A federated data-driven evolutionary algorithm for expensive multi-/many-objective optimization

Jinjin Xu^{1,2} · Yaochu Jin² · Wenli Du¹

Received: 21 June 2021 / Accepted: 11 August 2021 / Published online: 24 August 2021
© The Author(s) 2021

Abstract

Data-driven optimization has found many successful applications in the real world and received increased attention in the field of evolutionary optimization. Most existing algorithms assume that the data used for optimization are always available on a central server for construction of surrogates. This assumption, however, may fail to hold when the data must be collected in a distributed way and are subject to privacy restrictions. This paper aims to propose a federated data-driven evolutionary multi-/many-objective optimization algorithm. To this end, we leverage federated learning for surrogate construction so that multiple clients collaboratively train a radial-basis-function-network as the global surrogate. Then a new federated acquisition function is proposed for the central server to approximate the objective values using the global surrogate and estimate the uncertainty level of the approximated objective values based on the local models. The performance of the proposed algorithm is verified on a series of multi-/many-objective benchmark problems by comparing it with two state-of-the-art surrogate-assisted multi-objective evolutionary algorithms.

Keywords Federated learning · Multi-/many-objective optimization · Bayesian optimization · Surrogate · Data-driven evolutionary optimization

Introduction

Many optimization problems have several conflicting objectives that need to be optimized concurrently, which are known as multi-objective optimization [11], and many-objective optimization when the number of objectives is larger than three [28]. Over the past decades, a pile of successful multi-objective evolutionary algorithms (MOEAs) have been developed, which can largely be categorized into decomposition-based MOEAs, such as the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [53] and reference vector guided evolutionary algorithm (RVEA) [6]; Pareto-dominance based, e.g., the elitist non-

dominated sorting genetic algorithm (NSGA-II) [12]; and performance indicator-based [14]. Note that many MOEAs combine decomposition or Pareto dominance with other criteria to address various challenges in multi-objective optimization [55].

It is well recognized, however, that most MOEAs require a large number of function evaluations before a set of diverse and well-converged non-dominated solutions can be found, which makes it hard for them to be directly applied to solve a class of data-driven optimization problems [24], whose objectives can be evaluated by means of conducting time-consuming computer simulations or costly physical experiments. Examples of real-world data-driven multi-objective optimization problems include blast furnace optimization [7], air intake ventilation system optimization [10], airfoil design optimization [29], bridge design optimization [34], design optimization for a stiffened cylindrical shell with variable ribs [57], optimization of operation of crude oil distillation units [19], resource allocation of trauma systems [47], and also neural architecture search [42,58], among many others.

Data-driven surrogate-assisted evolutionary algorithms were first developed for single-objective optimization [23].

✉ Yaochu Jin
yaochu.jin@surrey.ac.uk

Jinjin Xu
jin.xu@mail.ecust.edu.cn

¹ Key laboratory of Advanced Control and Optimization for Chemical Processess, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

² Department of Computer Science, University of Surrey, Guildford, GU2 7XH, UK

In recent years, more attention has been paid to data-driven surrogate-assisted multi-objective optimization [2,9]. Similar to MOEAs, surrogate-assisted MOEAs can also be divided into four groups. The first group is based on decomposition, where surrogates are built to approximate the sub-problems formulated by the Tchebycheff function using random weights, known as ParEGO [27]. Alternatively, surrogates can also be constructed for each individual objective function to assist a decomposition-based MOEA, such as the Gaussian process assisted MOEA/D, called MOEA/D-EGO [54], K-RVEA [8], a Kriging-assisted RVEA, and surrogate-ensemble-assisted MOEA, called SAEMO [56]. The second group employs the non-dominance relation for environmental selection, and surrogates are adopted for approximating the objective functions [18,46], or in local search [31]. In Ref. [32,35], a surrogate is built for estimating the dominance relationship, and consequently only a single surrogate is required. The third category relies on a performance indicator-based MOEA, e.g., a Gaussian process-assisted hypervolume-based MOEA [15], and an enhanced inverted generational distance-based MOEA assisted by a dropout deep neural network (EDN-ARMOEA) [19]. Finally, surrogates have also been applied to assist other MOEAs, such as the surrogate-assisted two-archive evolutionary algorithm [40]. A good overview of data-driven evolutionary optimization can be found in a recently published book [25].

Despite its impressive success, data-driven evolutionary optimization is faced with several challenges. One is the lack of data, which is very expensive to collect. To address this issue, advanced learning techniques such as semi-supervised learning [17,21,41], knowledge transfer between different surrogates using transfer optimization [20,52], and between different objectives using transfer learning [48,49] have been adopted in surrogate-assisted optimization. Another solution is to reduce the computational cost of data collection by parallelizing expensive computer simulations [1,5] in data-driven surrogate-assisted evolutionary optimization.

Little attention has been paid to data privacy concerns in data-driven optimization. One exception is our previous work reported in Ref. [51], which proposes a federated data-driven evolutionary algorithm (FDD-EA) for single-objective optimization based on federated learning [33,60]. As its name suggests, federated learning aims to collaboratively train a global model using data distributed on multiple local devices without transmitting the local data to a central server. Instead, each client train a local model based on the local data and then the parameters (or the gradients) of the local models are transmitted to the server to generate the global model. Similarly, in FDD-EA, the central server averages the local models uploaded by the clients with a sorted averaging algorithm [51]. Then, the global and local models are made use of for approximating the objective function and estimating the uncertainty of the approximated fitness, based on which an

infill criterion is designed for surrogate management in the federated optimization environment. Federated data-driven optimization has a wide range real-world applications, in such as the design of chemical reactors [36] and the optimization of engineering operation [39], where the entire system contains a large number of sub-systems and the collected data on each sub-system are not allowed to be sent to a central storage. By extending FDD-EA, this work proposes a federated data-driven evolutionary algorithm for multi-objective optimization. The main contributions of this paper are summarized as follows.

- A federated data-driven multi-/many-objective evolutionary optimization algorithm (FDD-MOEA) is presented, in which multiple clients collaboratively train a global surrogate without uploading local data. To the best of our knowledge, this is the first work that enables federated data-driven multi-/many-objective optimization that does not require to store the local data on a central server, thereby preserving the privacy of the data.
- An ensemble-based acquisition function is adopted for FDD-MOEA, in which the global surrogate is used to provide the approximated fitness and the local surrogates are used to provide an estimate of the uncertainty of the approximated fitness.

The remainder of this paper is arranged as follows. The next section presents the concepts of multi-objective optimization, federated learning and acquisition function. The main workflow and mechanisms of the proposed method, FDD-MOEA, are illustrated in the subsequent section. Then, various empirical studies on test suites are given. The final section summarizes the contributions and future directions of this paper.

Background and motivations

In this part, we briefly illustrate the concept of multi-objective optimization, RBFN surrogate, federated learning and acquisition function.

Multi-objective optimization

Denote the minimization of vector of objectives by

$$\min_x \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots, f_M(\mathbf{x}))^T, \quad (1)$$

subject to $\mathbf{x} \in R^d$,

where $\mathbf{x} \in R^d$ is the decision vector with dimension d , M is the number of objectives and $M \geq 2$, and the profile formed

by the projections of the aforementioned Pareto optimal solutions in the objective space is called Pareto front (PF). Note that this work focuses on the optimization of unconstrained MOPs.

