Multi-objective methods in Federated Learning: A survey and taxonomy

Maria Hartmann[1], **Grégoire Danoy**[1,2] and **Pascal Bouvry**[2]
[1]SnT, University of Luxembourg
[2]FSTM/DCS, University of Luxembourg
{maria.hartmann, gregoire.danoy, pascal.bouvry}@uni.lu,

Abstract

The Federated Learning paradigm facilitates effective distributed machine learning in settings where training data is decentralized across multiple clients. As the popularity of the strategy grows, increasingly complex real-world problems emerge, many of which require balancing conflicting demands such as fairness, utility, and resource consumption. Recent works have begun to recognise the use of a multi-objective perspective in answer to this challenge. However, this novel approach of combining federated methods with multi-objective optimisation has never been discussed in the broader context of both fields. In this work, we offer a first clear and systematic overview of the different ways the two fields can be integrated. We propose a first taxonomy on the use of multi-objective methods in connection with Federated Learning, providing a targeted survey of the state-of-the-art and proposing unambiguous labels to categorise contributions. Given the developing nature of this field, our taxonomy is designed to provide a solid basis for further research, capturing existing works while anticipating future additions. Finally, we outline open challenges and possible directions for further research.

1 Introduction

The Federated Learning (FL) paradigm allows the training of machine learning models in the difficult setting where training data is distributed and compartmentalised. Instead of centralising available data, FL performs local training in distribution, with the resulting local models aggregated periodically across participants. Though originally designed to mitigate privacy concerns, the method has also shown great success in other use cases, including communication-restricted settings such as drone networks [Brik *et al.*, 2020] or computationally costly settings such as the tuning of large language models [Che *et al.*, 2023].

However, as Federated Learning is being adopted for increasingly diverse applications and real-world use cases, new challenges are emerging, many linked to the need to balance different conflicting requirements: (i) Heterogeneity between participants caused by data imbalances or differing hardware capabilities can lead to divergent local models that cannot easily be aggregated without loss of model utility [Karimireddy *et al.*, 2019]. Designing mitigation strategies for this raises the problem of fairness – the choice between sacrificing the performance of some individual clients or that of the global model. (ii) The cost of FL in terms of communication and computation resources scales with the size of the model and the number of update messages; yet reducing either may come at the cost of decreasing model utility [Zhu *et al.*, 2021]. (iii) Strategies for mitigating privacy leakage, the problem of exposing confidential information to potential attackers through client updates, may degrade other aspects of the federated system in turn. For example, adding noise to client updates may obscure sensitive information effectively, but reduce model performance as well [Geng *et al.*, 2024].

All these scenarios can be modelled as multi-objective problems, with each problem-specific performance metric represented as a separate objective. Under this multi-objective perspective, problems are solved with explicit consideration for several characteristics, potentially conflicting, and solutions can represent different optimal trade-offs between all objectives. As such, the approach can assist users in making informed decisions about complex FL problems by presenting explicit choices where a single-objective approach would yield none. Indeed, these general advantages of multi-objective methods have been recognised across disciplines, and the field of multi-objective optimisation (MOO) has been thriving for decades. This success opens another interesting avenue of research in connection with federated learning: deploying FL methods to facilitate multi-objective learning in distribution, where problems would otherwise be difficult to solve for participants that cannot share local training data.
Recent works in the literature have begun to combine federated learning with MOO methods to address a wide range of challenges. However, the broader context of the intersection between MOO and FL has not yet been discussed. This work aims to provide a first such systematic overview, identifying general challenges and parallels, and formulating a novel taxonomy to classify existing work while highlighting open directions of research. Many FL strategies already use (linear) combinations of multiple functions as objectives, but do not consider the problem from a multi-objective angle. The first works to explicitly introduce multi-objective methods to Fed-

erated Learning aimed to improve federated aggregation and introduce fairness between clients [Hu *et al.*, 2022], followed by approaches introducing other, system-wide aggregation parameters [Mehrabi *et al.*, 2022]. Another early adoption of MOO was in hyperparameter optimisation for FL [Zhu and Jin, 2020]. More recently, research has also begun into supporting the inverse scenario: developing strategies to federate the solving of multi-objective problems by distributed clients, e.g. [Yang *et al.*, 2023][Hartmann *et al.*, 2023]. The contributions of this work can be summarised thus:

- We propose a novel taxonomy of algorithms combining MOO methods and FL, offering a unified naming system for works at the intersection of two previously largely separate fields with separate naming conventions.
- We present a thorough review of the state of the art, categorising and contrasting existing works.
- We highlight open questions and offer perspectives on open avenues for future research.

