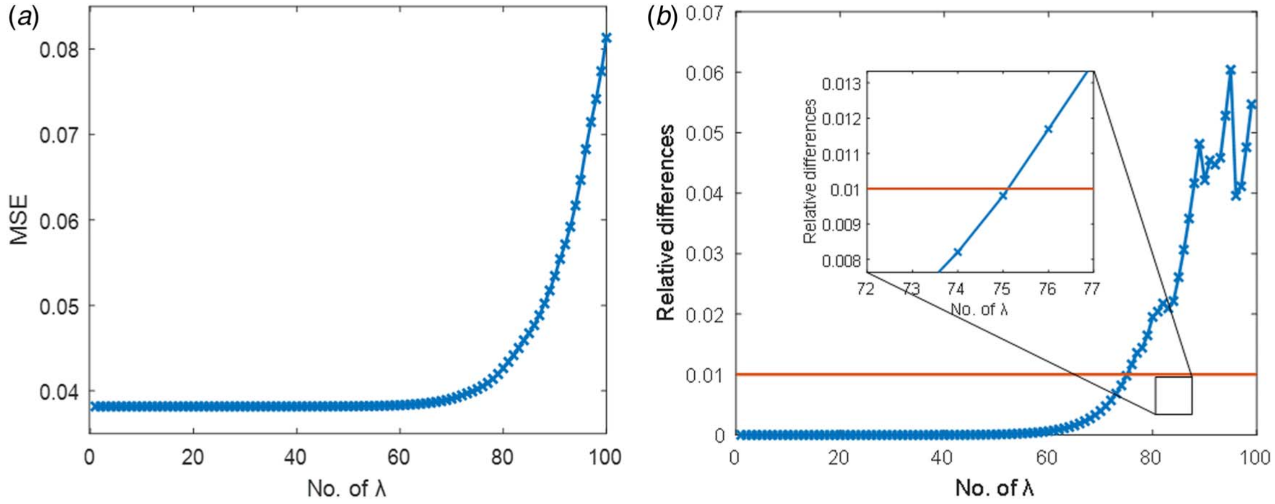

Fig. 1 MSE and relative MSE differences plots in LASSO regression: (*a*) MSE versus the number of λ values and (*b*) relative differences of MSE values for two adjacent λ values

on the subset of dimensions for constraints to find feasible solutions afterward. To utilize the expensive simulation efficiently, only the candidates selected based on the prediction values from ANNs are evaluated by the simulation. In this section, the components of the algorithm are introduced first and the overall flowchart of the algorithm is summarized at the end.

2.1 Dimension Selection Strategy Using LASSO. LASSO is a variable selection method derived from the ordinary least square (OLS) regression method [35]. LASSO is a method of choice for cases where several potential features or variables are most influential. LASSO is not employed alone in our approach but in conjunction with an L2 regularization approach such as OLS methods or ANN, for example. As an L1 regularization method, LASSO requires a few numbers of samples to achieve a relatively accurate prediction. Additionally, since the goal of using LASSO in this article is for variable selection rather than for function prediction, using nonlinear selection methods such as random forest or autoencoder methods would bring additional computing costs. In addition, those nonlinear approaches demand large training data sets and thus are not compatible with the goal of optimization using the least number of samples in this work. Additionally, the process is iterative, and the selection is also updated with the iterations. Thus, LASSO is selected to realize the dimension selection strategy. The details are presented as follows.

Given a linear regression model $\hat{Y} = w_0 + \boldsymbol{X}^T\boldsymbol{w}$, where $\boldsymbol{X}$ represents the position, $\hat{Y}$ is the predicted response from the regression model, and w_0 and $\boldsymbol{w}$ are the coefficients for the linear model. The LASSO regression is to find the value w_0 and $\boldsymbol{w}$ to solve the minimization problem as shown in Eq. (1).

$$\min_{w_0,\, \boldsymbol{w}} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{1}$$

where n is the number of positions, λ is a nonnegative parameter, and p is the number of the weights $\boldsymbol{w}$. For a given λ, due to the property of the L1 regression function (i.e., the second term in Eq. (1)), some elements in w_0 and $\boldsymbol{w}$ will be zero. By gradually increasing the value of λ, more coefficients become zero, which means fewer variables contribute to the linear model. However, the prediction accuracy of the linear model will become worse with the decreasing number of selected variables. Thus, the question is, what value of λ is to be selected to determine the regression model? In this article, to find the optimal set of selected variables, an accuracy criterion is proposed to select the value of λ. The largest λ value (λmax) that gives a nonnull model is calculated. Then, a vector of a geometric sequence of λ values is generated in the range [10^{-4}*λ max, λ max]. For each λ value, the problem in Eq. (1) is solved to obtain a regression model and the regression error of the model is estimated. The regression errors of the models (i.e., mean square error (MSE)) with two adjacent λ values are compared. If the relative difference between two regression errors is larger than a threshold (e.g., 10^{-2}), it means an important variable is missing in the model with the larger λ value. Thus, the regression model using the smaller λ value is selected, and the variables whose coefficient is not zero are selected as the important variables. Note that the important variables are those that have a large impact on the objectives. According to the selection result, the set of variables can be divided into two subsets: one includes the selected variables from LASSO regression that have a large impact on the objectives and the other includes the rest of the variables that are used in constraint handling. In this article, $\boldsymbol{X}_{\boldsymbol{var\_obj}}$ is used to represent the variables selected in the objective subset, while $\boldsymbol{X}_{\boldsymbol{var\_con}}$ represents the variables in the constraint subset.

An example of how the dimension selection strategy works is presented as follows. In this case, the first objective of the CF6 problem (equations in the Appendix) is approximated using LASSO regression. First, λmax is determined as 0.1519 since when λ is larger than the value, all the weights (w_0 and $\boldsymbol{w}$) in Eq. (1) shrink to zero. Then, a geometric sequence vector of λ is generated in the range [1.5e-5, 0.1519], where 100 λs are generated. For each λ value, Eq. (1) is solved to obtain the regression model. The prediction errors presented by MSE are plotted in Fig. 1(*a*), while the relative differences of MSE values for two adjacent λ values are plotted in Fig. 1(*b*). As the threshold is set to 0.01, the 75th λ is selected, where $\lambda = 0.0148$. For the given λ for the CF6 problem, the weight for the ninth variable ($\boldsymbol{X}_9$) is zero. Therefore, $\boldsymbol{X}_{\boldsymbol{var\_obj}}$ is [$\boldsymbol{X}_1$, $\boldsymbol{X}_2$, …, $\boldsymbol{X}_8$, $\boldsymbol{X}_{10}$] and $\boldsymbol{X}_{\boldsymbol{var\_con}}$ is $\boldsymbol{X}_9$.

Since LASSO is a linear regression method but most of the simulations are nonlinear, a log10 transformation is performed on the positions ($\boldsymbol{X}$) and their objective values ($\boldsymbol{Y}$) to create nonlinearity. Then, the regression problem in Eq. (1) is modified to Eq. (2).

$$\min_{w_0\ \boldsymbol{w}} \sum_{i=1}^{n} (\log_{10} Y_i - \log_{10} w_0 + (\log_{10} \boldsymbol{X_i})^T \boldsymbol{w})^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{2}$$

An adaptive strategy is used in the proposed algorithm in that the LASSO regression model is updated at each iteration. In the early iterations, few samples have been generated. Some variables are not selected as the objective subset, which means more variables can be used in constraint handling (details in Sec. 3.1) to find feasible solutions. Additionally, the uncertainty caused by the lack of samples may lead to the selection varying at each iteration, which increases the explorative ability of the algorithm. On the other hand, by updating the LASSO regression with new samples, errors in selections can be corrected. As a result, the optimization can focus on the real important variables at the latter iterations for exploitation. More details are presented in Sec. 3.1.

2.2 Artificial Neural Network Approximation.

Multilayer neural networks [36] are employed to approximate the objectives and constraints to reduce the number of expensive simulations in the optimization. Compared to conventional surrogates, such as RBF and Kriging, ANN is a regression model that does not always go through every sample point. The partial model with the selected variables can be regarded as a sparse dataset with zero values for the nonselected variables. In Ref. [37], it is reported that ANN performs better in sparse data regression problems compared to the Kriging model. The usage of ANNs in this article is to judge which newly generated sample is not dominated by others at the current iteration. New ANNs are constructed at each iteration and only used at the iteration. For this purpose, a multilayer ANN satisfies the requirement. Thus, ANNs with two hidden layers and four hidden neurons for each layer are selected to construct for each objective and constraint, respectively. The nonlinear tansigmoid function is used as the activation function in all ANNs. For the objective functions, only the variables selected from the LASSO regression are used as the input nodes. At each iteration, the number of inputs for the objective ANNs is updated and the ANNs are re-trained with all the existing samples. On the other hand, all the variables are used to construct the ANNs for constraints, and the ANNs will also be updated for all samples.

2.3 GWO Updating Mechanism.

The GWO updating mechanism [38], as shown in Eq. (3), is employed to update the position of the samples to generate new candidates at each iteration,

$$
\begin{aligned}
&\boldsymbol{x}^{new}_{\boldsymbol{var\_obj}} = \frac{\boldsymbol{x}_{1,\boldsymbol{var\_obj}} + \boldsymbol{x}_{2,\boldsymbol{var\_obj}} + \boldsymbol{x}_{3,\boldsymbol{var\_obj}}}{3}\\
&\boldsymbol{x}_{1,\boldsymbol{var\_obj}} = \boldsymbol{x}_{\alpha,\boldsymbol{var\_obj}} - A_1 \cdot D_\alpha\\
&\boldsymbol{x}_{2,\boldsymbol{var\_obj}} = \boldsymbol{x}_{\beta,\boldsymbol{var\_obj}} - A_2 \cdot D_\beta\\
&\boldsymbol{x}_{3,\boldsymbol{var\_obj}} = \boldsymbol{x}_{\delta,\boldsymbol{var\_obj}} - A_3 \cdot D_\delta\\
&D_\alpha = |C_1 \cdot \boldsymbol{x}_{\alpha,\boldsymbol{var\_obj}} - \boldsymbol{x}_{\boldsymbol{var\_obj}}|\\
&D_\beta = |C_1 \cdot \boldsymbol{x}_{\beta,\boldsymbol{var\_obj}} - \boldsymbol{x}_{\boldsymbol{var\_obj}}|\\
&D_\delta = |C_3 \cdot \boldsymbol{x}_{\delta,\boldsymbol{var\_obj}} - \boldsymbol{x}_{\boldsymbol{var\_obj}}|\\
&A_i = 2ar_{1,i} - a, \quad i = 1, 2, 3\\
&C_i = 2 \cdot r_{2,i}, \quad i = 1, 2, 3\\
&a = \frac{2(\text{maxNFE} - \text{NFE})}{\text{maxNFE}}
\end{aligned}
\tag{3}
$$

where r_1 and r_2 are random numbers generated in the range [0, 1]. To speed up the updating process, the dimensions that have a large influence on the objective values participate in the updating mechanisms. The nonselected dimensions remain at the original value and are updated in the constraint handling process to find a feasible solution. The parameter a is used to balance the exploitation and exploration in the GWO updating process, which is controlled by the current number of function evaluations and the maximal function evaluations (maxNFE).