Radial basis function networks

Radial basis function networks (RBFNs) [13] are widely adopted surrogate models due to their high computational efficiency and robustness on high-dimensional problems. The first hidden layer connected to the input of an RBFN is composed by a series of centers $\{c_1, \dots, c_i, \dots, c_q\}$, where q is the number of centers, and $c_i \in R^d$ is the i th center. This layer calculates the distance between a sample point $x \in R^d$ and the i th center, $\phi(\|x - c_i\|)$, where $\phi(\cdot)$ represents a radial basis function, and in this work we select the Gaussian function with a spread δ_i as the basis function,

$$\phi(\|x - c_i\|) = e^{-\frac{\|x - c_i\|^2}{2\delta_i^2}}. \quad (2)$$

Suppose $X \in R^{n_k \times d}$ and $Y \in R^{n_k \times M}$ represent a dataset with n_k (indexing by j) sample pairs ($x_j \in R^d$, $y_j \in R^M$), then the corresponding output of the RBFN can be calculated by

$$\begin{aligned} & \begin{bmatrix} \hat{y}_{1,1} & \dots & \hat{y}_{1,M} \\ \vdots & & \vdots \\ \hat{y}_{n_k,1} & \dots & \hat{y}_{n_k,M} \end{bmatrix} \\ &= \begin{bmatrix} \phi(\|x_1 - c_1\|) & \dots & \phi(\|x_1 - c_q\|) \\ \vdots & & \vdots \\ \phi(\|x_{n_k} - c_1\|) & \dots & \phi(\|x_{n_k} - c_q\|) \end{bmatrix} \\ & \begin{bmatrix} w_{1,1} & \dots & w_{1,M} \\ \vdots & & \vdots \\ w_{q,1} & \dots & w_{q,M} \end{bmatrix} + [b_1 \dots b_M], \end{aligned} \quad (3)$$

which can further be simplified to be

$$\hat{y}_j = \Phi_j w + b, \text{ for } j = 1, 2, \dots, n_k, \quad (4)$$

where $\Phi_j = [\phi(\|x_j - c_1\|), \dots, \phi(\|x_j - c_q\|)]$ denotes the distance between sample x_j and all centers, and $w = [w_1, \dots, w_M] \in R^{q \times M}$, in which $w_M = [w_{1,M}, w_{2,M}, \dots, w_{q,M}]^T$ are the weights of the connecting M th attribute of y_j and all centers, and $b = [b_1, \dots, b_M] \in R^{1 \times M}$ is the bias, $\delta = \{\delta_1, \delta_2, \dots, \delta_q\} \in R^q$ stands for the spread vector of the RBFN. Hence, we can use $[w, \delta, c, b]$ to represent the RBFN.

In terms of the training process, the centers of an RBFN is usually determined by random selection or K-means clustering. The least square method and backpropagation are two most popular methods [13] to train the weights w and

biases b . The least square method optimizes the weights and the biases through matrix inversion (linear algebra), whereas backpropagation uses strategies such as the stochastic gradient descent (SGD) method [37] to iteratively optimize the parameters.

Compared with GPs, RBFNs are efficient for training and it is less sensitive to the increase of training data size. However, an RBFN cannot directly provide the uncertainty of its predictions like GPs, making it unsuitable for applying an infill criterion for surrogate model management.

Federated learning

Federated learning [33] originates from the attempt to address the privacy concerns in distributed learning [30], which has been widely studied and applied in real-world applications due to its capability of privacy protection and parallel computing [50,60]. Generally, the global optimization objective of a federated learning system can be written by

$$\min_{W_s} F(W_s) = \sum_{k=1}^{\lambda N} p_k F_k(W_k), \quad (5)$$

where N is the total number of clients, λ is a coefficient that reflects how many clients will participate in model update in each communication round, and p_k is the weight that controls the importance of the participating client k , which is usually determined by the size of local dataset D_k ,

$$p_k = \frac{|D_k|}{\sum_{k=1}^{\lambda N} |D_k|}, \quad (6)$$

and F_k denotes the local objective, which is

$$F_k(W_k) = \sum_{i=1}^{n_k} L(W_k; x_i, y_i), \quad (7)$$

where L is the loss function.

As McMahan [33] suggested, the global model W_s can be conducted by averaging a set of local models W_k ($k = \{1, 2, \dots, \lambda N\}$) rather than collecting distributed local data, which can be denoted by

$$W_s = \sum_{k=1}^{\lambda N} p_k W_k. \quad (8)$$

To summarize, federated learning is a distributed learning framework, in which a local model is trained on each client. The trained local models are then uploaded onto the server and aggregated into a global model. The global model is then downloaded to all participating clients to be further trained.

Ideally, the global model will converge to the one that is trained on all local data that are centrally stored.

It should be emphasized that the main motivation of the present work is to achieve data-driven optimization of complex systems, in which the data is collected and distributed on multiple clients. Due to privacy concerns and other considerations, the data are not allowed to be transmitted to a central server. To the best of our knowledge, all existing work on data-driven evolutionary optimization assumes that the sampled data are available for training a global surrogate model using centralized learning, with one exception reported very recently in Ref. [51], which proposed, for the first time, an algorithm for federated data-driven single-objective optimization. Actually, this work is a natural extension of [51], aiming to propose a federated data-driven evolutionary multi-objective optimization algorithm, FDD-MOEA for short.

Acquisition function

Acquisition function (AF), also known as infill criterion or model management, is designed to decide where to sample new data (new candidate solutions) in the decision space, i.e., to determine which candidate solutions $\mathbf{x}$ should be selected for calculating their objective values $\mathbf{f}(\mathbf{x})$, usually by means of time-consuming numerical simulations or physical experiments. AFs were originally developed in Bayesian optimization [38], where the Gaussian process is used as the surrogate, although it can be easily extended to any other types of surrogates capable of providing uncertainty information, such as ensemble surrogates [18], or dropout neural networks [19].

An AF must be able to balance the exploitation and exploration [26], and several successful infill criteria have been widely adopted by surrogate-assisted MOEAs, such as Expectation Improvement (EI) [26] and Lower Confidence Bound (LCB) [44]. In this work, we choose LCB to minimize Eq. (9), which is

$$LCB(\mathbf{x}) = \hat{\mathbf{f}}(\mathbf{x}) - \alpha \hat{\mathbf{s}}(\mathbf{x}), \quad (9)$$

where α is a hyperparameter which is set to 2 as suggested in [16] in the experimental studies.

Federated data-driven multi-objective optimization

In this section, we will briefly introduce the federated construction of the global model, the workflow of the proposed FDD-MOEA, and the used federated acquisition function.

Federated surrogate construction

Clients: Suppose $W_k = [\mathbf{w}_k, \mathbf{c}_k, \delta_k, \mathbf{b}_k]$ represents the surrogate on the k^{th} client, and let $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_B\}$ represent the predicted outputs with a batch size of B ($1 \leq B \leq n_k$), we have

$$\hat{\mathbf{y}}_i = W_k(\mathbf{x}_i) = \Phi_i \mathbf{w} + \mathbf{b}, \text{ for } i = 1, 2, \dots, B \quad (10)$$

then the loss function (L in Eq. (7)) of each local RBFN used in this work can be written as

$$L = \frac{1}{2B} \sum_{i=1}^B (\hat{\mathbf{y}}_i - \mathbf{y})^2, \quad (11)$$

and the centers and spreads of the RBFN are determined by K-means maximum distance between centers, then the gradients of weights $\mathbf{w}_k$ and $\mathbf{b}_k$ are calculated by

$$\frac{\partial L}{\partial \mathbf{w}_k} = \frac{1}{B} \sum_{i=1}^B \frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{w}_k} (\hat{\mathbf{y}}_i - \mathbf{y}), \quad (12)$$

$$\frac{\partial L}{\partial \mathbf{b}_k} = \frac{1}{B} \sum_{i=1}^B \frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{b}_k} (\hat{\mathbf{y}}_i - \mathbf{y}). \quad (13)$$

By replacing $\hat{\mathbf{y}}_i$ with Eq. (10), we have

$$\frac{\partial L}{\partial \mathbf{w}_k} = \frac{1}{B} \sum_{i=1}^B \Phi_i (\hat{\mathbf{y}}_i - \mathbf{y}), \quad (14)$$

$$\frac{\partial L}{\partial \mathbf{b}_k} = \frac{1}{B} \sum_{i=1}^B (\hat{\mathbf{y}}_i - \mathbf{y}). \quad (15)$$

In this way, each client iteratively trains the weights and biases of the local W_k with a learning rate η and a local epoch E . In addition, the batch size B is set to 1 in training the local models in this work, mainly because the number of training data on each client is very limited in data-driven optimization.

