An athlete–referee dual learning system for real-time optimization with large-scale complex constraints

Zhang, Yuchen; Liu, Jizhe; Xu, Yan; Dong, Zhao Yang

Published in:
Knowledge-Based Systems

Published: 08/07/2023

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY-NC-ND

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1016/j.knosys.2023.110566

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.
An athlete–referee dual learning system for real-time optimization with large-scale complex constraints

Yuchen Zhang, Jizhe Liu, Yan Xu, Zhao Yang Dong

School of Electrical Engineering and Telecommunications, The University of New South Wales, UNSW Sydney, NSW 2052, Australia
School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

ARTICLE INFO

Article history:
Received 16 November 2022
Received in revised form 8 February 2023
Accepted 10 April 2023
Available online 14 April 2023

Keywords:
Data-driven optimization
Deep learning
Machine learning
Neural networks

ABSTRACT

Constrained optimization (CO) has made a profound impact in solving many real-world problems. Due to the high computation burden in exact solvers, data-driven CO based on machine learning techniques is recently receiving extensive research interests for its capability to solve CO problems in real time. The existing data-driven CO approaches only serve for optimization problems with rather simple constraints that can be directly incorporated into model training. However, constraints that are computationally infeasible or burdensome to evaluate are commonly experienced in realistic optimization applications, especially in the engineering sector. This paper proposes an athlete–referee dual learning system (ARDLS) for end-to-end CO with large-scale complex constraints, where an athlete model is trained as the main optimizer while a referee model is trained as a probabilistic constraint classifier to guide the athlete training. A risk-based constrained loss function is designed to fine-tune the athlete model for constraint satisfaction. A case study on electric power system emergency control application is conducted to validate the proposed ARDLS, where the testing results demonstrate the excellent capability of ARDLS to improve the likelihood of satisfying large-scale complex constraints in CO.

© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Constrained optimization (CO) has been widely used as an effective tool to formulate and solve many real-world problems. Based on the complex principles behind the target applications, complicated constraints are usually inevitable in the optimization. Various exact solvers have long been used to precisely solve CO problems. Due to the high computation burden in the iterative process, these solvers are time-consuming on large-scale problems and can hardly fit in the real-time computation timeframe. Such time delay in finding the solutions makes exact solvers inappropriate for optimization tasks on dynamic systems with time-to-time variations, such as electricity systems [1], water systems [2] and mechanical systems [3]. We mostly see current optimization solving technologies to be applied for long-term time-ahead problems such as planning and scheduling, rather than more time-urgent applications such as dynamic operation and real-time control. Faster solvers for optimization with complex constraints have seen imperative demand in the engineering market, with the potential capability to maintain optimal engineering performance in real time.

Data-driven techniques have recently been employed to improve the solving efficiency of optimization, and machine learning has taken a pivotal role in these achievements. They can be categorized as learning-augmented optimization approaches and end-to-end optimization learning approaches [4]. Learning-augmented optimization uses machine learning to aid the performance of existing optimization solvers. For example, a learning restart strategy was proposed in [5] to minimize the computational cost of optimization. A rule-based learning method was proposed in [6] to ignore some optimization variables and consequently lead to faster solvers. In [7,8], the size of the CO problems to be solved was reduced by learning active constraints to enhance solving efficiency. Learning-augmented variants of branching and bound algorithms have also been developed to solve combinatorial optimization problems [9,10]. However, such augmented solvers still require iterative runtime by nature, which makes some potential real-time applications impractical. By contrast, end-to-end optimization learning focuses on implementing machine learning models to directly estimate the optimal solution for the optimization problem, which can fulfill the real-time requirements of most applications.