$\boldsymbol{x}_\alpha$, $\boldsymbol{x}_\beta$, and $\boldsymbol{x}_\delta$ represent the three leaders of the wolves (i.e., the three best-performing samples) to guide the position updating process, which are selected following the criteria in the original multi-objective GWO algorithm [39]. The entire design space is divided into multiple subspaces, and the subspace with a smaller number of nondominated solutions has a large chance to be selected. In this case, the proposed algorithm employs the GWO updating mechanism to ensure the spread of the Pareto set.

2.4 Constraint Handling Operations.

The constraint handling operations are performed at the initial sampling step, the potential nondominated solution determination step, and the current Pareto set determination step. Note that a nondominated solution is used among the newly generated samples at one iteration to represent the sample that is not dominated by other newly generated samples. On the other hand, if one sample is not dominated by all the other existing samples, it is called a Pareto solution. At initial sampling, the goal is to generate an initial Pareto set. The second handling operation is the key part that adjusts the values for the subset of variables to satisfy constraints. This operation is performed after updating the positions of the samples using GWO. The last operation is a checking process to ensure all the solutions in the Pareto set are feasible. The details are introduced as follows.

Constraint operation 1: *Initial Pareto set determination considering the feasibility*

Initial samples are generated randomly in the design space and their feasibility is checked to determine the initial Pareto set. As per the requirements of the GWO updating mechanism, there is at least one solution in the current Pareto set. Thus, when there are no feasible solutions found from the initial sampling, a constraint operation is performed to find a potential Pareto solution for GWO. If one solution is the optimum for one objective, it could be a Pareto solution. Following this logic, an ANN is constructed for one objective, and a single-objective local search is performed on the ANN to obtain the potential Pareto solution. To be clarified, the ANN constructed in *constraint operation 1* is only used in the initial sampling step. ANNs are constructed for each objective at the following optimization steps.

Step 1.1: Start point determination

- Calculate the maximal constraint violation values for each sample.
- Select the sample with the smallest violation value as the starting point, $\boldsymbol{x}_0$.

Step 1.2: ANN construction

- Construct ANN_obj($\boldsymbol{X}$) based on a randomly selected objective.
- Construct the ANN_con($\boldsymbol{X}$) for all constraints.

Step 1.3: Local search

- Use sequential quadratic programming (SQP) to perform a local search on ANN_obj and ANN_con. Obtain the solution as $\boldsymbol{x}_{fe}$.
- Evaluate $\boldsymbol{x}_{fe}$ using simulations to obtain the objectives response ($\boldsymbol{f}_{fe}$) and constraints response ($\boldsymbol{g}_{fe}$).
- Create the Pareto set, $[\boldsymbol{X}_P, \boldsymbol{F}_P, \boldsymbol{G}_p] \leftarrow [\boldsymbol{x}_{fe}, \boldsymbol{f}_{fe}, \boldsymbol{g}_{fe}]$.
- Store the new sample $[\boldsymbol{x}_{fe}, \boldsymbol{f}_{fe}, \boldsymbol{g}_{fe}]$ into the sample base.

Constraint operation 2: Local search considering feasibility

After updating the dimensions related to objectives, the new positions of the samples can be presented as $[\boldsymbol{X}^{new}_{\boldsymbol{var\_obj}}, \boldsymbol{X}_{\boldsymbol{var\_con}}]$. For each sample $\boldsymbol{x}^{new} = [\boldsymbol{x}^{new}_{\boldsymbol{var\_obj}}, \boldsymbol{x}_{\boldsymbol{var\_con}}]$, the following constraint operation is performed.

Step 2.1: Feasibility checking

- Estimate the constraint values for the sample using ANNs, $\hat{\boldsymbol{g}} = \text{ANN\_con}(\boldsymbol{x}^{new})$.
- Check the feasibility of the sample. If the sample is feasible, return $\boldsymbol{x}^{new,c} = [\boldsymbol{x}^{new}_{\boldsymbol{var\_obj}}, \boldsymbol{x}_{\boldsymbol{var\_con}}]$, else go to *Step 2.2*.

Step 2.2: Local search

- Randomly change the values of the dimensions for constraints, $\boldsymbol{x}^{new}_{\boldsymbol{var\_con}}$.
- Check the feasibility of the new sample, $\boldsymbol{x}^{new,c} = [\boldsymbol{x}^{new}_{\boldsymbol{var\_obj}}, \boldsymbol{x}^{new}_{\boldsymbol{var\_con}}]$ by the predicted values from ANNs.
- If the sample is feasible, return the new sample $\boldsymbol{x}^{new,c}$; otherwise, repeat this step until a feasible solution is found or the maximum repetition number is reached. In this case, the maximum repetition number is set to 20.

Note that no feasible solution may be found at this step. In this case, one sample will be randomly selected from the set of samples to be evaluated by the simulation.

Constraint operation 3. This operation is performed at the end of each iteration when determining the current Pareto set. As infeasible solutions may exist from the initial samples, the feasibility of the solutions in the current Pareto set is checked. When the number of current solutions is larger than one and the infeasible solution exists, the infeasible solution is removed from the current set; otherwise, the infeasible solution is the only solution in the current set, and it will be kept until new solutions are found.

2.5 Overall Algorithm.

2.5 Overall Algorithm. Figure 2 shows the flowchart of the L-ANN-GWO algorithm. The details of each step are listed as follows.

Step 1: Initial random sampling:
- Generate N uniform random points $\boldsymbol{X}_{wolf}$, where N is the number of wolves.
- Evaluate samples by the simulation, $[\boldsymbol{F}_{wolf}, \boldsymbol{G}_{wolf}] \leftarrow$ Simulation ($\boldsymbol{X}_{wolf}$).

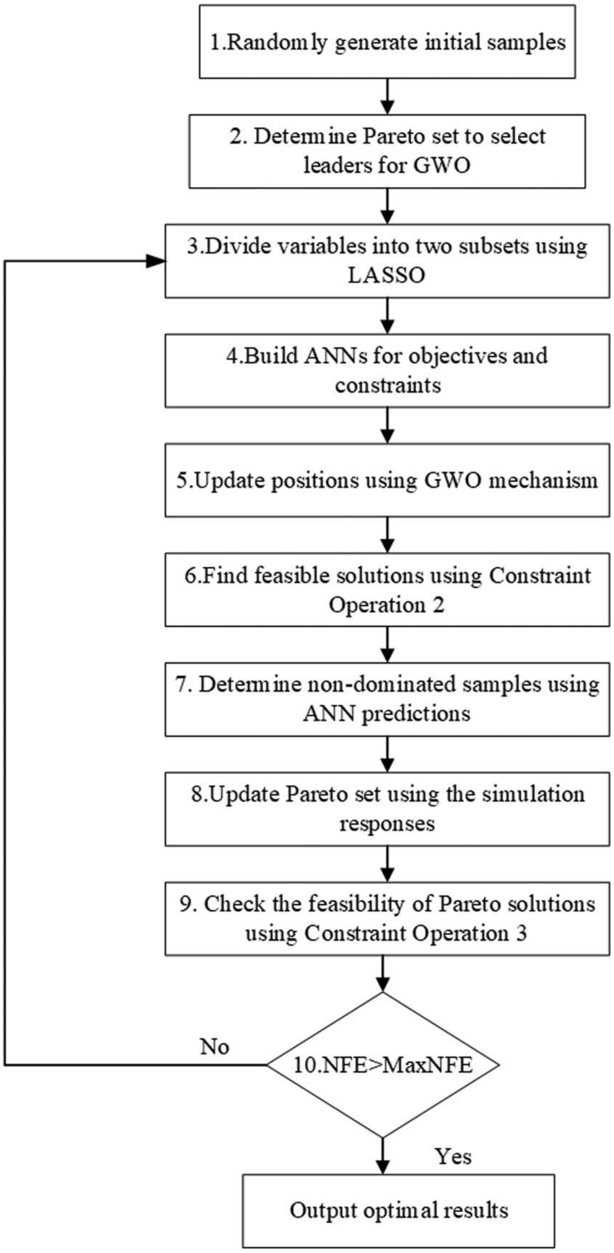
1.Randomly generate initial samples
2. Determine Pareto set to select leaders for GWO
3.Divide variables into two subsets using LASSO
4.Build ANNs for objectives and constraints
5.Update positions using GWO mechanism
6.Find feasible solutions using Constraint Operation 2
7. Determine non-dominated samples using ANN predictions
8.Update Pareto set using the simulation responses
9. Check the feasibility of Pareto solutions using Constraint Operation 3
No
10.NFE>MaxNFE
Yes
Output optimal results

Fig. 2 Flowchart of L-ANN-GWO

- Store initial samples in the sample base, $[\boldsymbol{X}, \boldsymbol{F}, \boldsymbol{G}] \leftarrow [\boldsymbol{X}_{wolf}, \boldsymbol{F}_{wolf}, \boldsymbol{G}_{wolf}]$.

Step 2: Pareto set determination
- If a feasible solution exists, determine the current Pareto set, $[\boldsymbol{X}_P, \boldsymbol{F}_P, \boldsymbol{G}_p]$.
- Else, perform constraint handling operation 1 to produce the Pareto set.

Step 3: Dimension selection
- Use LASSO to divide the variables into two subsets: one for objectives, $\boldsymbol{X}_{\boldsymbol{var\_obj}}$, and the other for constraints, $\boldsymbol{X}_{\boldsymbol{var\_con}}$.

Step 4: ANN approximation
- Train ANN_obj with respect to $\boldsymbol{X}_{\boldsymbol{var\_obj}}$.
- Train ANN_con with respect to $\boldsymbol{X}$.

Step 5: Position updating
- Update the position of the selected variables via Eq. (3), $\boldsymbol{X}^{\boldsymbol{new}}_{wolf,\boldsymbol{var\_obj}} \leftarrow \mathrm{GWO}(\boldsymbol{X}_{wolf,\,\boldsymbol{var\_obj}},\ \boldsymbol{X}_P)$.