Server: After the local model on the k th client (W_k) is obtained and uploaded to the server, the sorted averaging algorithm, as described in Algorithm 1 and first reported in Ref. [51], will be adopted to aggregate all received local models to create the global model W_s . Then, the server will use W_s to assist an MOEA. Finally, W_s and the found potential solution $\mathbf{x}$ to the chosen clients, the k th client will evaluate the $\mathbf{x}$ by the expensive objectives and update the local dataset to further update local model W_k , which can be illustrated by Fig. 1. The overall framework will be described in “Overall framework”.

Fig. 1 An illustration of local training with server model **Algorithm 1: Sorted Averaging [51]**

Input : Local surrogates $W_k = [\mathbf{w}_k, \mathbf{c}_k, \delta_k, \mathbf{b}_k]$, global surrogate W_s .

foreach $k = 1, 2, 3 \dots \lambda N$ **do**

index $\leftarrow$ Sort according to the distance of each center in $\mathbf{c}_k$ from the origin

$\mathbf{c}_k, \mathbf{w}_k, \delta_k \leftarrow \mathbf{c}_k[\text{index}], \mathbf{w}_k[\text{index}], \delta_k[\text{index}]$

$W_k = [\mathbf{c}_k; \mathbf{w}_k; \delta_k; \mathbf{b}_k]$

end

$W_s \leftarrow \sum_{k=1}^{\lambda N} p_k W_k$

Output: Server model W_s

Overall framework

As shown in Fig. 2, the proposed FDD-MOEA differs from those of ensemble-based methods in that it utilizes sorted model averaging to obtain a global model W_s for fitness approximation instead of averaging predictions of ensemble. In addition, the data partitions for each local models are determined by the operating conditions of the clients and cannot be actively manipulated as done conventional ensemble construction when the data is centrally stored. It also be stress that in principal, any MOEAs can be applied to the proposed framework. In our experimental studies, we choose NSGA-II and RVEA to examine the performance of the proposed framework for MOPs and MaOPs, respectively.

Algorithm 2 summarizes the overall workflow of the proposed FDD-MOEA. Before the optimization starts, Latin hypercube sampling (LHS) is adopted to generate $11d - 1$ points in the decision space, where d is the dimension of the decision space. And these initial points are evaluated by the expensive real objective functions (data collection), the collected data pairs are used as the initial training dataset of for the local models. Then, the k th client will train the local RBFN W_k with its own dataset and upload the trained W_k to the server. The server aggregates all uploaded local models by means of sorted averaging to obtain the global model W_s . After that, the acquisition function (AF) will be formed using W_s and all local models (refer to “Modified LCB”), in which the averaged W_s will replace the expensive objective func-

tions to predict the objective values, and the local models will collaboratively estimate the uncertainty of the predictions.

Algorithm 2: Federated data-driven MOEA.

Input : Dimension of the decision space d ; Expensive real objective functions f ; Maximum iterations R ; Maximum number of training samples l ; Number of clients N ; Number of selected samples per round m ; Failure probability of the communication p_f .

Output: Solutions X and their objective values Y

$X \leftarrow$ Latin hypercube sampling $11d - 1$ points

for $i = 1$ to $11d - 1$ **do**

$Y[i] \leftarrow f(X[i])$

end

for $r = 1, 2, 3 \dots R$ **do**

for Client = 1, 2, 3 ... λN in parallel **do**

// communication succeeded $(1 - p_f)$

evaluate Y_{new} by f

$X, Y \leftarrow X \cup X_{new}, Y \cup Y_{new}$

if $|X| > l$ **then**

$X_k, Y_k \leftarrow \text{select}(X, Y, l)$ // fast non-dominated sorting

end

$W_k \leftarrow$ local training (X_k, Y_k)

end

Server does:

$W_s \leftarrow$ Algorithm 1

AF $\leftarrow$ Modified LCB (Section 3.3)

solutions $\leftarrow$ MOEA(AF)

$X_{new} \leftarrow$ select m new solutions

end

end

Then, an MOEA is chosen to optimize the acquisition functions, and m new solutions (specified by the user) will be selected from the results of MOEA through clustering. Note, however, that duplicate solutions will be removed before clustering.

- (1) Removal of redundant solutions. It is very common when the optimization completes, the final popula-

Fig. 2 The overall framework of the proposed FDD-MOEA

tion of the MOEA contains redundant solutions or extremely similar solutions. In this work, we calculate the Euclidean distances between solutions of the current population in the decision space and only one will be kept if the distance between two solutions is less than a threshold (10^{-6} in this work). Note that the number of solutions after removing the redundant solutions may be smaller than m , in particular for high-dimension problems. In this case, we will save the existing solutions and restart the MOEA until the stopping criterion is satisfied.

- (2) Clustering. After removing the redundant solutions, the remaining ones will be clustered using the K -means algorithm with a center size of m .

In federated optimization, there are many challenges that may lead to difficulties in collecting well distributed data, making it hard to achieve a high-quality global model. For example, in this work, the m selected candidate solutions in each round will be broadcast to all participating clients and each will evaluate all the m samples. As a result, the total FEs will be λNm (assuming no communication failures), which is λN times more than in centralized optimization. Note, however, that in federated optimization, the evaluations on different clients are done in parallel, and these additional FEs do not introduce any new data samples compared to the centralized ones. This ensures fair performance comparisons between the federated and centralized optimization. In principle, we can send each of the m solutions to m different clients, in which case the total number of FEs will be m in one round. This will result in two differences. First, the computation time of each client will be approximately $1/m$ compared to the centralized paradigms. Second, the samples on different clients will become increasingly different as the optimization proceeds, which will result in non-independent and identically distributed (non-iid) data, making the training of a global model challenging [59].

In this work, we consider two main challenges, namely communication failures and client dropout. We define a probability of communication failure p_f to denote whether the communication between a certain client and the server is successful in one particular iteration of surrogate management. If the communication is successful, then all m candidate solutions selected by the server will be passed to the participating clients, evaluated, and added to the local training dataset for the next round training. Otherwise, the m new solutions will not be evaluated by that particular client. Consequently, the larger the communication failure probability, the stronger the data will be non-iid.

Client dropout is also caused by communication problems. Like in federated learning, we use a participation ratio λ to define the percentage of clients that participate in each round of model update and surrogate management. The smaller participation ratio, the smaller the number of local models, which may result in a poorer global model.

To be consistent with the settings in Ref. [18], the initial number of training data is set to $11d - 1$, and we also limit the maximum number of training data to be $l = 11d - 1 + 25$. A fast non-dominated sorting is applied to select l data for model training.