https://doi.org/10.1016/j.knosys.2023.110566
0950-7051/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
In the realm of end-to-end CO, several data-driven approaches have been developed to achieve real-time optimization capability. For instance, in [11–14], a lagrangian dual method has been used to convert CO into an unconstrained optimization problem. Reinforcement learning techniques have been used in [14–16] to train the machine learning model to satisfy the optimization constraints. A moving target method was proposed in [17] for data-driven CO, which is a decomposition-based approach that alternates master and learner steps to enforce constraints in supervised learning. Most of these methods require the evaluation of constraints during model training, which is only feasible for solving CO problems with constraints that can be evaluated straightforwardly, such as the optimal power flow problem for electric power systems [14]. However, many engineering problems are modelled in large-scale complex constraints that are computationally burdensome to evaluate. These constraints, if incorporated into the iterative training process, can result in practically unacceptable training burdens. Moreover, the existing approaches cannot rigorously guarantee the satisfaction of optimization constraints, leaving huge gaps in practical applications.

This paper proposes an athlete–referee dual learning system (ARDLS) for real-time end-to-end optimization with large-scale complex constraints that are infeasible to be directly evaluated during the training stage, which can serve as a universal data-driven approach for constrained optimization. The main contributions of the ARDFS are as follows:

1. The dual learning system includes the training of a main optimizer, namely the athlete model, and a probabilistic constraint classifier, namely the referee model, is proposed for CO learning. The additional referee model in the proposed system can approximate any type of constraint to guide the optimizer training, which opens the way for constraint learning as guidance of optimization learning with complex constraints.

2. A risk-based constrained loss function is designed to fine-tune the athlete model towards constraint satisfaction. The loss function is the weighted sum of the cost of satisfying and violating the constraints. The weights are factorized by the probabilistic constraint classification output from the referee model and a masking function skewed towards more conservative constraint-satisfying results. By applying the proposed loss function in athlete model training, the CO solution is driven towards a higher likelihood to satisfy the constraints.

3. A metaheuristic solver is used for instance labelling for athlete model training, and concurrently serving as a data sampling resource of constraint learning for the referee model. The intermediate metaheuristic results accumulate the instances to build the dataset to train the referee model, which significantly improves the overall efficiency of training data preparation for the dual learning system.

The proposed ARDLs has been validated through a case study on a practical engineering application namely electric power system emergency control (EPSEC). EPSEC is to decide the optimal stability control strategy when the electricity system enters its emergency state following a large physical disturbance, which is a time-critical optimization task with complicated dynamic security constraints represented in large-scale differential–algebraic equations (DAE) [18]. The test results demonstrate that, as compared to unconstrained training, the proposed ARDLs can significantly enhance the likelihood of satisfying the complex constraints in CO, which verifies the effectiveness of the proposed dual learning system in CO learning.

2. Preliminaries: End-to-end constrained optimization learning

CO refers to the optimization problem having an objective to be minimized and a set of constraints to be satisfied. A CO problem can be generally formulated as follows

$$y = \arg \min_{z} \{ f(x, z) | z \in C \}$$  

where $x$ contains the variables to describe the condition of the optimization environment, $z$ contains the decision variables to be optimized given the environment condition $x$, $f(\cdot)$ is the objective function of CO, and $C$ is the feasible space of $z$ defined by the constraints of CO. $y$ is the optimal solution for this CO problem.

In a generic view, the CO problems under different environmental conditions can be seen as instances to build up a dataset $\psi = \{ x_i, y_i \}_{i=1}^{\infty}$, and then the end-to-end CO can be converted into a machine learning problem as follows

$$\hat{y}_i = G(x_i, \theta_2) | \hat{y}_i \in C_{x_i}, i = 1, 2, \ldots, N$$

where $G(\cdot)$ represents the approximation function of the machine learning model to map the target (i.e. the variables to be optimized) features from the input (i.e. the environment condition variables) features, $\theta_2$ is the parameter vector of model $G(x_i, \theta_2)$, and $\hat{y}_i$ is the approximated output to represent the real-time solution to the CO problem with the input $x_i$. While $\hat{y}_i$ is derived from $G(x_i, \theta_2)$, it also needs to satisfy the constraint $\hat{y}_i \in C_{x_i}$ in the meantime, which makes the machine learning problem converted from CO substantially distinct from the classic supervised learning with unconstrained mapping.