Step 6: Constraint operation 2
- Perform *constraint operation 2* to generate the feasible sample set, $\boldsymbol{X}^{new,c}_{wolf} = [\boldsymbol{X}^{new,c}_{wolf,\boldsymbol{var\_obj}},\ \boldsymbol{X}^{new,c}_{wolf,\boldsymbol{var\_con}}]$.
- Output the feasible solution set $\boldsymbol{X}^{new,c}_{fe}$.

Step 7: Nondominated solution determination
- If $\boldsymbol{X}^{new,c}_{fe} \notin \boldsymbol{\Phi}$, find the nondominated solutions from $\boldsymbol{X}^{new,c}_{fe}$ according to the objective predicted value from ANNs, $\boldsymbol{X}_{non-dom} \leftarrow findNonDom(\boldsymbol{X}^{new,c}_{fe},\quad ANN\_obj(\boldsymbol{X}^{new,c}_{fe}))$.
- Otherwise, a sample is randomly selected from $X^{new,c}_{wolf}$, $\boldsymbol{X}_{non-dom} \leftarrow randSelect(\boldsymbol{X}^{new,c}_{wolf})$.
- Evaluate the nondominated solution(s) by simulation, $[\boldsymbol{X}_{non\text{-}dom}, \boldsymbol{F}_{non\text{-}dom}, \boldsymbol{G}_{non\text{-}dom}] \leftarrow \mathrm{simulation}(\boldsymbol{X}_{non\text{-}dom})$.
- Store the samples in the sample base, $[\boldsymbol{X}, \boldsymbol{F}, \boldsymbol{G}] \leftarrow [[\boldsymbol{X}, \boldsymbol{F}, \boldsymbol{G}];[\boldsymbol{X}_{non\text{-}dom}, \boldsymbol{F}_{non\text{-}dom}, \boldsymbol{G}_{non\text{-}dom}]]$.

Step 8: Pareto set updating
- Check the feasibility of $\boldsymbol{G}_{non\text{-}dom}$ by the real constraint values $(\boldsymbol{G}_{non\text{-}dom})$, $\boldsymbol{X}_{non\text{-}dom,fe} \leftarrow checkFeasible(\boldsymbol{X}_{non\text{-}dom}, \boldsymbol{G}_{non\text{-}dom})$.
- If $\boldsymbol{X}_{non-dom,fe} \notin \boldsymbol{\Phi}$, update the Pareto set, $[\boldsymbol{X}_P, \boldsymbol{F}_P, \boldsymbol{G}_p] \leftarrow updatePareto([\boldsymbol{X}_P, \boldsymbol{F}_P, \boldsymbol{G}_p], [\boldsymbol{X}_{non\text{-}dom,fe}, \boldsymbol{F}_{non\text{-}dom,fe}, \boldsymbol{G}_{non\text{-}dom,fe}])$.
- Otherwise, Pareto set remains.

Step 9: Constraint operation 3
- Perform the *constraints operation 3* to check the current Pareto set.

Step 10: Stopping criterion checking
- If the number of simulation evaluations is larger than the preset maximum number of function evaluations, terminate the optimization; otherwise, go back to *Step 3*.

3 Numerical Benchmark Problems Tests and Results

To test and verify the performance of the proposed algorithm, seven constrained multi-objective optimization problems [3] are employed in this section. The function of each benchmark is shown in the Appendix. Note that the objectives and constraints of the benchmarks are assumed to be black-box and expensive simulation models. In this case, the maximum number of function evaluations is set to a small number (i.e., 500) for all the algorithms to test the effectiveness of the proposed algorithm when the number of simulations is limited.

3.1 Comparison with ANN-GWO. The effectiveness of the dimension selection strategies used in the proposed algorithm is tested first by comparing it to the ANN-GWO, which does not contain the dimension selection stage. Different from L-ANN-GWO, in ANN-GWO, all variables are used to construct the ANNs for objectives and participate in the GWO updating process. Accordingly, *constraint operation 2* is not performed in the ANN-GWO. Instead, the candidate solutions are checked by the predicted values from the constraint ANN models, and only the feasible nondominated solutions are evaluated by the actual objective and constraint functions. In the tests, the maximum number of function evaluations for both methods is set to 500. The number of wolves is set to be 20.

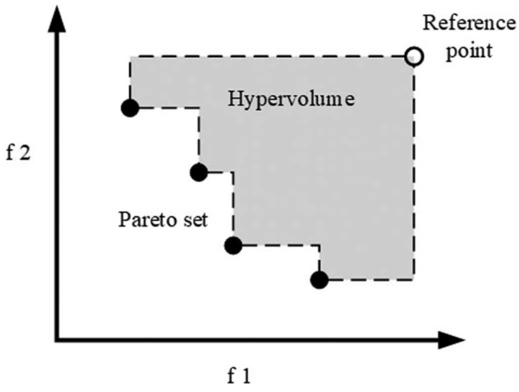

Fig. 3 2D illustration of the hypervolume criterion

Table 2 Dimension selection example for CF6 problem (a black dot means the dimension was selected)

No. of iteration	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	●	●	●	●	●		●	●	●	●
2	●	●	●	●	●		●	●	●	●
3	●	●	●	●	●	●		●	●	●
4	●	●	●	●	●	●		●	●	●
5	●	●	●	●	●		●	●	●	●
10	●	●	●	●	●		●	●	●	●
15	●	●	●	●	●	●	●	●	●	●
20	●	●	●	●	●	●	●	●	●	●
30	●	●	●	●	●	●	●	●	●	●
50	●	●	●	●	●	●	●	●	●	●
99	●	●	●	●	●	●	●	●	●	●

The performance criterion, hypervolume [40], is applied to judge the performance of each algorithm. The hypervolume measures the volume in the objective space that is encapsulated by the Pareto frontier and a reference point. To make the reference point consistent in the comparison of different algorithms, the point is defined as the maximum of all objective values in all three sets of Pareto solutions. For a given Pareto solution in a set of solutions, the hypercube using the solution and the reference point as the diagonal corners is defined and the volume of the hypercube is calculated. The summation of the volumes of the hypercubes defined by all the Pareto solutions is defined as the hypervolume of the Pareto set. As shown in Fig. 3, the area within the dotted line indicates the hypervolume of the Pareto set. A large hypervolume value means that the obtained solutions are close to the actual Pareto frontier, which indicates a better performance of the optimization algorithm. In this article, all the benchmark problems are solved by each algorithm 30 times, and the mean and standard deviation values are shown in Table 1.

As shown in Table 1, by adding the dimension selection strategy, L-ANN-GWO outperforms ANN-GWO in all the test problems with higher hypervolume values. The column named rate of success shows the percentage of the runs that find feasible Pareto solutions out of the total 30 runs. For example, for P113mod, the number of runs finding feasible Pareto solutions using ANN-GWO is 10 out of 30; therefore, the rate of success is 33.3%. Tables 2 and 3 show examples of selected variables in L-ANN-GWO solving CF6 and P116mod problems concerning the iterations. Note that the algorithm terminates according to the maximum number of function evaluations. As shown in step 7 in Sec. 2.5, every feasible nondominated solution generated at one iteration is evaluated by the simulation, which means the number of function evaluations at each iteration may vary for different test problems. Thus, the total number of iterations (i.e., the iteration number in the last row of Tables 2 and 3) is different for different problems. For the functions in the CF6 problem, every dimension shares similar importance to both objectives. As shown in Table 2, a few dimensions are not selected at the early stage of the optimization due to the lack of samples. By updating the LASSO selection with newly generated samples, all the dimensions are selected from the 15th iteration, which reflects well the mathematical definition of the problem. At the early stage, the update mechanism focusing on some of the dimensions makes the algorithm more aggressive to generate more nondominant solutions. As a result, when all the variables participated in the optimization, more nondominant solutions increase the spread of the Pareto frontier according to the leader selection strategy in the GWO updating mechanism.

Different from the CF6 problem, the objective functions of P116mod are only influenced by six variables, [x_1, x_2, x_3, x_{11}, x_{12}, x_{13}]. As shown in Table 3, the variables x_4, x_5, …, x_{10} are mistakenly selected at the early iterations. Then, the error is corrected by updating the LASSO regression with new samples. In the last few iterations, seven variables, [x_1, x_2, x_3, x_{10}, x_{11}, x_{12}, x_{13}], are selected to participate in the GWO updating process. In this case, the variables of the objectives are distinguished by the dimension selection strategy. The GWO update can focus on the variables affecting the objectives, and other variables are used to ensure the feasibility of the solutions with no change to the objective values. To be noticed, x_{10} is selected mistakenly even at the last 12 iterations. This may be because the log10 transfer creates a nonlinearity for each variable that may have caused this problem.

Compared to ANN-GWO, by performing the dimension selection strategy and the related *constraint operation 2*, L-ANN-GWO succeeded to find feasible solutions in all the tests. In the constraint handling process proposed in this article, the key is to find the first feasible solution as the leader. Since the GWO updating mechanism used in this article has no constraint handling operations, the feasibility of the new samples generated from GWO is unknown. In ANN-GWO, missing *constraint operation 2* generates randomness in the process of finding feasible solutions. On the other

Table 1 Comparison of L-ANN-GWO with ANN-GWO

	L-ANN-GWO		ANN-GWO	
	Hypervolume	Rate of success	Hypervolume	Rate of success
CF6	**21.71 (1.24)**	100%	21.00 (1.08)	100%
CF7	**254.77 (9.75)**	100%	245.15 (19.13)	100%
P113mod	**8.60E + 05 (4.81E + 04)**	100%	6.89E + 05 (5.28E + 04)	33.3%
TP3mod	**89.40 (27.09)**	100%	74.08 (32.72)	66.7%
P106mod[a]	**6.6E + 03 (0.54E + 03)**	–	–	–
P116mod	**8.45E + 03 (1.16E + 03)**	100%	6.26E + 03 (2.46E + 03)	100%
Beam	**0.026 (0.001)**	100%	0.025 (0.001)	60%

Note: Data are presented as mean (standard deviation). The best performances are presensed in bold face.
[a]P106mod is a special case, which is explained in Sec. 4.