Modified LCB

As we stated in “Acquisition function”, the acquisition function prefers solutions with high uncertainty and better performance. To achieve this, we utilize the global surrogate model together with the local models to provide predictions of candidate solutions and estimate the uncertainty of the predictions.

First, once the server model W_s is aggregated, the evaluated objective values $\hat{f}(x)$ in Eq. (9) is calculated by

$$\hat{f}(x) = W_s(x). \quad (16)$$

Table 1 Test instances and their properties

Problem	M ($d = 20$)	d ($M = 3$)	Domain	Pareto front
DTLZ1	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Linear
DTLZ2	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Concave
DTLZ3	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Concave, biased
DTLZ4	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Concave, biased
DTLZ5	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Degenerate
DTLZ6	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Degenerate
DTLZ7	3, 5, 10, 20	10, 20, 40, 80	[0, 1]	Disconnected

Table 2 Mean (std) of the IGD values obtained by FDD-MOEA, HeE-MOEA, GP-MOEA and K-RVEA for different decision dimensions when $M = 3$

Problems	d	FDD-MOEA	HeE-MOEA	GP-MOEA	K-RVEA
DTLZ1	10	101.71 (2.0e + 01)	111.95 (2.2e + 01) =	116.18 (1.4e + 01) =	92.962 (2.2e + 01) =
	20	327.79 (3.8e + 01)	356.33 (4.3e + 01) =	363.78 (3.5e + 01) +	330.06 (4.7e + 01) =
	40	864.74 (5.5e + 01)	900.74 (5.4e + 01) +	896.17 (4.0e + 01) +	887.47 (4.4e + 01) =
	80	1949.2 (8.5e + 01)	2078.1 (6.5e + 01) +	1991.7 (6.9e + 01) =	2028.0 (7.9e + 01) +
DTLZ2	10	0.1738 (1.9e−02)	0.1958 (2.0e−02) +	0.2028 (7.1e−02) +	0.1829 (3.6e−02) =
	20	0.2969 (3.1e−02)	0.3261 (2.1e−02) +	0.5353 (6.1e−02) +	0.6032 (7.5e−02) +
	40	0.4214 (2.7e−02)	0.7913 (6.6e−02) +	0.9432 (1.0e−01) +	1.7960 (1.3e−01) +
	80	0.6889 (3.2e−01)	2.1337 (3.1e−01) +	1.9965 (1.7e−01) +	4.6585 (2.1e−01) +
DTLZ3	10	285.16 (5.7e + 01)	315.84 (6.3e + 01) =	330.81 (5.7e + 01) +	236.74 (4.3e + 01) −
	20	1056.8 (7.6e + 01)	1073.5 (1.2e + 02) =	1094.8 (9.1e + 01) +	941.95 (1.3e + 02) −
	40	2734.7 (1.4e + 02)	2807.9 (1.0e + 02) +	2810.3 (1.6e + 02) +	2776.3 (1.7e + 02) =
	80	6323.1 (3.4e + 02)	6363.2 (2.3e + 02) =	6366.2 (3.1e + 02) =	6382.9 (1.9e + 02) =
DTLZ4	10	0.7199 (9.5e−02)	0.8060 (1.1e−01) =	0.6583 (7.6e−02) −	0.4139 (1.3e−01) −
	20	0.9331 (2.2e−01)	0.9614 (2.5e−02) +	1.0231 (1.1e−01) +	0.8715 (1.4e−01) =
	40	0.9823 (1.1e−02)	1.2422 (4.5e−02) +	1.9227 (1.4e−01) +	2.4490 (1.0e−01) +
	80	1.3043 (4.8e−02)	2.5379 (1.4e−01) +	2.7526 (6.2e−02) +	5.0597 (1.5e−01) +
DTLZ5	10	0.0604 (1.3e−02)	0.1125 (2.0e−02) +	0.1466 (2.6e−02) +	0.1067 (3.6e−02) +
	20	0.1528 (2.7e−02)	0.2060 (3.5e−02) +	0.4607 (1.0e−01) +	0.4018 (8.0e−02) +
	40	0.2643 (3.1e−02)	0.6681 (4.8e−02) +	0.9121 (1.4e−01) +	1.6052 (1.4e−01) +
	80	0.5754 (3.2e−02)	1.9576 (1.7e−01) +	1.7653 (2.0e−01) +	4.5680 (2.5e−01) +
DTLZ6	10	6.7366 (7.2e−02)	5.8677 (4.3e−01) −	5.7072 (3.2e−01) −	3.2101 (4.9e−01) −
	20	15.394 (1.4e−01)	15.037 (4.0e−01) −	13.158 (3.7e−01) −	11.016 (9.6e−01) −
	40	32.312 (3.4e−01)	32.561 (6.8e−01) =	32.194 (6.8e−01) =	30.688 (7.6e−01) −
	80	66.721 (2.3e−01)	67.808 (7.4e−01) =	68.032 (6.2e−01) +	67.980 (7.1e−01) =
DTLZ7	10	5.9812 (9.8e−01)	1.8227 (6.3e−01) −	0.9151 (3.2e−01) −	0.1570 (5.7e−02) −
	20	7.9733 (7.1e−01)	4.4650 (7.6e−01) −	4.3612 (7.1e−01) −	0.3631 (1.7e−01) −
	40	9.3081 (9.4e−01)	6.5439 (7.2e−01) −	5.8972 (1.0e + 00) −	6.4692 (2.0e + 00) −
	80	9.7757 (3.3e−01)	8.7749 (3.5e−01) −	9.8455 (2.3e−01) =	9.5087 (4.4e−01) =
+/-/=	−	14/6/8	17/5/6	10/9/9

Fig. 3 The IGD convergence profiles on DTLZ1

Fig. 4 The IGD convergence profiles on DTLZ2

Fig. 5 The IGD convergence profiles on DTLZ3

Fig. 6 The IGD convergence profiles on DTLZ4

Fig. 7 The IGD convergence profiles on DTLZ5

Fig. 8 The IGD convergence profiles on DTLZ6

Fig. 9 The IGD convergence profiles on DTLZ7

Similar to the ensemble-based strategy, the variance $\hat{s}^2(\mathbf{x})$ of the predictions can be estimated by

$$\hat{s}^2(\mathbf{x}) = \frac{1}{\lambda N - 1} \left[\sum_k^{\lambda N} \left(W_k(\mathbf{x}) - \hat{f}(\mathbf{x}) \right)^2 \right]. \quad (17)$$

Consequently, we can easily calculate the LCB values for optimization using the MOEA.

Numerical studies

In the following, the benchmark problems and parameter settings used in the empirical studies are introduced and then the comparative results are presented.

Benchmark problems

As listed in Table 1, we adopt the DTLZ test suite for comparative studies, and 49 test instances in total with different number of decision variables and objectives are used. According to Ref. [22], an MOEA may get trapped in local Pareto front of DTLZ1 and DTLZ3, which will lead to poor performance. And the PF of DTLZ2 are located in an octant of the unit sphere, which is often used to test the performance of an MOEA with a large number of objectives. It is hard for an MOEA to find evenly distributed solutions in the objective space for DTLZ4 due to the biased projection from decision space to the objective space [18], and the mapping parameter α of DTLZ4 is set to 100 in this work. DTLZ5 and DTLZ6 have degenerate Pareto optimal fronts, and each front is supposed to be an arc in the objective space. The Pareto front of DTLZ7 is disconnected.

Parameter settings

Recall that the number of objectives and the dimension of the decision variable are denoted by M and d , respectively. The basic parameter settings are as follows.