The objective of supervised learning is to find the model parameters that minimize the loss function $L(y, \hat{y})$. However, the classic loss functions based on plain statistical errors treat feasible and infeasible outputs equally, leaving the approximated optimization results with high likelihood of violating the constraints.

Taking into account the constraints in end-to-end optimization learning, Lagrangian relaxations could be used to incorporate the constraints into the loss function, so the original CO can be converted into an unconstrained optimization problem [11–14]. However, these methods require evaluations of the constraints during training, which is only computationally feasible for simple and differentiable constraints. In case of complex constraints that cannot be directly evaluated and/or differentiated, more generalized methodologies would be needed.

3. athlete–referee dual learning system

An ARDLs is proposed in this paper as a universal approach to develop the constraint-aware machine learning model for real-time CO. This dual learning system refers to the training of two machine learning models: an athlete model that is the data-driven optimizer shouldering the overall CO task and a referee model that is a probabilistic classifier shouldering the constraint evaluation and validation tasks. The working principle of ARDLs is to approximate the constraint using the referee model that can be seamlessly incorporated into the constrained loss function of the athlete model. The well-trained athlete model will be ready to be used for real-time CO applications. The overall learning scheme of the proposed ARDLs is presented in Algorithm 1 which is divided into 3 stages: athlete pre-training, referee training, and constraint-oriented athlete fine-tuning. These 3 stages will be elaborated in the sequel.
Algorithm 1: Learning Scheme of ARDLS

Stage 1: Athlete Pre-training

\textbf{input}: \( \psi = \{x_i, y_i\}_{i=1}^N \): athlete training data

\( \alpha_k \): athlete learning rate

\textbf{for epoch} \( k = 0, 1, \ldots \) \textbf{do}

\textbf{foreach} \( \{x_i, y_i\} \leftarrow \text{minibatch} \) \textbf{do}

\( L(y, \hat{y}) \leftarrow \text{MSE}(y, \hat{y}) \): regular regression loss function

\( \theta_A \leftarrow \theta_A - \alpha_k \nabla_{\theta_A}(L(y, \hat{y})) \): gradient descent

\textbf{output}: \( A(x, \theta_A) \): pre-trained athlete model

Stage 2: Referee Training

\textbf{input}: \( \zeta = \{x_j, z_j, c_j\}_{j=1}^M \): referee training data

\( \alpha_k \): referee learning rate

\textbf{for epoch} \( k = 0, 1, \ldots \) \textbf{do}

\textbf{foreach} \( \{x_j, z_j, c_j\} \leftarrow \text{minibatch of size} \ a \) \textbf{do}

\( L(c, p) \leftarrow \text{CrossEntropy}(c, p) \): regular classification loss function

\( \theta_B \leftarrow \theta_B - \alpha_k \nabla_{\theta_B}(L(c, p)) \): gradient descent

\textbf{output}: \( R(x, z, \theta_R) \): referee model

Stage 3: Constraint-oriented Athlete Fine-tuning

\textbf{input}: \( \psi = \{x_i, y_i\}_{i=1}^N \): athlete training data

\textbf{for epoch} \( k = 0, 1, \ldots \) \textbf{do}

\textbf{foreach} \( \{x_i, y_i\} \leftarrow \text{minibatch of size} \ a \) \textbf{do}

\( L_c(x, y, \hat{y}) \leftarrow \text{CLF}(x, y, \hat{y}) \): risk-based constrained loss function

\( \theta_A \leftarrow \theta_A - \alpha_k \nabla_{\theta_A}(L_c(x, y, \hat{y})) \): gradient descent

\textbf{output}: \( A^c(x, \theta_A) \): complete athlete model for CO

3.1. Athlete pre-training stage

This stage is to prepare a preliminary version of the athlete model \( A(x, \theta_A) \) on the training dataset \( \psi = \{x_i, y_i\}_{i=1}^N \). A regular MSE loss function is used at this stage to train the athlete model so that this pre-trained athlete model strives to deliver outputs close to the target optimal solutions that have been constraint validated. However, without strict constraint regulation in the training process, the natural error generated from the approximation may drive the athlete model outputs to infeasible regions. This pre-training aims to prepare an accurate athlete model as a hot start for the subsequent constraint-oriented fine-tuning.