Table 3 Dimension selection example for P116mod problem

No. of iteration	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
1	●	●	●			●	●	●	●		●	●	●
2	●	●	●			●	●	●	●		●	●	●
3	●	●	●	●		●		●	●	●	●	●	●
4	●	●	●	●	●	●		●	●	●	●	●	●
5	●	●	●			●		●			●		●
10	●	●	●	●	●		●	●	●	●	●	●	●
15	●	●	●	●	●		●	●	●	●	●	●	●
20	●	●	●			●	●		●	●	●	●	●
30	●	●	●			●	●			●	●	●	●
35	●	●	●							●	●	●	●
47	●	●	●							●	●	●	●

Table 4 Dimension selection example for P106mod problem

No. of iteration	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
21	●	●	●	●		●		
22	●	●	●	●		●		
23	●	●	●	●				
24	●	●	●	●		●		

hand, due to the lack of samples at the early stage, it is common that some of the dimensions are not selected from the LASSO selection method. Thus, via a local search on the nonselected variables, the chance to find feasible solutions at an early stage of the optimization increases. Therefore, the proposed L-ANN-GWO can successfully find feasible solutions in those problems where ANN-GWO fails (i.e., P113mod, TP3mod, and Beam).

For the P106mod problem, the objective functions are influenced by [x_1, x_2, x_3, x_4, x_6], and the constraints are with respect to [x_4, x_5, x_7, x_8]. A Pareto solution [$f_1 = 2100, f_2 = -0.95$] with [$x_1 = 100$, $x_2 = 1000$, $x_3 = 1000$, $x_4 = 10$, $x_6 = 10$] can be found to dominate almost any other solution. The values of x_5, x_7, and x_8 vary to satisfy the constraints. In other words, for P106mod, there exist multiple feasible Pareto solutions that have the same objective values. It is observed that ANN-GWO always finds those solutions with the same values for x_1, x_2, x_3, x_4, and x_6, but different values for x_5, x_7, and x_8. In this case, the rate of success for ANN-GWO is 100% but there is only one point shown in the Pareto frontier in the objective space. That is why the hypervolume value and rate of success are not reported in Table 1. L-ANN-GWO also has the same situation in 10 out of 30 tests, but it can also find other solutions such as [$f_1 = 2104$, $f_2 = -0.95$] and [$f_1 = 2100$, $f_2 = -0.93$] in other runs. One reason to find other solutions is that when one dimension always has the same value for the samples, the LASSO dimension selection process has the chance to eliminate this dimension to participate in the GWO updating mechanism. As shown in Table 4 of one example in the dimension selection, x_6 is not selected in the 23rd iteration. Then, the local search in *constraint operation 2* has a chance to modify the value of x_6 according to the feasibility to reach other Pareto solutions. As a result, a Pareto solution different from [$f_1 = 2100$, $f_2 = -0.95$] may be generated during the constraint handling step.

3.2 Comparison With Other Algorithms.

Two multi-objective optimization algorithms (NGPM [41] and MOFEPSO [42]), which are based on evolutionary algorithms, as well as one multi-objective BO method (MOEGO [34]) are employed as a comparative benchmark with the proposed algorithm. The NPGM (NSGA-II Program in Matlab v1.4) adds a rudimentary expensive constraint handling process in the NSGA-II algorithm. On the other hand, MOFEPSO is a multi-objective particle swarm optimization-based employing constraints handling strategies for feasible and infeasible particles. In MOEGO, the constrained achievement scalarization function optimization is solved to generate new samples at each iteration under the Bayesian optimization framework. NGPM, MOFEPSO, and MOEGO are recently included in MATLAB [41–43]. The maximum function evaluation for each algorithm is set to be 500. The population size is set to 20 for L-ANN-GWO and NPGM and 10 for MOFEPSO following the setting in Ref. [3]. Each problem is repeated 30 times, and the mean and standard deviation values of hypervolume are shown in Table 5. Note that since the reference points are different, the hypervolume values of L-ANN-GWO are different between Tables 1 and 5.

As shown in Table 5, L-ANN-GWO has the largest mean hypervolume values in six out of seven tests. MOFEPSO performs the best for CF6. For the TP3mod problem, however, MOFEPSO cannot find any feasible solutions in the given computational budget. MOEGO fails to find feasible solutions for the Beam problem. Compared to another surrogate model-based optimization algorithm, MOEGO, the proposed algorithm outperforms in all the tests at the same computational budget. Especially, MOGEO fails for the Beam problem, which is high-dimensional (30 dimensions) with many constraints (21 constraints).

To illustrate the comparison clearly, one single run that can represent the average performance of each algorithm is selected and plotted in Fig. 4. For the CF6 problem, most of the solutions from L-ANN-GWO are gathered in the lower f1 values, which reduces the performance of L-ANN-GWO. For CF7, L-ANN-GWO can generate more solutions with lower f1 function values, which leads to higher performance compared to NGPM. For the P113mode problem, although the performance of each algorithm is similar in the middle part, more solutions are found by L-ANN-GWO at each end of the frontier. The solutions for TP3mod, generated from L-ANN-GWO, clearly outperform the solutions found by NGPM. For the P106mod problem, L-ANN-GWO finds solutions with smaller values for the second objective, while MOFEPSO focuses on the smaller value of the first objective. Both algorithms have better performance than NGPM. The beam problem is a difficult problem with 30 design variables and 21 constraints. As shown in the results and plot, L-ANN-GWO reaches better solutions than the other two algorithms in dealing with a large number of design variables and many constraints. For the

Table 5 Comparison of L-ANN-GWO with NGPM, MOFEPSO, and MOEGO for benchmark problems using hypervolume values with best performances in boldface

	L-ANN-GWO	NGPM	MOFEPSO	MOEGO
CF6	76.18 (2.64)	78.44 (2.08)	**79.86 (1.32)**	66.97 (4.85)
CF7	**549.48 (14.92)**	534.55 (28.44)	518.26 (20.06)	439.13 (25.98)
P113mod	**8.75E + 05 (4.95E + 04)**	8.11E + 05 (3.70E + 04)	8.44E + 05 (1.10E + 04)	7.21E + 5 (6.36E + 04)
TP3mod	**92.29 (23.29)**	84.53 (32.65)	–	48.75 (7.33)
P106mod	**9.44E + 04 (1.86E + 04)**	8.57E + 04 (2.85E + 04)	9.04E + 04 (2.91E + 04)	5.44E + 04 (0.69E + 04)
P116mod	**7.46E + 03 (1.05E + 03)**	4.19E + 03 (1.06E + 03)	6.30E + 03 (1.98E + 03)	0.44E + 03 (0.13E + 03)
Beam	**0.067 (0.0050)**	0.042 (0.0376)	0.037 (0.0052)	—

Note: Data are presented as mean (standard deviation).

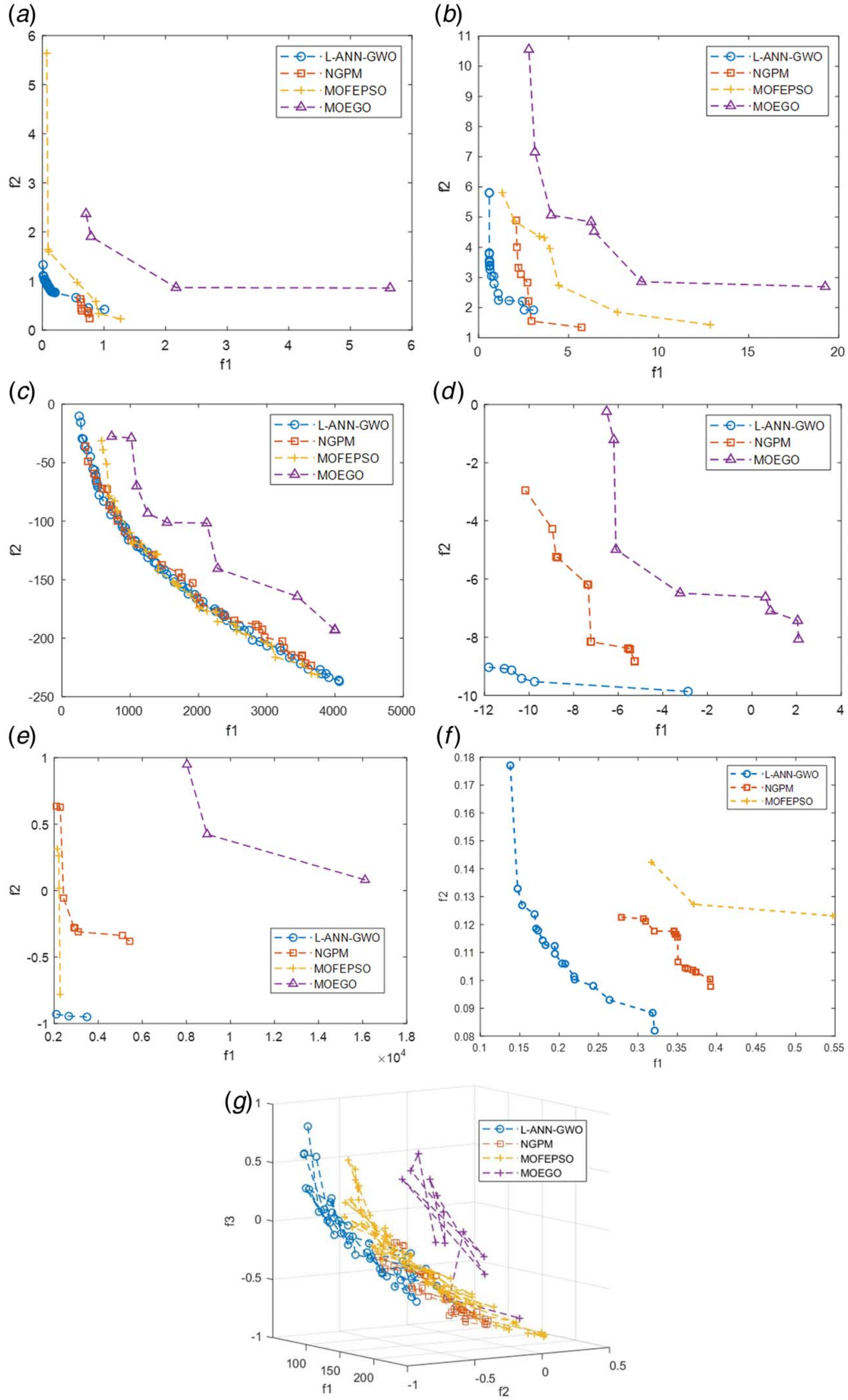

Fig. 4 Single-run solution plots of objectives of the numerical benchmark problems: (*a*) CF6, (*b*) CF7, (*c*) P113mod, (*d*) TP3mod, (*e*) P106mod, (*f*) Beam, and (*g*) P116mod

tri-objective optimization problem, P116mod, the results of L-ANN-GWO and MOFEPSO are similar and L-ANN-GWO can find more results on the edges, while the results of NGPM are gathered in a smaller area in the objective space. As a result, L-ANN-GWO performs well in numerical-constrained optimization problems when the number of function evaluations is limited. For some specific problems (e.g., CF6), the variable selection process using LASSO may generate results with a lower value for one objective. In that case, tightening the regression model accuracy threshold in LASSO regression will enable the algorithm to select more variables. The variables influencing any objective will have a large chance to be selected. On the other hand, the Pareto frontiers found by MOEGO are always dominated by the frontiers found by the other three algorithms.