1. The number of independent runs is set to 20 for each test instance, and each time we sample $11d - 1$ points in the decision space using LHS.
2. NSGA-II is chosen as the MOEA for MOPs, the population size and the number of generations are both set to 50. The crossover distribution index and its probability are set to 20, 1.0. And the corresponding mutation distribution index and probability are 20 and $1/d$.
3. RVEA is adopted as the MOEA for MaOPs, and the number of reference vectors are set to 105, 126, 275, 420 for 3, 5, 10, 20 objectives, respectively.
4. As suggested in Ref. [18], the number of RBFN centers is limited to $\sqrt{M+d} + 3$, the optimization process starts at $11d - 1$ fitness evaluations (FEs) and ends at $11d - 1 + 120$ FEs, and five new FEs are performed (five new training samples are collected) in each round, which means that the number of iteration R is 24 in a single run. The maximum size of training data l is set to $11d - 1 + 25$.
5. The hyperparameters of federated learning are set as follows: the local epoch number $E = 20$, the learning rate $\eta = 0.06$, the number of clients $N = 10$, and the participation ratio $\lambda = 0.9$. As previously discussed, for each individual client, the solutions to be sampled at each iteration as identified by the surrogate management strategy by the server may not be implemented on each client with a communication failure probability $p_f = 3\%$. To make a fair comparison with ensemble-based algorithms, the initial solutions are the same on all clients.

Table 3 Mean (std) of the IGD values obtained by FDD-MOEA, HeE-MOEA, GP-MOEA and K-RVEA for different numbers of objectives when $d = 20$

Problems	M	FDD-MOEA	HeE-MOEA	GP-MOEA	K-RVEA
DTLZ1	5	212.97 (4.0e + 01)	281.33 (2.4e + 01) +	291.38 (2.7e + 01) +	236.16 (32.e + 01) +
	10	130.24 (2.9e + 01)	159.46 (2.2e + 01) +	155.67 (2.5e + 01) +	110.32 (2.6e + 01) –
	20	0.4324 (7.9e–02)	0.4335 (7.8e–02) =	0.5084 (1.2e–01) =	0.4398 (1.1e–01) =
DTLZ2	5	0.6704 (4.0e–02)	0.7940 (6.0e–02) +	0.9390 (5.9e–02) +	0.7543 (5.7e–02) +
	10	0.9413 (2.1e–02)	0.9172 (3.5e–02) –	1.0124 (4.3e–02) +	0.7958 (5.5e–02) –
	20	1.0057 (1.4e–02)	1.0043 (1.4e–02) =	1.0103 (1.5e–02) =	0.9059 (3.5e–02) –
DTLZ3	5	667.77 (9.6e + 01)	939.50 (6.3e + 01) +	909.28 (7.1e + 01) +	799.77 (7.7e + 01) +
	10	380.93 (7.9e + 01)	538.91 (7.5e + 01) +	548.25 (9.0e + 01) +	402.85 (6.0e + 01) =
	20	1.4774 (2.1e–01)	1.4482 (2.2e–01) =	1.5485 (2.6e–01) =	1.3590 (1.6e–01) –
DTLZ4	5	1.1841 (5.4e–02)	1.1077 (1.9e–02) –	1.3370 (6.7e–02) +	1.0117 (1.1e–01) –
	10	1.0113 (4.3e–02)	1.1466 (3.1e–02) =	1.2009 (4.2e–02) +	1.0561 (7.3e–02) +
	20	0.7874 (9.1e–03)	0.8248 (1.1e–02) +	0.8276 (9.8e–03) +	0.7943 (1.7e–02) +
DTLZ5	5	0.3709 (3.4e–02)	0.2099 (4.0e–02) –	0.6635 (8.4e–02) +	0.4276 (7.9e–02) +
	10	0.2654 (2.5e–02)	0.1793 (3.2e–02) –	0.4124 (5.6e–02) +	0.2534 (4.4e–02) =
	20	0.0109 (1.1e–03)	0.0118 (1.1e–03) +	0.0126 (1.3e–03) +	0.0118 (1.2e–03) +
DTLZ6	5	8.4899 (1.2e + 00)	13.100 (5.4e–01) +	12.627 (3.7e–01) +	9.9443 (7.6e–01) +
	10	5.8207 (7.8e–01)	8.8257 (2.8e–01) +	8.5187 (3.1e–01) +	7.4196 (5.1e–01) +
	20	0.2301 (1.1e–01)	0.5749 (1.2e–01) +	0.4608 (9.2e–02) +	0.1929 (1.1e–01) =
DTLZ7	5	12.584 (1.6e + 00)	4.6368 (8.1e–01) –	1.2730 (2.6e–01) –	1.2727 (4.6e–01) –
	10	22.764 (2.7e + 00)	5.9195 (1.5e + 00) –	4.7156 (6.4e–01) –	2.3166 (1.3e–01) –
	20	4.7349 (1.1e + 00)	3.6157 (6.3e–01) –	2.9332 (8.1e–02) –	2.6058 (4.9e–01) –
+/-/=	–	10/7/4	15/3/3	9/8/4	

6. Inverted generational distance (IGD) [4] is adopted as the performance indicator, which calculates the Euclidean distance between the non-dominated solutions obtained by an algorithm and a set of reference points sampled from the theoretical Pareto front. The result of 20 independent runs are analyzed by the Wilcoxon rank sum at a significance level of 0.05, where symbol ‘+’ means that FDD-MOEA outperforms the compared method, ‘=’ indicates the two algorithms perform equally well, and ‘–’ means that the compared method outperforms FDD-MOEA.

Results and discussion

We implement the proposed FDD-MOEA with Python 3.7 [45], and the RBFN is constructed from scratch using the stochastic gradient descent method. The code of NSGA-II is taken from Pymoo [3], and the K-RVEA are implemented by PlatEMO v2.8.0 [43], and the performance indicators are also calculated on PlatEMO. All experiments are conducted on a computer with Intel(R) Core(TM) i7-8700 CPU @3.20GHz 3.19 GHz, Windows 10, version 1909.

Performance comparison on three-objective test instances

The first set of experiments compares FDD-MOEA with three centralized data-driven MOEAs, namely HeE-MOEA [18], GP-MOEA [18, 19] and K-RVEA [8] on three-objective DTLZ functions with a dimension of the decision space $d = 10, 20, 40, 80$. Specially, HeE-MOEA utilizes a heterogeneous ensemble to assist the optimization, while GP-MOEA is a variant of HeE-MOEA that adopts a GP as the surrogate for approximating each objective function. Note that HeE-MOEA, GP-MOEA and FDD-MOEA use NSGA-II as the baseline optimizer, while K-RVEA is based on RVEA. The parameter settings of the four methods are the same as defined in “Parameter settings”. The average IGD results on all these test instances are summarized in Table 2.