3.2. Referee training stage

In practice, directly incorporating complex constraints into machine learning would result in discontinuity and/or non-differentiability of the solution space. To overcome this drawback, an important innovation in the proposed ARDLS is to train an explicit machine learning model (i.e. referee model) for constraint approximation instead of the direct computation of constraints. Considering the inevitable error in the constraint approximation results, the referee model is selected as a probabilistic classifier for soft approximation of complex constraints. The referee model is presented as follows

\[ p(z_i \in C_{z_i}) = R(x_i, z_i, \theta_k) \] (3)

where \( z_i \) is an arbitrary decision variable, \( p(z_i \in C_{z_i}) \) represents the probability of \( z_i \) satisfying the complex constraints based on environment variables \( x_i \), and \( \theta_k \) is the parameter vector to be trained. Theoretically, there is no limitation on the approach to achieve probabilistic learning as long as \( R(x, z, \theta_k) \) is smoothly modelled using differentiable layers. Some candidate methods for probabilistic learning are softmax layers, Bayesian learning, probabilistic learning are softmax layers, Bayesian learning, pinball loss function [20], and Kullback–Leibler divergence [21]. In this paper, a softmax layer is adopted for simplicity.

This referee model is trained on the dataset \( \zeta = \{x_j, z_j, c_j\}_{j=1}^M \) using regular cross entropy loss function for classification, where \( s_j \) is a binary decision label for constraint validation which indicates the satisfaction/violation of the complex constraints. In this paper, we use \( c_j = 1 \) to represent the satisfaction of constraints, while \( c_j = 0 \) representing the violation of any constraint. Once established, the referee model will be ready for constraint-oriented athlete fine-tuning.

3.3. Constraint-oriented athlete fine-tuning stage

In this stage, the parameters of the pre-trained athlete model are further tuned to adapt to the optimization constraints. This refined athlete model for CO is noted as \( A^c(x, \theta_A) \). This fine-tuning is achieved by a risk-based constrained loss function as follows

\[ L_c(x, y, \hat{y}) = \text{CLF}(x, y, \hat{y}) = \sum_{i=1}^{n} \omega_i (y_i - \hat{y}_i)^2 + (1 - \omega_i)Q_i \] (4)

where \( \omega_i = M(p(y_i \in C_{y_i})) \cdot p(y_i \in C_{y_i}) \)

\[ = M(R(x_i, y_i, \theta_k)) \cdot R(x_i, y_i, \theta_k) \] (5)

where \( M(k) = \frac{1 + \tanh(50 \times (k - d))}{2} \) (6)

In (4)-(6), \( p(y_i \in C_{y_i}) \) is estimated by the referee model, \( Q_i \) is a large number representing the cost of violating any constraint, \( M(\cdot) \) is a masking function to filter out the unreliable estimation of \( p(y_i \in C_{y_i}) \), \( n \) is the number of instances in a mini-batch. The loss on each training instance is represented as the weighted sum of a regular component and a penalty component, where the regular component is the standard MSE loss and penalty component is the cost of violating any constraint. The weight
value depends on the probability of satisfying the constraints and the masking factor $M(p(y_i \in C_k))$. For a probabilistic classifier, we treat the classification output as less confident when the estimated probability value is closer to the midpoint of its feasible range. To implement this confidence-based masking, a parameter $d$ is used in the tanh-based masking function in (6) to control the boundary of the confident output region of the probabilistic classifier. The masking functions $M(k)$ with different $d$ values are shown in Fig. 1. Only the instance with a probability prediction output greater than $d$ is seen as confidently satisfying the optimization constraint. This parameter $d$ should be greater than 0.5 to filter out the unconfident outputs, and thereby derates the contribution of unreliable constraint validation results on the MSE term.