The rest of this work is organised as follows: Section 2 reviews important notions from the fields of FL and MOO. Section 3 introduces our taxonomy, discussing in detail each category and relevant works from the literature. Finally, we offer a conclusion and perspectives on future research in Section 4.

2 Background

In this section, we briefly introduce fundamental concepts from the fields of federated learning and multi-objective optimisation in preparation for the main body of the survey.

2.1 Federated Learning

The Federated Learning [McMahan *et al.*, 2017] paradigm was originally designed to solve arbitrary (neural network-based) machine learning problems in a difficult distributed setting. This setting is characterised by (i) the available data originating in distribution, with no control over the composition of the resulting datasets, and (ii) a restriction on transmitting private client information, including raw training data, between participants. FL overcomes the constraint introduced by (ii) by training separate local models in distribution on each dataset holder, or client, and aggregating only the resulting models across clients – see Figure 1.

A more detailed general framework of the Federated Learning strategy is presented in Algorithm 1, with colours highlighting the correspondence of code segments to different levels of the federated system (to be presented in detail in Section 2.3). First, the federated system is initialised with the identity of the server, a list of participating clients, and the definition of the underlying learning problem to be solved. Additional hyperparameters are passed depending on the specific algorithm, defining e.g. the architecture of the neural network to be trained, a client sampling rate, gradient thresholds, or any other parameter required by the algorithm. Then, the local learning process begins. During each federated training round, a set of clients is selected for participation. These clients each carry out local training and return the resulting models to the server. These local models are aggregated periodically by the server into a single global model incorporating the locally learned information. The global model is then passed back to the local clients to continue the next local training round. Expressed formally, the FL process aims to find a global model θ that generalises to all available data, i.e.

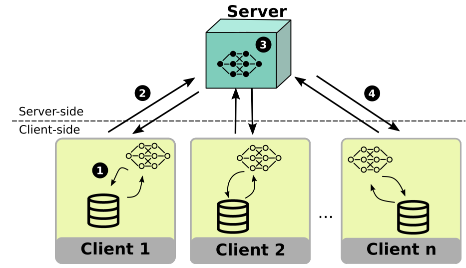

Figure 1: The FL paradigm. During each round, clients perform local model training (1), then transmit their local models to the server (2) for aggregation into a single global model (3). The global model is returned to the clients (4) to begin the next training round.

Algorithm 1 The general Federated Learning framework.

Input: Server, list of clients, local learning problem.
Parameter: Optional list of hyperparameters.
Output: Global model θ.

```
Initialise system parameters.
while stopping condition not satisfied do
    for all participating clients do
        while local stopping condition not satisfied do
            Perform training on local data.
        end while
        Transmit local model to server.
    end for
    Aggregate local models to obtain new global model.
    Return global model to clients.
end while
return global model.
```

$$minimise_{\theta} f(\theta, \mathcal{D}), \qquad (1)$$

where $\mathcal{D} := \bigcup_i^n \mathcal{D}_i$, with $\mathcal{D}_i$ the dataset of the i-th client. Imbalances between client datasets, as can be caused by characteristic (i), represent a significant challenge to the model aggregation step of FL algorithms. Indeed, any type of heterogeneity between clients, e.g. in terms of hardware capability or feature distribution, may have an adverse impact on the convergence of the federated model. Mitigating the impact of various types of client heterogeneity remains an active field of study. Other major research topics in FL include the reduction of resource consumption – mainly computing and communication cost – and how to protect against malicious actors. For a comprehensive overview of the state of the art in the field, we refer to [Kairouz *et al.*, 2021].