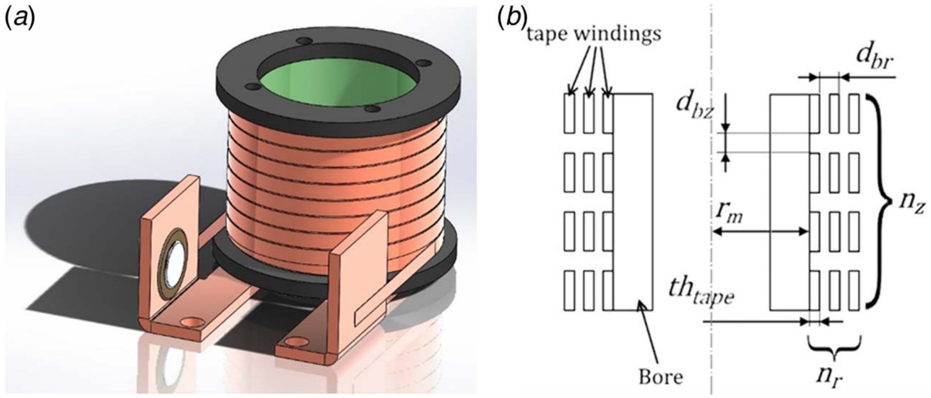

Fig. 5 Solenoid magnet design problem: (*a*) CAD model of the solenoid and (*b*) geometry parameters of the solenoid

4 Application in Solenoid Magnetic Design

Due to the capability of generating high magnetic fields, the magnet using 2G HTS tape is becoming an attractive solution for different engineering fields, such as particle accelerators. However, the nonlinear material property of the HTS tape makes it difficult and time consuming to calculate the magnetic field for an HTS magnet. Thus, the proposed algorithm is applied to deal with an optimization design problem of an HTS magnet using YBCO tape. Figure 5(*a*) shows the CAD model of a solenoid.

When applying an HTS magnet in the particle accelerators to guide the particle beam, a larger central magnetic field is needed when the speed of the particle increases. Additionally, superconducting materials are usually very expensive. Therefore, the goal is to design a magnet geometry so that the magnetic field at the center of the magnet is maximized with a minimal amount of required superconductor materials (measured by the length of the HTS tape). Finite element method (FEM) simulation with commercial software (COMSOL) is used to compute the magnetic field distribution in the coil, and each simulation can take up to 30 min. It is estimated that detailed mapping of the design space (e.g., six dimensions in this study) takes hundreds of simulations, so that there is a clear motivation to apply the L-ANN-GWO algorithm to reduce the magnet design time. Before going to the nonlinear HTS tape, we first solved a simpler case where the HTS tape was replaced with Cu tape. FEM simulation for Cu tape lasts only 2–3 min. We used the Cu tape simulation to benchmark the algorithm performance with the other three algorithms, NGPM, MOFEPSO, and MOEGO. After showing the capability of the proposed algorithm in dealing with the FEM simulation-based optimization problem, L-ANN-GWO is applied to solve the HTS tape magnetic field design problem.

It is worth mentioning that the model is parametric so that the design constraints (e.g., area of the uniform field at the center of the coil, or field homogeneity requirement) can be easily changed for future optimization studies. Moreover, the optimization algorithm is directly linked with the FEM solver, so that the expert user's time for preparing the FEM simulations and interpreting the results is minimized.

4.1 Formulation of the Optimization Problems. The magnetic design includes defining the coil geometry and choosing the transport current (I_t). The geometry is described by the number of turns in each winding (in each horizontal coil layer), the number of windings (number of coil layers stacked vertically), the thickness of the tape, the distance of tapes vertically, and the distance between tapes horizontally (radially), see Fig. 5(*b*) and Table 6. The goal of this design optimization case is to

- maximize the central magnetic field ($B0$)
- minimize the length of the tape conductor (L)

The following design criteria must be fulfilled:

- In the 2 cm × 2 cm area at the center of the solenoid, the difference between the maximal and minimal magnetic field (Bsq_{diff}) is less than 0.1 T. (The unit T is Tesla for the magnetic flux density.)
- In Cu tape: the maximum magnetic field (norm) in any of the tapes (B_{tape}) is 2 T.
- In HTS tape: the transport current does not exceed the critical current, $I_c(B)$, in any of the tapes. This ensures that the tape stays in the superconducting state. Note, that the tape I_c is a function of the magnetic field seen by the tape.
- The upper and lower bounds of each design variable are shown in Table 7.

The length of the tape (L) is calculated by the equation of the length of the spiral in polar coordinates, as shown in Eq. (4).

$$\begin{aligned} l &= \int_{\phi_0}^{\phi_1} \frac{d}{2\pi}\sqrt{1+\phi^2}d\phi \\ \phi_0 &= \frac{r_m}{d_{br}+th_{tape}} \times 2\pi \\ \phi_1 &= \phi_0 + 2\pi \times n_r \\ d &= d_{br} + th_{tape} \end{aligned} \tag{4}$$

Table 6 Geometry parameters of the solenoid

Parameters	Descriptions
L	The total length of the tape
th_{tape}	The thickness of the tape
n_r	Number of tapes in each winding
n_z	Number of windings in the coil
r_m	The radius of the magnet bore
d_{br}	Distance between tapes in each winding
d_{bz}	Distance between windings

Table 7 Design variables of the magnet design test

Design variables	Upper and lower bounds
n_r	[9, 36]
n_z	[4, 12]
r_m (cm)	[2.5, 10]
d_{br} (mm)	[0.05, 0.1]
I_t (A)	[100, 300]
d_{bz} (mm)	[0.25, 1]

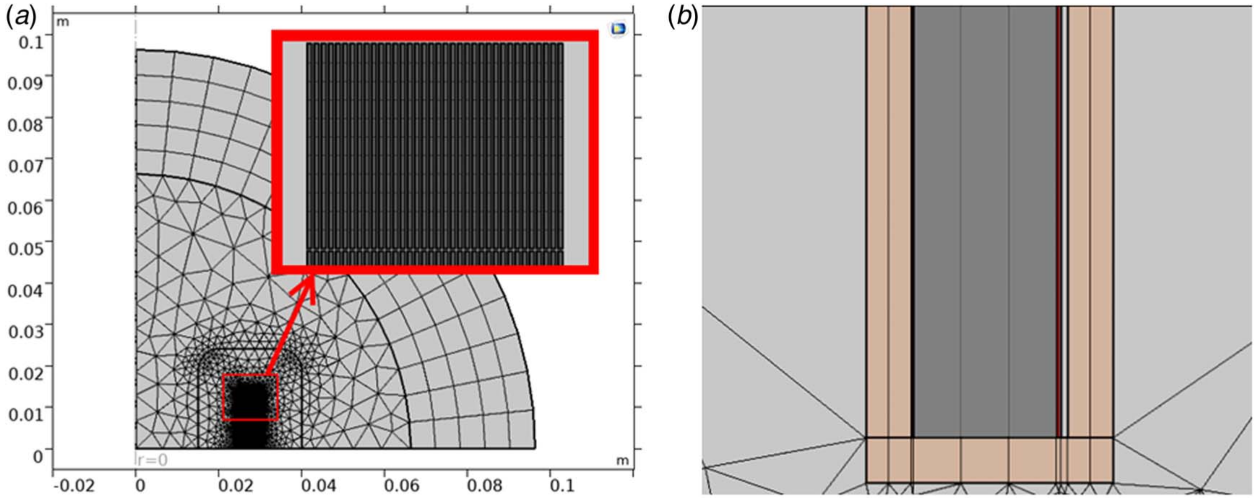

Fig. 6 Global mesh of the solenoid in the FEM model with an enlarged view of turns of winds in the red box (*a*), and a mesh for 2G HTS tape where different colors mark different layers: orange—copper; dark gray—substrate; light gray—silver; and brown—YBCO (*b*). (*a*) Global mesh and (*b*) detailed mesh between layers.

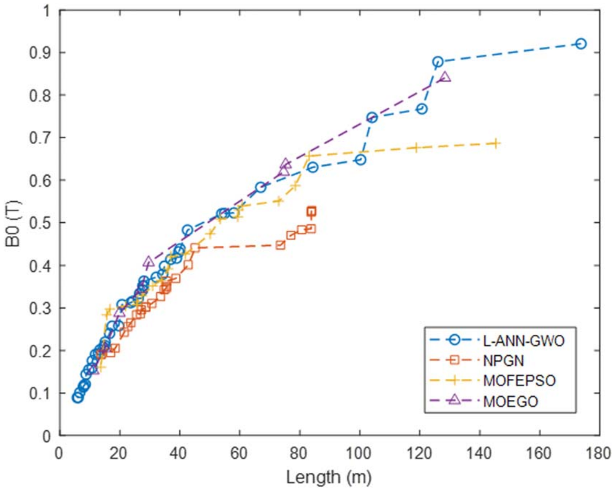

Fig. 7 Comparison of Pareto solutions to the Cu tape solenoid magnetic field design

The total length of the tape used is calculated by multiplying the number of windings, n_z and the length of each winding, l, i.e., $L = n_z \times l$. In this article, a 2G HTS tape provided by S-innovations [44] is used in the tests. The thickness of the tape, th_{tape}, is 109 μm. The Cu tape problem uses the same th_{tape} value.

The magnetic field in the center and the magnetic field-related criteria, i.e., $B0$, Bsq_{diff}, B_{tape}, and I_c, are computed from the FEM simulation. Maxwell's equations are solved for both the Cu tape magnet and the HTS magnet using the FEM solver in COMSOL. A 2D axisymmetric geometry for the solenoid model is used for the simulation. An A-H formulation, presented in Ref. [45], is chosen for Maxwell's expressions because it is high simulation speed and high stability. The mesh used in the simulation is shown in Fig. 6. The simulation and optimization are run on a laptop with Intel Core i7-7700HQ @2.8 Hz and 32 GB RAM.