In practice, the set of constraints can be a mixture of different constraints. Since an approximation technique is used in the proposed ARDLS to validate the constraints, only complex constraints that are computationally infeasible to be directly incorporated into the loss function are recommended to be validated by the referee model in (3). For constraints that can be computed straightforwardly, such as linear constraints, they can be directly evaluated in loss function computation during the fine-tuning stage of the athlete model.

3.4. Naming convention

The names of the proposed dual models, i.e. athlete and referee, are inspired by the training of an athlete in sports. Athletes are the persons trained towards their optimal sports performance, who can be seen as an optimizer. In a sports training activity, the athlete aims to achieve optimal performance, but in the meanwhile needs to follow the strict rules confined to the sports game. These rules can be projected as the constraints in optimization. In many cases, it is difficult to accurately judge the satisfaction/violation of the sport’s rules, then a referee is needed to give professional judgement on the sport’s rules during athlete training. Over long-term training with the aid of a referee, the athlete can build a sense of the boundary of the sport’s rules while pursuing optimal sports performance. This process well reflects the interaction of athlete and referee models in the proposed ARDLS for CO training.

4. Dataset preparation

Since there are two supervised learning models (i.e. athlete and referee) involved in the approach to collaboratively implement CO learning, two datasets need to be prepared to train them individually. In this paper, we assume there are no historically labelled instances which is the case in most complex CO applications, so synthetic instance sampling and labelling are required for supervised learning datasets. We also assume that the CO problem can be originally solved by classic solvers, and the aim of using machine learning is to achieve the real-time implementation of CO.

Provided a set of input instances sampled from the feasible input space $X = [x_i]_{i=1}^{N}$, the optimal solutions to the CO problem in (1) need to be derived for instance labelling of the athlete training dataset, whereas the constraint satisfaction condition needs to be validated for each arbitrary pair $[x_i, z_j]$. Considering the natural linkage between the two datasets $Y = [y_i]_{i=1}^{N}$ and $C = [c_j]_{j=1}^{M}$, metaheuristic solvers are recommended to prepare these two datasets, mainly for the following reasons:

(1) In many applications, there may not be a viable deterministic approach to solve the complex CO problem and the true optimal solution is usually unobtainable. In these situations, a metaheuristic solver can be used universally to label the instances.

(2) Metaheuristic solvers are population-based algorithms that populate stochastic decision variable instances in each generation, and constraint validation is an essential step to check the feasibility of each populated solution, which can accumulate a large number of valid randomized instances for referee training. This means the training datasets for referee models can be built in parallel during the metaheuristic labelling of athlete training instances, which tends to significantly improve the overall efficiency of dataset generation for dual learning.

(3) Over the metaheuristic optimization process, the decision variable values tend to be concentrated towards the target optimal solutions. Such instances distributed close to the target values coincide with the distribution of the outputs from the pre-trained athlete model with minimized MSE, which to a large extent avoids domain shift between training and application data of the referee model.

The typical workflow of dataset preparation through metaheuristic optimization is shown in Fig. 2. For the athlete model, the optimal solution derived at the end of the optimization process is used as the true label of each training instance $x_i$. For the referee model, the training instances are accumulated throughout the population process in the metaheuristic algorithm, and the labels of these instances are derived while validating their constraint satisfaction. This constraint validation is an essential step in evaluating the fitness function value of the individual populated decision variable vector $z_j$. Following this process, the input vector of each referee model instance is $[x_i, z_j]$, while the corresponding label is the constraint validation result $c_j$. It can be seen that the training instances for athlete and referee models are collected concurrently in each optimization run. This means metaheuristic optimization provides not only the target optimal solutions, but also a byproduct resource to collect the instances and labels for constraint learning.