From the results in Table 2, we can observe that FDD-MOEA obtains the best IGD value on 16 out of 28 test instances, compared with HeE-MOEA, GP-MOEA and K-RVEA. These results confirm that the proposed FDD-MOEA works effectively for solving MOPs in a federated system. We also note that K-RVEA performs better than HeE-MOEA and GP-MOEA. It should be stressed that the purpose of proposing FDD-MOEA is not for developing a new MOEA that performs better than the state-of-the-art data-driven opti-

Table 4 Mean (std) of the IGD values obtained using different λ for different numbers of objectives when $d = 30$

Problems	M	$\lambda = 0.50$	$\lambda = 0.60$	$\lambda = 0.70$	$\lambda = 0.80$
DTLZ1	3	475.72 (6.4e + 01)	496.86 (7.1e + 01)	471.51 (7.5e + 01)	457.47 (6.0e + 01)
	5	398.62 (6.3e + 01)	400.85 (6.2e + 01)	389.34 (5.7e + 01)	380.88 (6.1e + 01)
	10	306.73 (2.6e + 01)	314.13 (3.6e + 01)	299.45 (4.9e + 01)	303.69 (4.6e + 01)
	20	129.90 (1.8e + 01)	124.11 (2.0e + 01)	116.39 (2.3e + 01)	117.44 (2.5e + 01)
DTLZ2	3	0.6341 (3.2e−02)	0.6197 (5.6e−02)	0.6616 (6.6e−02)	0.6439 (5.2e−02)
	5	0.9431 (5.2e−02)	0.9586 (4.3e−02)	0.9878 (5.5e−02)	0.9665 (6.3e−02)
	10	1.1574 (3.2e−02)	1.1349 (4.3e−02)	1.1651 (4.3e−02)	1.1255 (5.2e−02)
	20	1.2389 (1.9e−02)	1.2367 (1.9e−02)	1.2420 (1.8e−02)	1.2404 (1.6e−02)
DTLZ3	3	1135.4 (2.1e + 02)	1177.3 (1.9e + 02)	1163.9 (2.3e + 02)	1083.4 (1.4e + 02)
	5	1173.5 (1.7e + 02)	1118.1 (1.1e + 02)	1055.4 (1.9e + 02)	1066.2 (1.5e + 02)
	10	775.74 (1.4e + 02)	790.38 (1.4e + 02)	811.34 (1.4e + 02)	771.46 (1.1e + 02)
	20	348.21 (7.2e + 01)	365.86 (7.3e + 01)	323.94 (6.0e + 01)	336.40 (6.9e + 01)
DTLZ4	3	1.2373 (9.9e−02)	1.2001 (8.7e−02)	1.2321 (7.3e−02)	1.1815 (6.7e−02)
	5	1.4359 (4.7e−02)	1.4242 (6.5e−02)	1.4235 (6.7e−02)	1.4427 (6.3e−02)
	10	1.4256 (4.7e−02)	1.4541 (6.4e−02)	1.4396 (6.4e−02)	1.4522 (4.6e−02)
	20	1.2358 (3.4e−02)	1.2330 (3.0e−02)	1.2319 (3.1e−02)	1.2292 (2.9e−02)
DTLZ5	3	0.5427 (5.6e−02)	0.5940 (6.7e−02)	0.5785 (6.8e−02)	0.5651 (7.2e−02)
	5	0.7087 (7.8e−02)	0.7082 (7.5e−02)	0.6974 (6.0e−02)	0.6608 (6.4e−02)
	10	0.5259 (4.8e−02)	0.5091 (4.7e−02)	0.5369 (4.3e−02)	0.5052 (5.7e−02)
	20	0.2264 (2.0e−02)	0.2269 (1.9e−02)	0.2318 (2.1e−02)	0.2173 (1.6e−02)
DTLZ6	3	15.439 (1.2e + 00)	15.305 (1.4e + 00)	15.439 (1.4e + 00)	14.819 (1.4e + 00)
	5	14.102 (1.5e + 00)	13.438 (1.3e + 00)	14.057 (1.5e + 00)	12.985 (1.6e + 00)
	10	11.013 (1.3e + 00)	11.130 (1.1e + 00)	10.910 (1.2e + 00)	11.079 (1.3e + 00)
	20	4.9225 (7.4e−01)	4.9826 (1.2e + 00)	5.2115 (7.9e−01)	5.1356 (8.5e−01)
DTLZ7	3	8.4360 (9.8e−01)	8.4062 (7.7e−01)	7.8610 (1.1e + 00)	8.0583 (8.1e−01)
	5	14.426 (1.2e + 00)	14.386 (1.5e + 00)	14.275 (1.2e + 00)	14.396 (2.2e + 00)
	10	28.383 (2.1e + 00)	28.492 (2.0e + 00)	27.903 (2.0e + 00)	27.884 (3.0e + 00)
	20	49.375 (4.8e + 00)	50.736 (4.7e + 00)	51.217 (6.5e + 00)	46.742 (5.3e + 00)

mization algorithms. Rather, it aims to propose an algorithm that can still work effectively when the data are distributed on different clients and are not allowed to centrally stored.

The convergence profiles of the compared algorithms on DTLZ1–DTLZ7 when $d = 10, 80$ over the iterations are plotted in Figs. 3, 4, 5, 6, 7, 8 and 9. From these results, we find that the convergence properties of different algorithms strongly differ on different test instances. One general conclusion is that the performance advantage of FDD-MOEA over the compared algorithms will become stronger as the search dimension increases. This might be attributed to the use of the RBFNs in FDD-MOEA, which has been demonstrated powerful for high-dimensional systems in previous research.

Performance comparison on MaOPs

To further examine the effectiveness of the proposed FDD-MOEA on distributed optimization of many-objective problems, we conduct experiments by setting the number of objectives $M = 5, 10, 20$ for a fixed dimension $d = 20$. In this case, all algorithms under comparison use RVEA as the baseline optimizer. The parameter settings for all methods are provided in “Parameter settings”. Table 3 presents the statistical results in terms of the IGD values.

Overall, FDD-MOEA performs slightly better than HeE-MOEA and K-RVEA, and much better than GP-MOEA on the MaOP test instances. Compared with HeE-MOEA, FDD-MOEA performs better on 10 out of 21 test instances, equally well and worse on four and seven instances, respectively. Similar to the results presented above, the performance of FDD-MOEA is consistently competitive on all DTLZ1–DTLZ5 instances. FDD-MOEA performs particularly well

Fig. 10 Converged IGD values when $M = 20$, $d = 30$ with different λ

on DTLZ6 when RVEA is adopted as the optimizer, in contrast to its performance when NSGA-II is used. On DTLZ7, FDD-MOEA is outperformed by HeE-MOEA. In addition, FDD-MOEA outperforms GP-MOEA on 15 out of 21 test instances, achieves similar results on three instances, and is underperformed on four instances.

In general, K-RVEA achieves the second best results on many-objective problems, and K-RVEA outperforms FDD-MOEA on the DTLZ2 and DTLZ7 test instances. Interestingly, when using RVEA as the optimizer, the perfor-

mance of FDD-MOEA on DTLZ6 problems has a significant improvement and it outperforms K-RVEA in all cases, which means that using RVEA as the optimizer makes FDD-MOEA more competitive when compared with K-RVEA.