Table 8 Hypervolume comparison with the other three algorithms

	L-ANN-GWO	NPGM	MOFEPSO	MOEGO
Hypervolume	89.21	59.73	77.00	80.66

In summary, the optimization problem for the Cu tape is formulated in Eq. (5) and that for the HTS tape in Eq. (6):

$$\begin{aligned} &\text{find } x = [n_r,\ n_z,\ r_m,\ d_{br},\ I_t,\ d_{bz}] \\ &\min L \\ &\max B0 \\ &\text{s.t. } Bsq_{diff} \le 0.1\ (T) \\ &B_{tape} \le 2\ (T) \\ &n_r \text{ and } n_z \text{ are integers} \end{aligned} \tag{5}$$

$$\begin{aligned} &\text{find } x = [n_r,\ n_z,\ r_m,\ hKp,\ I_t] \\ &\min L \\ &\max B0 \\ &\text{s.t. } Bsq_{diff} \le 0.1\ (T) \\ &I_t \le I_c \\ &n_r \text{ and } n_z \text{ are integers} \end{aligned} \tag{6}$$

In the HTS tape problem, the parameters d_{br} and d_{bz} are related to the Kapton insulation thickness (hKp) around the tape. The relation to the distance parameters is expressed as follows:

$$d_{br} = d_{bz} = 2 * hKp \tag{7}$$

Therefore, in the HTS tape problem, hKp is defined as one of the design variables to replace d_{br} and d_{bz}.

4.2 Optimization Results for Cu Tape Problem.

L-ANN-GWO, along with NGPM, MOFEPSO, and MOEGO, is applied to solve the Cu tape solenoid magnetic design problem. The maximum number of function evaluations for all three algorithms is set to 300. The population size for NGPM and the wolve size of L-ANN-GWO are set to 20, while the number of particles in MOFEPSO is set to 10. The Pareto frontiers obtained from three algorithms are shown in Fig. 7. The hypervolume values for the three Pareto sets are presented in Table 8. The running time for each of the algorithms is around 6 h.

As shown in the results, with the limited number of function evaluations, L-ANN-GWO generates better solutions compared to NGPM and MOFEPSO. When the length of the tape is in the range between 20 m and 50 m, the results of the three algorithms are comparable with L-ANN-GWO outperforming the other two algorithms. In specific, L-ANN-GWO finds more solutions whose

Table 9 Design variable and objective values of the baseline design and two examples of Pareto solutions

	n_r	n_z	r_m (cm)	hKp (μm)	I_t (A)	B0 (T)	Length (m)
Baseline	9	8	4	30	200	0.1929	18.4213
Solution 1	11	9	2.56	18.99	146.30	0.2677	16.3799
Solution 2	9	8	2.50	46.22	146.68	0.2046	11.7015

length is lower than 50 and the magnetic field is higher. When the length of the tape is longer than 50 m, the advantages of L-ANN-GWO become greater. On the other hand, compared to the MOEGO, L-ANN-GWO generates more solutions along the Pareto frontier, especially when the length of the tape is lower than 50. That is the reason that the Hypervolume value of L-ANN-GWO is larger than that of the MOEGO. As a result, L-ANN-GWO leads to designs with a much stronger magnetic field with the same tape length as the other three algorithms.

4.3 2G Second-Generation High-Temperature Superconductor Tape Problem Results and Analysis.

A baseline of the 2G HTS tape design is simulated first. The values of the design variables, the length of the tapes, and the central magnetic field value are shown in Table 9. The magnetic field distribution for the baseline design is shown in Fig. 8. Then, L-ANN-GWO is applied in a 2G HTS tapes magnetic design problem to test the performance of the proposed algorithm in real superconductivity magnetic design cases. As the 2G HTS tape is used in the simulation, one simulation may last from several minutes to half an hour to obtain the magnetic field results, according to the number of windings and the number of tapes in each winding. Considering the computational cost of the simulation, the number of function evaluations is limited to 200. The number of wolves is set to be 20. The total running time for the optimization is around 50 h.

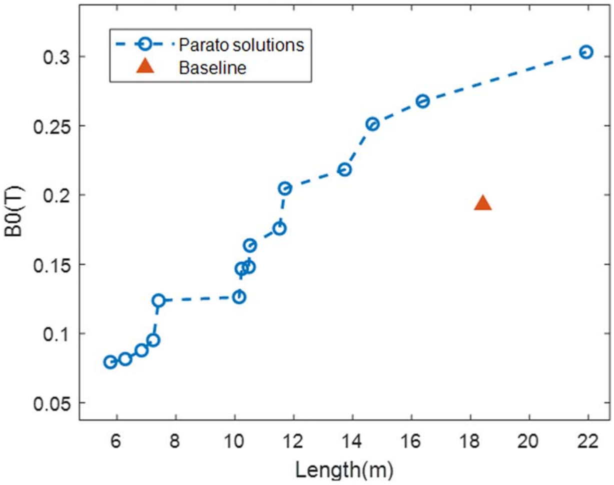

Fig. 9 Pareto frontier of the 2G HTS tape solenoid magnet optimization

Figure 9 shows the Pareto frontier obtained from the optimization, compared with the baseline design. In summary, L-ANN-GWO provides 15 feasible Pareto solutions using 200 simulation evaluations in the 2G HTS tape magnetic design case. The central magnetic field varies from 0.08 T to 0.30 T with the length of the tape between 5.7 m and 22.0 m. Table 9 shows two examples of the Pareto solutions obtained. Compared to the baseline design, the magnetic field can be improved by 31% with a similar tape length (solution 1), or the length of the tape can be reduced by 33% when the magnetic field is similar to the baseline (solution 2). The magnetic field distributions for the two solutions are shown in Fig. 10. The magnetic field strength at the original point of both Figs. 7 and 9 shows the values of *B0* for the three designs. Thus, solution 1 in Fig. 10(*a*) shows the highest *B0* value, while solution 2 is comparable to the baseline. Compared to the baseline design, the HTS tape windings in both solutions are closer to the original point, as shown in the *x*-axis values in Figs. 8 and 10. In other words, the radius of the magnet bore (r_m) in both solutions is smaller than that in the baseline design. In Fig. 8, the magnetic field reduces starting from 0.02 m to the center of the bore, which means that the HTS tape is too far from the center. On the other hand, by reducing the r_m values, solution 1 and solution 2 generate a larger magnet field at the center of the

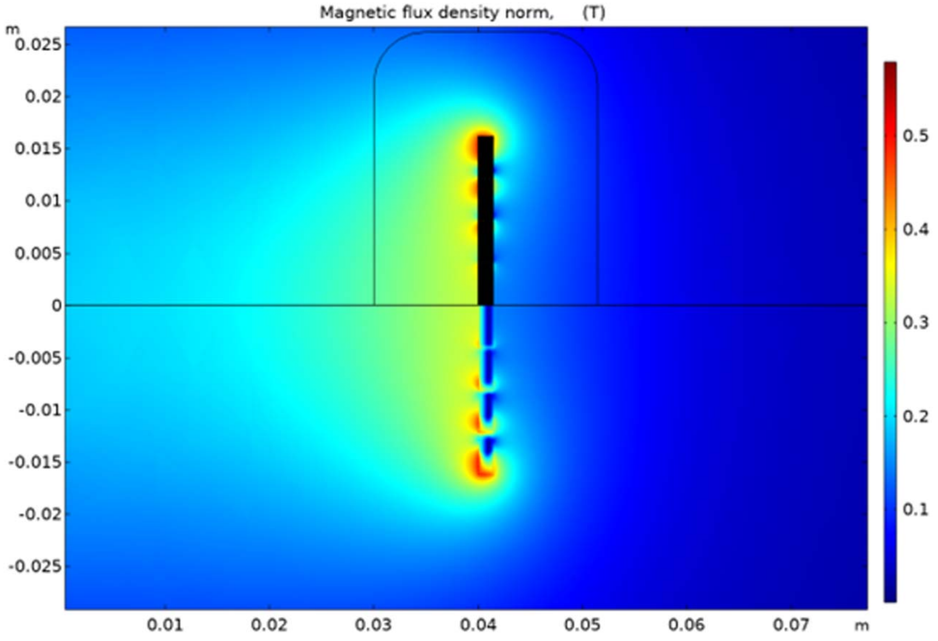

Fig. 8 Magnetic field distribution of the baseline design. (*x* and *y* axes show the dimensions of the area in meters; the color bar shows the magnetic field strength in unit T.)

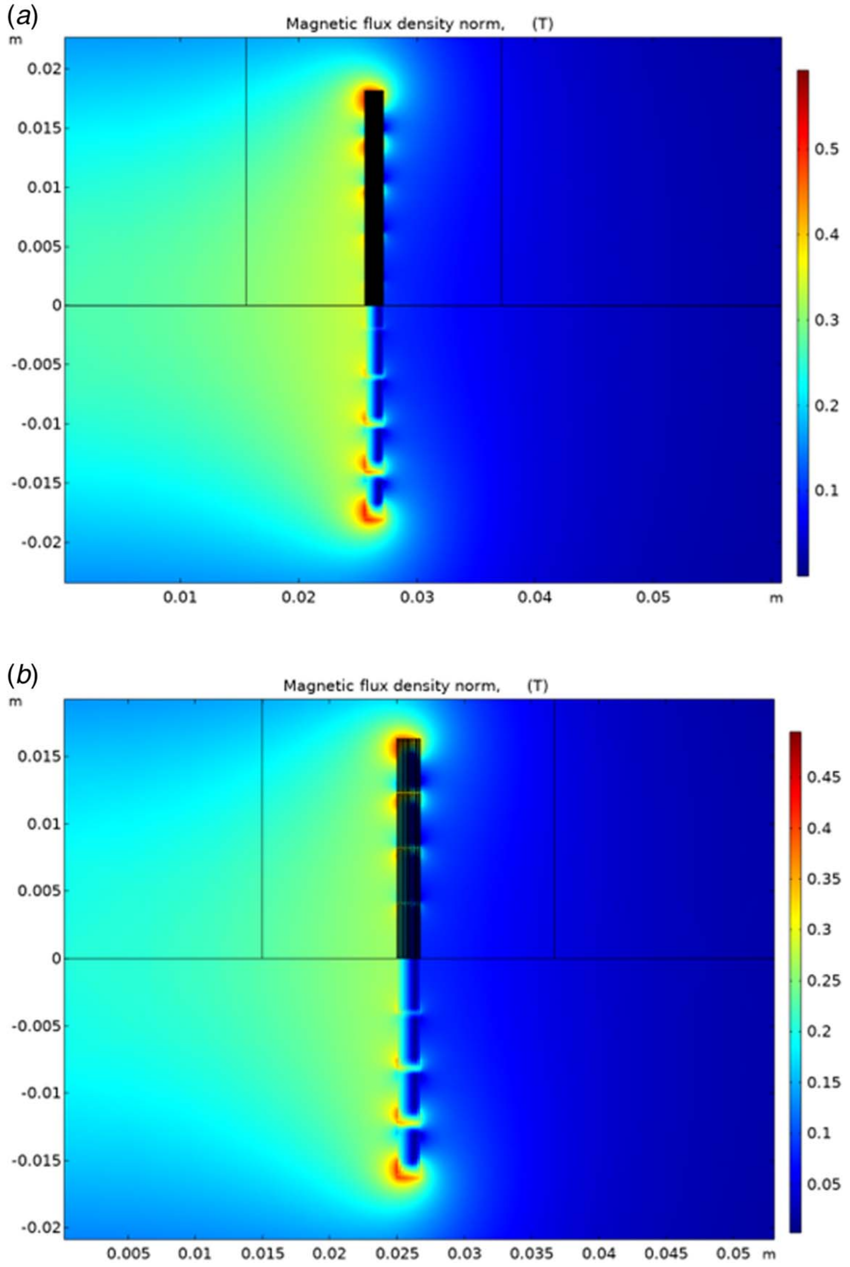

Fig. 10 Magnetic field distributions of the two example solutions: (*a*) solution 1 and (*b*) solution 2