5. Case study: Electric power system emergency control

The proposed ARDLS and the associated data-driven CO approach are applied for real-time prediction of the optimal action for EPSEC which is a power engineering optimization application with large-scale complex constraints. An electric power system is a network of electrical components to generate, transmit, distribute and use electricity. EPSEC concerns the situation when
a large disturbance, such as a short circuit fault or loss of a component, takes place and drives the electric power system to an emergency state with potentially unstable propagation, so appropriate emergency control action must be taken to regain system stability. One common type of control action is to reduce customer loads, namely emergency load shedding. However, load reduction will incur costs to the electric utilities as compensation for the unmet electricity demand. Therefore, utilities usually aim to minimize the load shedding cost while making sure the emergency control action is effective to stabilize the system, which results in the following CO problem

$$\text{Min } f(z) = \sum_{b=1}^{B} z_b$$

subject to

$$0 \leq z_b \leq z_{b_{\text{max}}}, b = 1, 2, \ldots, B$$

where $b$ represents each load bus in the system, $z_b$ is the amount of load reduced on bus $b$. Objective (7) is to minimize the total amount of load reduction in the EPSEC action in order to minimize the associated load reduction cost. There are bounding constraints (8) on each $z_b$, where the amount of load reduction must be positive and cannot exceed the upper limit of allowable reduction capacity. These bounding constraints are not a concern because they can be directly modelled in machine learning through output normalization. The complex constraint to be specially considered in this CO problem is the dynamic security constraint (8). It ensures a positive stability index value $\eta (s_T)$ which indicates the system becomes stable after taking the control action. This means the control action is effective to stabilize the system. The $s_T$ represents the final state variable values at the end of the observation window following the control action, which is derived from the following DAEs representing the electric power system dynamics

$$\begin{align*}
\dot{s}_T &= U(s_T, x_T, u_T) \\
0 &= V(s_T, x_T, u_T), \quad t \in [t_0, T]
\end{align*}$$

where $s_T$, $x_T$, and $u_T$ consist of the state variables, system variables, and control variables. To derive the $s_T$ in (8), the DAEs in (9) need to be iteratively solved over the observation window merely based on the knowledge of the initial system condition, the encountered disturbance and the control action. In the power industry, such large-scale DAEs can be solved by running time-domain simulation (TDS) using industrial software such as PSS/E and PSCAD. However, constraint (8) is highly non-linear and the iterative solving process is time-consuming, which means this DAE-based constraint is hard to be directly evaluated and incorporated into the loss function for model training.

Electric power system stability can be checked by the criteria of various electric quantities, including generator rotor angles, bus voltages, and system frequency. In this paper, generator rotor angles are extracted as the state variables to check the stability of the system. The simulated generator rotor angle trajectories in a stable (i.e. $\eta (s_T) > 0$) and an unstable (i.e. $\eta (s_T) < 0$) examples are shown in Fig. 3. It can be seen that the rotor angles of all generators remain close to each other in the stable state, while the rotor angles of some generators start to diverge from other generators in the unstable state, which will lose the synchronization of generators and drive cascading failures of system components. In the worst case of unstable propagation, blackouts in a large area can result.

In the literature, several data-driven approaches have been developed for real-time EPSEC. In [22,23], extreme learning machines are used for EPSEC in terms of different stability criteria. However, the optimization constraint was not specifically considered in model training. Further in [18], the dynamic security constraint was incorporated in deep learning through a linear approximation method, which cannot accurately validate such complex constraints. Therefore the ARDSL proposed in this paper serves as the first approach to universally incorporate complex constraints into optimization learning.
5.1. Dataset generation

Two benchmark electric power systems are selected to generate the datasets and validate the proposed ARDLS. They are the New England 39-bus system and the Nordic 41-bus system. The network structures of the two systems are shown in Fig. 4. For each power system, two different disturbances are considered. Therefore, four independent sets of datasets are generated to validate the proposed ARDLS, and each set consists of an athlete dataset and a referee dataset. The New England and Nordic datasets have 3435 and 4024 instances, respectively. For each instance in the datasets, the load power at each bus is uniformly randomized between 80% and 120% of their rated values, and the other input variables in $x_i$ are calculated using the optimal power flow technique. The label (i.e. the optimal amount of load reduction at each location) of each instance is obtained via metaheuristic optimization with a population size of 100 and maximum iterations of 40. During the metaheuristic optimization, TDSs are run in PSS/E to validate the dynamic security constraints. The result of each TDS run is saved to build up the referee training dataset $\zeta$. An example of population distribution in each optimization iteration and the distribution of the instances accumulated throughout a metaheuristic optimization run are shown in Fig. 5. It can be seen that the instances tend to be concentrated and close to the target values, which demonstrates an excellent distribution to match with the pre-trained athlete model whose output has also been trained to be as close as possible to the target value. This verifies the rationality of using the metaheuristic data to validate the constraint satisfaction of the pre-trained athlete model output for parameter fine-tuning.