Influence of λ

The participation ratio, λ , is an important parameter in federated learning systems. In federated data-driven optimization, a low participation rate will lead to a decrease in the amount

Table 5 Mean (std) of the IGD values obtained using different p_f for different numbers of objectives when $d = 30$

Problems	M	$p_f = 1\%$	$p_f = 3\%$	$p_f = 5\%$	$p_f = 10\%$
DTLZ1	3	425.28 (5.9e + 01)	428.37 (8.1e + 01)	443.01 (9.1e + 01)	456.45 (6.6e + 01)
	5	401.79 (5.3e + 01)	370.45 (5.7e + 01)	381.59 (5.4e + 01)	406.36 (6.4e + 01)
	10	289.57 (5.1e + 01)	293.33 (5.7e + 01)	294.34 (4.9e + 01)	295.57 (5.2e + 01)
	20	123.13 (2.4e + 01)	124.45 (2.0e + 01)	122.01 (1.7e + 01)	119.86 (2.2e + 01)
DTLZ2	3	0.6255 (4.8e−02)	0.6432 (5.9e−02)	0.6295 (5.4e−02)	0.6526 (4.6e−02)
	5	0.9737 (4.5e−02)	0.9676 (5.2e−02)	0.9596 (5.8e−02)	0.9984 (5.4e−02)
	10	1.1639 (5.1e−02)	1.1481 (3.4e−02)	1.1628 (5.2e−02)	1.1549 (4.9e−02)
	20	1.2301 (1.7e−02)	1.2376 (2.1e−02)	1.2382 (1.6e−02)	1.2424 (1.5e−02)
DTLZ3	3	1055.2 (1.1e + 02)	1057.7 (1.6e + 02)	1014.7 (1.4e + 02)	1108.5 (1.6e + 02)
	5	1036.4 (1.7e + 02)	1001.7 (2.3e + 02)	1066.3 (1.6e + 02)	1025.0 (2.1e + 02)
	10	710.35 (1.5e + 02)	689.86 (1.2e + 02)	811.14 (1.4e + 02)	793.48 (1.6e + 02)
	20	303.42 (5.8e + 01)	319.20 (6.2e + 01)	325.92 (5.6e + 01)	328.17 (6.3e + 01)
DTLZ4	3	1.2583 (8.7e−02)	1.2479 (7.9e−02)	1.2221 (7.1e−02)	1.2166 (9.9e−02)
	5	1.4474 (7.0e−02)	1.4226 (6.2e−02)	1.4283 (4.9e−02)	1.4276 (7.5e−02)
	10	1.4347 (4.4e−02)	1.4493 (2.7e−02)	1.4681 (4.8e−02)	1.4540 (5.2e−02)
	20	1.2269 (3.3e−02)	1.2165 (2.7e−02)	1.2339 (3.1e−02)	1.2342 (3.0e−02)
DTLZ5	3	0.5753 (6.2e−02)	0.5554 (5.8e−02)	0.5708 (4.5e−02)	0.5725 (6.5e−02)
	5	0.6717 (8.8e−02)	0.7073 (3.9e−02)	0.7044 (6.4e−02)	0.6980 (6.3e−02)
	10	0.5118 (7.5e−02)	0.5363 (5.5e−02)	0.5153 (6.2e−02)	0.5217 (6.1e−02)
	20	0.2251 (2.1e−02)	0.2200 (1.8e−02)	0.2329 (2.1e−02)	0.2300 (1.9e−02)
DTLZ6	3	14.487 (1.5e + 00)	14.060 (2.0e + 00)	14.562 (1.5e + 00)	13.952 (1.3e + 00)
	5	12.484 (2.0e + 00)	13.263 (2.0e + 00)	13.457 (1.4e + 00)	12.686 (1.5e + 00)
	10	9.8617 (1.6e + 00)	10.235 (1.3e + 00)	10.400 (9.1e−01)	10.425 (1.5e + 00)
	20	4.8421 (9.5e−01)	4.7037 (7.3e−01)	5.0091 (8.1e−01)	4.9551 (8.3e−01)
DTLZ7	3	8.0467 (1.0e + 00)	8.3347 (8.5e−01)	8.2695 (8.5e−01)	8.2858 (8.6e−01)
	5	13.971 (2.0e + 00)	14.128 (1.4e + 00)	14.413 (1.4e + 00)	14.118 (1.9e + 00)
	10	28.991 (1.7e + 00)	28.430 (2.2e + 00)	27.823 (2.7e + 00)	28.451 (1.7e + 00)
	20	47.002 (5.1e + 00)	48.388 (6.4e + 00)	50.248 (4.7e + 00)	50.102 (4.3e + 00)

of training data on each client. To examine the influence of the participation ratio λ on the performance of FDD-MOEA, empirical experiments are conducted by varying the values of λ , while the number of decision variables is fixed to $d = 30$ and RVEA is adopted as the optimizer.

As listed in Table 4, the results indicate that FDD-MOEA is relatively insensitive to the participation ratio and in general, a larger participation ratio contributes to better performance, which is expected. Furthermore, we also investigate the influence of λ on the performance of FDD-MOEA DTLZ1-7 problems in terms of IGD for $M = 20$, $d = 30$. The performance changes of FDD-MOEA over different λ values are presented in Fig. 10. The experimental results show that higher participation ratios will lead to an improvement in the performance of FDD-MOEA, mainly because the total amount of selected solutions is limited, and the amount of local training data on each client may significantly decrease with a lower participation ratio λ .

Influence of p_f

The aim of federated data-driven EAs is to solve of distributed optimization problems without collecting private data stored on local clients [51] and hence, communications between clients and the server when exchanging surrogate information are unavoidable. However, there are many potential reasons that may cause communication failures or packet losses, such as network congestion or when communication network is not available. As mentioned in “Overall framework”, we assume that each client has a failure probability of p_f when communicating with the server, which means that the m solutions obtained in this iteration cannot be passed to the client. In this part, we study the impact of p_f on the performance of FDD-MOEA, and the experimental results with $p_f = 1\%$, 3% , 5% and 10% are summarized in Table 5, and the converged IGD values of the compared algorithms are plotted in Fig. 11 when $M = 20$.

Fig. 11 Converged IGD values when $M = 20$, $d = 30$ with different p_f

Specifically, the dimension of the decision space d and the participation ratio are fixed to 30 and 0.90, respectively, and RVEA is set as the optimizer. As we can see, when p_f is set to 1% and 3%, the performance of FDD-MOEA does not change much and becomes slightly when p_f is increased to 5% or 10%. This is also expected since the larger p_f is, the less data will be available for model management on each client.

Conclusions and future work

It may become unrealistic to assume that all data can be made available for centralized training of surrogates in data-driven evolutionary optimization. To remove this strong assumption,

we consider the challenging situation in which data are distributed on different local clients and propose a federated data-driven multi-/many-objective evolutionary optimization algorithm in this work. By leveraging federated learning, a global surrogate is trained collaboratively for fitness approximations, a sorted averaging algorithm is used to aggregate the local models, and the uploaded local surrogates are utilized to provide the uncertainty of solutions. In addition, a modified acquisition function combining local and global information is proposed for surrogate management. Empirical results indicate that the proposed algorithm achieves comparable results on 49 benchmark problems with the state-of-the-art of centralized data-driven evolutionary algorithms.

However, the proposed algorithm does not perform well on the problems with disconnected Pareto fronts, in which

case the local model may strongly differ from each other, causing considerable model divergence. Hence, in the future, we will investigate more robust modelling techniques and new acquisition functions for handling strongly non-iid data for data-driven optimization.