bore. Additionally, for solution 1, *B0* is further improved by increasing the number of windings and the number of tapes in each winding, while the total length of tapes is reduced by reducing the thickness *hKp*. The current (I_t) is reduced to fulfill the constraints of the constraints. For solution 2, reducing r_m leads to a decrease in the length. The values of *hKp* and I_t are adjusted to fulfill the constraints.

5 Conclusion

This work proposes a dimension selection-based multi-objective optimization algorithm with the capability of handling expensive constraints. The motivation for such an effort lies in the very tight requirements involved in the 2G HTS magnetic field design. The results demonstrate that the new L-ANN-GWO algorithm outperforms the NGPM, MOFEPSO, and MOEGO in most of the numerical benchmark problems and the Cu tape solenoid magnetic field optimization problem. Finally, 15 Pareto solutions were generated from the optimization of the 2G HTS design.

Besides finding good designs for the HTS design case, key contributions of the proposed method are listed as follows:

- A new dimension selection-based multi-objective optimization algorithm is designed for simulation-based problems. The algorithm is benchmarked and successfully applied to a complex real-world engineering design problem.
- The dimension selection strategy using LASSO regression improves the efficiency of the optimization when the number of simulations is limited. The dimensions selected from LASSO have large influences on the objective values. Thus, focusing on those dimensions in the GWO updating mechanism generates large improvement in the objective values. Additionally, since the LASSO regression model is updated along the optimization process, more accurate regression models are generated and gradually more dimensions are considered in the GWO updating process, which helps to find new nondominant solutions at the latter stage of the optimization. Additionally, the error selection due to the lack of data at the

early iterations is corrected due to the updating process at each iteration.

- The separation of variables for objective minimization and constraint satisfaction shows promises. Since the nonselected dimensions have less impact on the objective values, those dimensions are more efficiently used to find a feasible solution than pursuing lower objective values. On the other hand, by modifying the values of selected and nonselected dimensions for separate purposes, new nondominant and feasible solutions can be generated more efficiently at the early stage of the optimization to provide high-quality leaders for GWO updates.

Since the algorithm is designed to solve the multi-objective optimization problem with two or three objectives, the ability of the proposed algorithm in dealing with many-objective optimization problems is not evaluated in this article. This will be our future work.

Funding Data

- Business Finland Project #3972/31/2019 (SMARAGDI).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

Appendix: Numerical Benchmark Problems in Sec. 4

CF6

$$
\begin{aligned}
&f_1(\boldsymbol{x}) = x_1 + \sum_{j\in J_1} y_j^2\\
&f_2(\boldsymbol{x}) = (1-x_1)^2 + \sum_{j\in J_2} y_j^2\\
&g_1(\boldsymbol{x}) = x_2 - 0.8x_1 \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - \operatorname{sign}(0.5(1-x_1)-(1-x_1)^2)\sqrt{|0.5(1-x_1)-(1-x_1)^2|} \geq 0\\
&g_2(\boldsymbol{x}) = x_4 - 0.8x_1 \sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - \operatorname{sign}(0.25(1-x_1)-0.5(1-x_1))\sqrt{|0.25(1-x_1)-0.5(1-x_1)|} \geq 0\\
&J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}\\
&J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}\\
&y_j = \begin{cases} x_j - 0.8x_1 \cos\left(6\pi x_1 + \dfrac{j\pi}{n}\right) & \text{if } j \in J_1\\ x_j - 0.8x_1 \sin\left(6\pi x_1 + \dfrac{j\pi}{n}\right) & \text{if } j \in J_2 \end{cases}\\
&x_1 \in [0, 1];\ \ x_2, \ldots, x_n \in [-2, 2];\ n = 10
\end{aligned}
\tag{A1}
$$

CF7

$$
\begin{aligned}
&f_1(\boldsymbol{x}) = x_1 + \sum_{j\in J_1} h_j(y_j)\\
&f_2(\boldsymbol{x}) = (1-x_1)^2 + \sum_{j\in J_2} h_j(y_j)\\
&g_1(\boldsymbol{x}) = x_2 - 0.8x_1 \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - \operatorname{sign}(0.5(1-x_1)-(1-x_1)^2)\sqrt{|0.5(1-x_1)-(1-x_1)^2|} \geq 0\\
&g_2(\boldsymbol{x}) = x_4 - 0.8x_1 \sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - \operatorname{sign}(0.25(1-x_1)-0.5(1-x_1))\sqrt{|0.25(1-x_1)-0.5(1-x_1)|} \geq 0\\
&J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}\\
&J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}\\
&y_j = \begin{cases} x_j - 0.8x_1 \cos\left(6\pi x_1 + \dfrac{j\pi}{n}\right) & \text{if } j \in J_1\\ x_j - 0.8x_1 \sin\left(6\pi x_1 + \dfrac{j\pi}{n}\right) & \text{if } j \in J_2 \end{cases}\\
&h_2(t) = h_4(t) = t^2;\ h_j(t) = 2t^2 - \cos(4\pi t) + 1 \ \text{ for } j = 3, 4, \ldots, n\\
&x_1 \in [0, 1];\ \ x_2, \ldots, x_n \in [-2, 2];\ n = 10
\end{aligned}
\tag{A2}
$$

P113mod

$$\begin{aligned}
f_1(\boldsymbol{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 \\
&\quad + (x_5-3)^2 + 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 \\
&\quad + (x_{10}-7)^2 + 45 \\
f_2(\boldsymbol{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \\
g_1(\boldsymbol{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0 \\
g_2(\boldsymbol{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0 \\
g_3(\boldsymbol{x}) &= 3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 - 120 \le 0 \\
&\quad x \in [-10, 10]
\end{aligned} \tag{A3}$$

TP3mod

$$\begin{aligned}
f_1(\boldsymbol{x}) &= 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \\
f_2(\boldsymbol{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \\
g_1(\boldsymbol{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \\
g_2(\boldsymbol{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \\
g_3(\boldsymbol{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0
\end{aligned} \tag{A4}$$

P106mod

$$\begin{aligned}
f_1(\boldsymbol{x}) &= x_1 + x_2 + x_3 \\
f_2(\boldsymbol{x}) &= 0.0025(x_4 + x_6) - 1 \\
g_1(\boldsymbol{x}) &= 0.0025(x_5 + x_7 - x_4) - 1 \le 0 \\
g_2(\boldsymbol{x}) &= 0.01(x_8 - x_5) - 1 \le 0 \\
& x_1 \in [1e2, 1e4], \ \{x_2, x_3\} \in [1e3, 1e4], \\
& \{x_4, x_5, x_6, x_7, x_8\} \in [10, 1e3]
\end{aligned} \tag{A5}$$

P116mod

$$\begin{aligned}
f_1(\boldsymbol{x}) &= x_{11} + x_{12} + x_{13} \\
f_2(\boldsymbol{x}) &= x_2 - x_3 \\
f_3(\boldsymbol{x}) &= x_1 - x_2 \\
g_1(\boldsymbol{x}) &= 0.002x_7 - 0.002x_8 - 1 \le 0 \\
g_2(\boldsymbol{x}) &= 50 - x_{11} - x_{12} - x_{13} \le 0 \\
g_3(\boldsymbol{x}) &= x_{11} + x_{12} + x_{13} - 250 \le 0 \\
g_4(\boldsymbol{x}) &= 1.262626x_{10} - 1.231059x_3x_{10} - x_{13} \le 0 \\
g_5(\boldsymbol{x}) &= 0.03475x_2 + 0.975x_2x_5 - 0.00975x_2^2 - x_5 \le 0 \\
g_6(\boldsymbol{x}) &= 0.03475x_3 + 0.975x_3x_6 - 0.00975x_3^2 - x_6 \le 0 \\
& \{x_1, x_2, x_3\} \in [0.1, 1], \ x_4 \in [1e-4, 0.1], \{x_5, x_6\} \in [0.1, 0.9], \\
& \{x_7, x_8\} \in [0.1, 1e3], x_9 \in [500, 1000], \ x_{10} \in [0.1, 500], \\
& x_{11} \in [1, 150], \ \{x_{12}, x_{13}\} \in [1e4, 150]
\end{aligned} \tag{A6}$$

Beam

$$\begin{aligned}
f_1(\boldsymbol{x}) &= \frac{P}{3E}\sum_{i=1}^{10}\left\{\frac{12}{x_{3i-2}x_{3i-1}^2}\left[\left(\sum_{j=i}^{10} x_{3j}\right)^3 - \left(\sum_{j=i+1}^{10} x_{3j}\right)^3\right]\right\} \\
f_2(\boldsymbol{x}) &= \sum_{i=1}^{10} x_{3i-2}x_{3i-1}x_{3i} \\
g_i(\boldsymbol{x}) &= \frac{6P}{x_{3i-2}x_{3i-1}^2}\sum_{j=i}^{10} x_{3j} - \sigma_{allow} \le 0, \quad i = 1, 2, \ldots, 10 \\
g_{10+i}(\boldsymbol{x}) &= \frac{x_{3i-1}}{x_{3i-2}} - AR \le 0, \quad i = 1, 2, \ldots, 10 \\
g_{21}(\boldsymbol{x}) &= L_{min} - \sum_{i=1}^{10} x_{3i} \le 0 \\
& x_{3i-2} \in [0.01, 0.05], \ x_{3i-1} \in [0.3, 0.65], \\
& x_{3i} \in [0.5, 1], \quad i = 1, 2, \ldots, 10 \\
& P = 50e3, \ E = 200e9, \ \sigma_{allow} = 35e7, \ AR = 25, \ L_{min} = 6
\end{aligned} \tag{A7}$$

References

[1] Wilson, B., Cappelleri, D., Simpson, T. W., and Frecker, M., 2001, "Efficient Pareto Frontier Exploration Using Surrogate Approximations," Optim. Eng., **2**(1), pp. 31–50.