5.2. Model training and tuning

In constraint-oriented tuning, $Q_i$ is evaluated as follows to represent the worst-case scenario of losing all the loads due to
Fig. 5. Examples of metaheuristic optimization: (a) evolution of population distribution over iterations for decision variable \( z_1 \), and (b) the decision variables \( z_2 \) vs \( z_1 \) distribution of instances accumulated throughout a complete optimization run.

\[ Q_i = \sum_{b=1}^{B} z_{b,i} \]  

The parameter \( d = 0.8 \) is used. The loss function trajectories of the athlete model trained on dataset #1 are shown in Fig. 6 as an example, where the loss functions at the pre-training and fine-tuning stages are the black and blue curves, respectively. The value of MSE component in the fine-tuning stage is also extracted and plotted as the red curve in the figure. It can be seen that both the MSE loss (black curve) and the constrained loss (blue curve) converge towards the end of both the pre-training and the fine-tuning stages. The MSE minimization at the pre-training stage results in a high value in the constrained loss function at the beginning of the fine-tuning stage, which indicates that the minimum of MSE cannot well represent the satisfaction of constraints. Although the new target of risk-based constrained loss function during the fine-tuning stage slightly drives up the originally minimized MSE value, tuning by this new loss function can successfully enable the constraint awareness of the machine learning model while doing optimization.

Fig. 6. Variations in loss functions in athlete pre-training and fine-tuning stages.

5.3. Learning algorithm selection

In the proposed ARDLS, the athlete model is theoretically open for any learning algorithm with back-propagation training, while the referee model needs to be differentiable to allow feasible training of the athlete model. To select the appropriate learning algorithm for each model, five deep algorithms and two shallow algorithms are compared for the athlete (pre-train) and the referee models, which are fully-connected deep neural network (F-DNN), convolutional neural network (CNN), graph convolutional network (GCN) [24], GraphSAGE [25], long short-term memory (LSTM), support vector machine (SVM) and random forest (RF). The aim of testing shallow algorithms for the athlete model is to investigate the effectiveness of deep learning on the target optimization problem. A differential variant of decision tree [26] is used to construct the RF to ensure its validity to be incorporated in the athlete training process. The testing results are displayed in Tables 1 and 2, where mean absolute percentage error (MAPE) is used as the performance metrics for the pre-trained athlete model while accuracy and F1-score are both used to investigate the classification performance of the referee model. For each dataset, 5-fold cross validation is used. It can be seen that, owing to the ability of embedding the topological graph of the electric power system (as shown in Fig. 4) into the deep learning model, GraphSAGE outperforms other learning algorithms in the pre-trained athlete model. For the referee model, F-DNN shows the highest F1-score in all datasets and the highest accuracy in two datasets, indicating the best overall constraint validation performance among all the test algorithms. Therefore, GraphSAGE and F-DNN are selected as the machine learning algorithms for the athlete and the referee models, respectively. It needs to be clarified that the main aim of this paper is to propose