Acknowledgements This work was supported by National Natural Science Foundation of China (Basic Science Center Program: 61988101), International (Regional) Cooperation and Exchange Project (61720106008), National Natural Science Fund for Distinguished Young Scholars (61725301, 61925305).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Akhtar T, Shoemaker CA (2019) Efficient multi-objective optimization through population-based parallel surrogate search. arXiv preprint [arXiv:1903.02167](https://arxiv.org/abs/1903.02167)
2. Allmendinger R, Emmerich MTM, Hakanen J, Jin Y, Rigoni E (2017) Data-driven surrogate-assisted multi-objective evolutionary optimization of a trauma system. *J Multi-Criteria Decis Anal* 24(1/2):5–24
3. Blank J, Deb K (2020) pymoo: Multi-objective optimization in python. *IEEE Access* 8:89497–89509
4. Bosman PA, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7(2):174–188
5. Briffoteaux G, Ragonnet R, Mezmaiz M, Melab N, Tuytens D (2020) Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times. *Future Gener Comput Syst* 113:454–467
6. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
7. Chugh T, Chakraborti N, Sindhya K, Jin Y (2017) Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodelling. *Mater Manuf Process* 32(1):1172–1178
8. Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2016) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans Evol Comput* 22(1):129–142
9. Chugh T, Sindhya K, Hakanen J, Miettinen K (2017) A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Comput* 23:3137–3166
10. Chugh T, Sindhya K, Miettinen K, Jin Y, Kratky T, Makkonen P (2017) Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp 1541–1548. IEEE
11. Coello Coello CA, González Brambila S, Figueroa Gamboa J, Guadalupe Castillo Tapia M, Hernández Gómez R (2020) Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex Intell Syst* 6(2):221–236
12. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197
13. Du KL, Swamy MNS (2014) Radial basis function networks. In: *Neural networks and statistical learning*, 1 edn. Springer, London, pp 299–335
14. Emmerich M, Beume N, Naujoks B (2005) An EMO algorithm using the hypervolume measure as selection criterion. In: *Evolutionary multi-criterion optimization*, pp 62–76
15. Emmerich MT, Giannakoglou K, Naujoks B (2006) Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodelling. *IEEE Trans Evol Comput* 10(4):421–439
16. Emmerich MT, Giannakoglou KC, Naujoks B (2006) Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodelling. *IEEE Trans Evol Comput* 10(4):421–439
17. Fu G, Sun C, Tan Y, Zhang G, Jin Y (2021) A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems. In: *Parallel problem solving from nature*, pp 125–139
18. Guo D, Jin Y, Ding J, Chai T (2018) Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Trans Cybern* 49(3):1012–1025
19. Guo D, Wang X, Gao K, Jin Y, Ding J, Chai T (2021) Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. *IEEE Trans Syst Man Cybern Syst*
20. Huang P, Wang H, Jin Y (2020) Transfer stacking from low-to-high-fidelity: a surrogate-assisted bi-fidelity evolutionary algorithm. *Appl Soft Comput* 92:106276
21. Huang P, Wang H, Jin Y (2021) Offline data-driven evolutionary optimization based on tri-training. *Swarm Evol Comput* 60:100800
22. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evol Comput* 10(5):477–506
23. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput* 9(1):3–12
24. Jin Y, Wang H, Chugh T, Guo D, Miettinen K (2018) Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans Evol Comput* 23(3):442–458
25. Jin Y, Wang H, Sun C (2021) *Data-driven evolutionary optimization*. Springer, New York
26. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13(4):455–492
27. Knowles J (2006) ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans Evol Comput* 10(1):50–66
28. Li B, Li J, Tang K, Yao X (2020) Many-objective evolutionary algorithms: a survey. *ACM Comput Surv* 6(2):221–236
29. Li JY, Zhan ZH, Wang H, Zhang J (2020) Data-driven evolutionary algorithm with perturbation-based ensemble surrogates. *IEEE Trans Cybern*

30. Li M, Andersen DG, Park JW, Smola AJ, Ahmed A, Josifovski V, Long J, Shekita EJ, Su BY (2014) Scaling distributed machine learning with the parameter server. In: 11th USENIX symposium on operating systems design and implementation (OSDI 14), pp 583–598
31. Lim D, Jin Y, Ong YS, Sendhoff B (2010) Generalizing surrogate-assisted evolutionary computation. *IEEE Trans Evol Comput* 14(3):329–355
32. Loshchilov I, Schoenauer M, Sebag M (2010) An EMO algorithm using the hypervolume measure as selection criterion. In: Congress on evolutionary computation, pp 471–478
33. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of artificial intelligence and statistics. PMLR, pp 1273–1282
34. Montoya MC, Nieto F, Hernández S, Kusano I, Álvarez A, Jurado J (2018) Cfd-based aeroelastic characterization of streamlined bridge deck cross-sections subject to shape modifications using surrogate models. *J Wind Eng Ind Aerodyn* 177:405–428
35. Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y (2019) A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput* 23(1):74–88
36. Park S, Na J, Kim M, Lee JM (2018) Multi-objective Bayesian optimization of chemical reactor design using computational fluid dynamics. *Comput Chem Eng* 119:25–37
37. Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat* 400–407
38. Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 104(1):148–175
39. Shankar Bhattacharjee K, Kumar Singh H, Ray T (2016) Multi-objective optimization with multiple spatially distributed surrogates. *J Mech Des* 138(9):091401
40. Song Z, Wang H, He C, Jin Y (2021) A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput*
41. Sun X, Gong D, Jin Y, Chen S (2013) A new surrogate-assisted interactive genetic algorithm with weighted semi-supervised learning. *IEEE Trans Cybern* 43(2):685–698
42. Sun Y, Wang H, Xue B, Jin Y, Yen GG, Zhang M (2020) Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Trans Evol Comput* 24(2):350–364
43. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: a matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87
44. Torczon V, Trosset M (1998) Using approximations to accelerate engineering design optimization. In: 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, p 4800
45. Van Rossum G, Drake FL Jr (1995) Python tutorial. Centrum voor Wiskunde en Informatica, Amsterdam
46. Wang H, Jin Y (2020) A random forest assisted evolutionary algorithm for data-driven constrained multi-objective combinatorial optimization of trauma systems. *IEEE Trans Cybern* 50(2):536–549
47. Wang H, Jin Y, Jansen JO (2016) Data-driven surrogate-assisted multi-objective evolutionary optimization of a trauma system. *IEEE Trans Evol Comput* 20(6):939–952
48. Wang X, Jin Y, Schmitt S, Olhofer M (2020) Transfer learning for gaussian process assisted evolutionary bi-objective optimization for objectives with different evaluation times. In: Genetic and evolutionary computation conference, pp 587–594
49. Wang X, Jin Y, Schmitt S, Olhofer M, Allmendinger R (2021) Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times. *Knowl Based Syst*
50. Xu J, Du W, Jin Y, He W, Cheng R (2020) Ternary compression for communication-efficient federated learning. *IEEE Trans Neural Netw Learn Syst*
51. Xu J, Jin Y, Du W, Gu S (2021) A federated data-driven evolutionary algorithm. arXiv preprint [arXiv:2102.08288](https://arxiv.org/abs/2102.08288)
52. Yang C, Ding J, Jin Y, Chai T (2020) Offline data-driven evolutionary optimization based on tri-training. *IEEE Trans Evol Comput* 24(3):409–423
53. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
54. Zhang Q, Liu W, Tsang E, Virginas B (2009) Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Trans Evol Comput* 14(3):456–474
55. Zhang X, Tian Y, Jin Y (2015) A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(6):761–776
56. Zhao Y, Sun C, Zeng J, Tan Y, Zhang G (2021) A surrogate-ensemble assisted expensive many-objective optimization. *Knowl Based Syst* 211:106520
57. Zhou Q, Wu J, Xue T, Jin P (2019) A two-stage adaptive multi-fidelity surrogate model-assisted multi-objective genetic algorithm for computationally expensive problems. *Eng Comput* 1–17
58. Zhou Y, Jin Y, Ding J (2020) Surrogate-assisted evolutionary search of spiking neural architectures in liquid state machines. *Neurocomputing* 406:12–23
59. Zhu H, Xu J, Liu S, Jin Y (2021) Federated learning on non-iid data: a survey. *Neurocomputing*
60. Zhu H, Zhang H, Jin Y (2021) From federated learning to federated neural architecture search: a survey. *Complex Intell Syst* 7(2):639–657

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.