[2] Shan, S., and Wang, G. G., 2005, "An Efficient Pareto Set Identification Approach for Multiobjective Optimization on Black-Box Functions," ASME J. Mech. Des., **127**(5), pp. 866–874.

[3] Cheng, G. H., Gary Wang, G., and Hwang, Y.-M., 2021, "Multi-Objective Optimization for High-Dimensional Expensively Constrained Black-Box Problems," ASME J. Mech. Des., **143**(11), p. 111704 .

[4] Brochu, E., Cora, V. M., and de Freitas, N., 2010, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, With Application to Active User Modeling and Hierarchical Reinforcement Learning," arXiv.

[5] Knowles, J., 2006, "ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems," IEEE Trans. Evol. Comput., **10**(1), pp. 50–66.

[6] Emmerich, M. T. M., Giannakoglou, K. C., and Naujoks, B., 2006, "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," IEEE Trans. Evol. Comput., **10**(4), pp. 421–439.

[7] Biswas, A., Fuentes, C., and Hoyle, C., 2022, "A Multi-Objective Bayesian Optimization Approach Using the Weighted Tchebycheff Method," ASME J. Mech. Des., **144**(1), p. 011703 .

[8] Tran, A., Eldred, M., McCann, S., and Wang, Y., 2022, "srMO-BO-3GP: A Sequential Regularized Multi-Objective Bayesian Optimization for Constrained Design Applications Using an Uncertain Pareto Classifier," ASME J. Mech. Des., **144**(3), p. 031705 .

[9] Abdolshah, M., Shilton, A., Rana, S., Gupta, S., and Venkatesh, S., 2018, "Expected Hypervolume Improvement With Constraints," International Conference on Pattern Recognition, Beijing, China, Aug. 20–24, pp. 3238–3243.

[10] Yang, K., Emmerich, M., Deutz, A., and Bäck, T., 2019, "Multi-Objective Bayesian Global Optimization Using Expected Hypervolume Improvement Gradient," Swarm Evol. Comput., **44**, pp. 945–956.

[11] Abdolshah, M., Shilton, A., Rana, S., Gupta, S., and Venkatesh, S., 2019, "Multi-Objective Bayesian Optimisation With Preferences Over Objectives," NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, Canada.

[12] Hernández-Lobato, D., Hernandez-Lobato, J., Shah, A., and Adams, R., 2016, "Predictive Entropy Search for Multi-Objective Bayesian Optimization," International Conference on Machine Learning, New York, June 19–24, pp. 1492–1501.

[13] Gu, Q., Wang, Q., Li, X., and Li, X., 2021, "A Surrogate-Assisted Multi-Objective Particle Swarm Optimization of Expensive Constrained Combinatorial Optimization Problems," Knowledge-Based Syst., **223**, p. 107049.

[14] Santana-Quintero, L. V., Coello Coello, C. A., Hernández-Díaz, A. G., and Velázquez, J. M. O., 2008, "Surrogate-Based Multi-Objective Particle Swarm Optimization," 2008 IEEE Swarm Intelligence Symposium., St. Louis, MO, Sept. 21–23, pp. 1–8.

[15] Shankar Bhattacharjee, K., Kumar Singh, H., and Ray, T., 2016, "Multi-Objective Optimization With Multiple Spatially Distributed Surrogates," ASME J. Mech. Des., **138**(9), p. 091401.

[16] Regis, R. G., and Shoemaker, C. A., 2013, "Combining Radial Basis Function Surrogates and Dynamic Coordinate Search in High-Dimensional Expensive Black-Box Optimization," Eng. Optim., **45**(5), pp. 529–555.

[17] Wu, Y., Long, T., Shi, R., and Gary Wang, G., 2021, "Mode-Pursuing Sampling Method Using Discriminative Coordinate Perturbation for High-Dimensional Expensive Black-Box Optimization," ASME J. Mech. Des., **143**(4), p. 041703.

[18] Wu, D., Hajikolaei, K. H., and Wang, G. G., 2017, "Employing Partial Metamodels for Optimization With Scarce Samples," Struct. Multidiscip. Optim., **57**(3), pp. 1329–1343.

[19] Chen, K., Laghrouche, S., and Djerdir, A., 2019, "Degradation Model of Proton Exchange Membrane Fuel Cell Based on a Novel Hybrid Method," Appl. Energy, **252**, p. 113439.

[20] Zhang, T., Li, F., Zhao, X., Qi, W., and Liu, T., 2022, "A Convolutional Neural Network-Based Surrogate Model for Multi-Objective Optimization Evolutionary Algorithm Based on Decomposition," Swarm Evol. Comput., **72**, p. 101081.

[21] Zhang, X., Tian, Y., Cheng, R., and Jin, Y., 2018, "A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization," IEEE Trans. Evol. Comput., **22**(1), pp. 97–112.

[22] Fan, Z., Yi, F., Li, W., Lu, J., Cai, X., and Wei, C., 2017, "A Comparative Study of Constrained Multi-Objective Evolutionary Algorithms on Constrained Multi-Objective Optimization Problems," 2017 IEEE Congress on Evolutionary Computation, San Sebastian, Spain, June 5–8, pp. 209–216.

[23] Woldesenbet, Y. G., Yen, G. G., and Tessema, B. G., 2009, "Constraint Handling in Multiobjective Evolutionary Optimization," IEEE Trans. Evol. Comput., **13**(3), pp. 514–525.

[24] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evol. Comput., **6**(2), pp. 182–197.

[25] Takahama, T., and Sakai, S., "Constrained Optimization by ϵ Constrained Particle Swarm Optimizer with ϵ-level Control," The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology, Muroran, Japan, May 25–27, pp. 1019–1029.

[26] Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., and Goodman, E., 2019, "Push and Pull Search for Solving Constrained Multi-Objective Optimization Problems," Swarm Evol. Comput., **44**, pp. 665–679.

[27] Tian, Y., Zhang, T., Xiao, J., Zhang, X., and Jin, Y., 2021, "A Coevolutionary Framework for Constrained Multiobjective Optimization Problems," IEEE Trans. Evol. Comput., **25**(1), pp. 102–116.

[28] Cai, Z., and Wang, Y., 2006, "A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization," IEEE Trans. Evol. Comput., **10**(6), pp. 658–675.

[29] Ray, T., Singh, H. K., Isaacs, A., and Smith, W., 2009, "Infeasibility Driven Evolutionary Algorithm for Constrained Optimization," Stud. Comput. Intell., **198**, pp. 145–165.

[30] Datta, R., and Regis, R. G., 2016, "A Surrogate-Assisted Evolution Strategy for Constrained Multi-Objective Optimization," Expert Syst. Appl., **57**, pp. 270–284.

[31] de Winter, R., van Stein, B., Dijkman, M., and Bäck, T., 2019, "Designing Ships Using Constrained Multi-Objective Efficient Global Optimization," *Lecture Notes in Computer Science*, Vol. 11331, Springer, Cham, pp. 191–203.

[32] de Winter, R., Bronkhorst, P., van Stein, B., and Bäck, T., 2022, "Constrained Multi-Objective Optimization With a Limited Budget of Function Evaluations," Memetic Comput., **14**(2), pp. 151–164.

[33] Bagheri, S., Konen, W., Emmerich, M., and Bäck, T., 2017, "Self-adjusting Parameter Control for Surrogate-Assisted Constrained Optimization Under Limited Budgets," Appl. Soft Comput., **61**, pp. 377–393.

[34] Hussein, R., and Deb, K., 2016, "A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-Objective Optimization," Proceedings of the Genetic and Evolutionary Computation Conference, Denver, CO, June 20–24, pp. 573–580.

[35] Tibshirani, R., 1996, "Regression Shrinkage and Selection via the Lasso," J. R. Stat. Soc. Ser. B, **58**(1), pp. 267–288.

[36] Papadrakakis, M., Lagaros, N. D., and Tsompanakis, Y., 1998, "Structural Optimization Using Evolution Strategies and Neural Networks," Comput. Methods Appl. Mech. Eng., **156**(1–4), pp. 309–333.

[37] Dutta, S., Bandopadhyay, S., Ganguli, R., and Misra, D., 2010, "Machine Learning Algorithms and Their Application to Ore Reserve Estimation of Sparse and Imprecise Data," J. Intell. Learn. Syst. Appl., **2**(2), pp. 86–96.

[38] Mirjalili, S., Mirjalili, S. M., and Lewis, A., 2014, "Grey Wolf Optimizer," Adv. Eng. Softw., **69**, pp. 46–61.

[39] Mirjalili, S., Saremi, S., Mirjalili, S. M., and Coelho, L. D. S., 2016, "Multi-Objective Grey Wolf Optimizer: A Novel Algorithm for Multi-Criterion Optimization," Expert Syst. Appl., **47**, pp. 106–119.

[40] Durillo, J. J., and Nebro, A. J., 2011, "jMetal: A Java Framework for Multi-Objective Optimization," Adv. Eng. Softw., **42**(10), pp. 760–771.

[41] Lin, S., 2022, "NGPM—A NSGA-II Program in Matlab v1.4," MATLAB Central File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4, Accessed January 24, 2022.

[42] Hasanoglu, M. S., 2022, "MOFEPSO: Multi-Objective Feasibility Enhanced Particle Swarm," MATLAB Central File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/68990-mofepso-multi-objective-feasibility-enhanced-particle-swarm, Accessed January 24, 2022.

[43] Tian, Y., Cheng, R., Zhang, X., and Jin, Y., 2017, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]," IEEE Comput. Intell. Mag., **12**(4), pp. 73–87.

[44] "2G HTS Wire," https://eng.s-innovations.ru, Accessed October 20, 2021.

[45] Robert, B. C., Fareed, M. U., and Ruiz, H. S., 2019, "How to Choose the Superconducting Material Law for the Modelling of 2G-HTS Coils," Mater., **12**(17), p. 2679.