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>MAPE New England Dataset #1</th>
<th>MAPE New England Dataset #2</th>
<th>MAPE Nordic Dataset #1</th>
<th>MAPE Nordic Dataset #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-DNN</td>
<td>3.63%</td>
<td>3.45%</td>
<td>4.45%</td>
<td>4.38%</td>
</tr>
<tr>
<td>CNN</td>
<td>3.62%</td>
<td>3.50%</td>
<td>4.30%</td>
<td>4.24%</td>
</tr>
<tr>
<td>GCN</td>
<td>3.82%</td>
<td>4.01%</td>
<td>4.85%</td>
<td>5.68%</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>3.44%</td>
<td>3.29%</td>
<td>4.18%</td>
<td>4.19%</td>
</tr>
<tr>
<td>LSTM</td>
<td>3.61%</td>
<td>3.51%</td>
<td>4.29%</td>
<td>4.25%</td>
</tr>
<tr>
<td>SVM</td>
<td>7.37%</td>
<td>7.23%</td>
<td>8.87%</td>
<td>9.07%</td>
</tr>
<tr>
<td>RF</td>
<td>5.61%</td>
<td>4.93%</td>
<td>5.35%</td>
<td>6.14%</td>
</tr>
</tbody>
</table>
a new framework level design for data-driven CO with large-scale complex constraints, and this framework provides the flexibility in choosing the machine learning algorithms according to the target optimization problem. The algorithms we selected here are only for validating the proposed framework, but the optimal design or selection of machine learning algorithms for EPSEC is out of the scope of this paper.

5.4. Model specification

The detailed structure of the athlete model using GraphSAGE algorithm is shown in Fig. 7. It is composed of two GraphSAGE layers followed by 5 fully-connected layers. The input features to the model include the bus voltages, generator output power, and load power, all of which are assembled together in a graph format with buses as the nodes and transmission lines as the connecting edges. 4-hop and 2-hop neighbour aggregations are used in the first and second GraphSAGE layers, respectively. The number of hidden nodes in the fully-connected layers are 256, 256, 128, 32, and 4, respectively.

5.5. Ablation study

After identifying the learning algorithms, the proposed ARDLS is established by using 80% of the instances in the athlete dataset for athlete model training/tuning and all instances in the referee dataset for referee model training. The rest 20% of athlete dataset instances are used for athlete model testing. In the proposed ARDLS, the ability of the athlete model to satisfy complex constraints is gained at the fine-tuning stage through (1) the incorporation of referee model and (2) the masking function $M(\cdot)$ to filter out the unreliable referee model outputs. To investigate their effectiveness, an ablation study is conducted on 3 models: the pre-trained athlete model (PAM), the fully-trained athlete model without masking (FAM), and the fully-trained athlete model with masking (FAM-M). The FAM-M represents the final data-driven optimizer developed by the proposed approach. The aim of comparing these three models is to (1) validate the effectiveness of constrained-oriented fine-tuning by comparing FAM with PAM, and (2) validate the effectiveness of the output masking in the proposed risk-based constrained loss function by comparing FAM-M with FAM.

The testing results of the 3 models over the 4 datasets are shown in Table 3. Two performance indices, MAPE and constraint violation rate (CVR) are used to demonstrate the performance of CO. The CVR is defined as the percentage of testing instances with $\hat{y}$ violating the complex constraint (8). It can be seen from Table 3 that PAM results in the lowest MAPE but very poor CVR, with an average 13.47% of the optimization solutions violating the constraints. This is because the PAM is trained by the regular loss function only to pursue the lowest prediction error but largely ignoring the satisfaction of constraints. In comparison, FAM and FAM-M achieve significantly lower CVR (maximum 6.17% solutions violating the constraints) with just a very small amount of increase in MAPE, demonstrating the effectiveness of the constrained-oriented fine-tuning using the referee model. Moreover, by comparing the FAM and FAM-M, it can be seen that incorporating the masking function successfully reduces the average CVR from 5.80% further down to 3.94% with little increase in MAPE, showing the effectiveness of the referee model output masking in driving the athlete model to be more constraint-oriented without significant impairment on optimality.

5.6. Computational efficiency

The main aim of using machine learning techniques for optimization is to achieve real-time optimization capability. Considering this, we have compared the computational efficiency of the proposed ARDLS with a classic metaheuristic solver on the same testing datasets. The average computation time of the two methods to optimize each instance is displayed in Table 4, where the proposed ARDLS significantly outperforms the metaheuristic solver and can complete an optimization task within a millisecond. These results can clearly verify the real-time optimization capability of the proposed ARDLS.
Data availability

The data that has been used is confidential.

References