2.2 Multi-objective optimisation

Multi-objective optimisation is concerned with solving problems in the presence of more than one objective. As an example, consider the problem of selecting hyperparameters for a neural network to simultaneously maximise model utility and minimise the cost of training. Instead of a single objective $f(x)$, such a multi-objective problem is expressed as a vector of n objectives $\vec{f}(x) := (f_1(x), \ldots, f_n(x))^T$. Note that individual objectives can conflict, i.e. in general no single solution can optimise all objectives simultaneously. Instead, MOO methods typically focus on identifying solutions that represent an optimal trade-off between objectives, where objective values are balanced so that no single objective can be improved without sacrificing the performance of another. Such trade-off solutions are known as *Pareto-optimal*. Pareto optimality can be difficult to determine in practice, where the optimal values achievable for each objective are unknown, so the weaker notion of *Pareto-dominance* is commonly used instead. A solution x is said to Pareto-dominate another solution y iff it outperforms y in at least one objective while matching or improving the value of all others. Formally,

$$x \succ_P y \iff \exists j f_j(x) > f_j(y) \land \forall i f_i(x) \geq f_i(y) \quad (2)$$

for a maximisation problem. Pareto-optimal solutions are not dominated by any others. The set of such solutions is known as the *Pareto front* (see Fig. 2). Most MOO algorithms are either designed to find such a Pareto front, or a single solution based on predefined requirements such as user preferences. A wide range of algorithmic approaches exists for both variants, tailored to different problem characteristics. In this work, we will discuss relevant MOO strategies as they appear; for a comprehensive overview we refer to [Talbi, 2009].

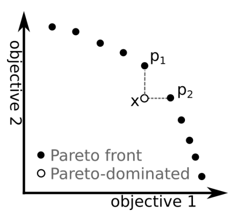

Figure 2: Pareto front and Pareto dominance. Shaded markers represent solutions on the Pareto front of a bi-objective maximisation problem; x is Pareto-dominated by p_1 and p_2.

2.3 Integrating multi-objective methods and Federated Learning

We note that multi-objective methods can be integrated with FL at different levels of the federated system, each with distinct implications for the algorithmic components involved. Based on this insight, we propose a three-level view of the federated system – see Fig. 3 and corresponding colours in Alg. 1. Adding multi-objective methods on top of a federated algorithm necessitates no modification of the underlying federation or local learning process; an example for such a method is offline hyperparameter tuning with respect to multiple requirements. On the other hand, introducing multi-objectivity at the federated level, e.g. for model aggregation on the server, forces adaptation at the top level as well: any hyperparameter algorithm running on the federated system must accommodate new parameters introduced by multi-objective methods. Finally, adding a multi-objective perspective to the lowest level in Fig. 3 – the client level – requires modifications across the entire system: (i) The local learning algorithm on each client must handle multi-objective problems; (ii) the federated algorithm must aggregate client submissions, which may include multi-objective gradients or be influenced by heterogeneous client objectives, and (iii) any hyperparameter must be adjusted once again.

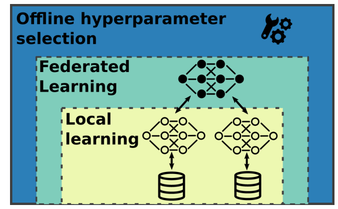

Figure 3: Relation of major categories of the taxonomy. Multi-objective methods can be integrated at different levels of the federated system: in the local learning process of clients, at system-level in the federated algorithm, or outside of the federated system.

3 Taxonomy: multi-objective methods in FL

In this section, we introduce our proposed taxonomy, discussing each category and the related existing work. The full taxonomy is shown in Figure 4. A first fundamental distinction is the purpose that multi-objective and federated methods each serve in an algorithm. We can identify two main broad categories: one where MOO methods are applied to enhance the functionality of a federated system, and the inverse, where FL is used in support of solving a general multi-objective problem in distribution. We refer to these categories as *Multi-Objective Federated Learning (MOFL)* and *Federated Multi-Objective Learning (FMOL)*, respectively, to indicate the different chaining of strategies. MOFL covers the majority of existing research, and is notably precisely equivalent to the top two layers as shown in Fig. 3 and introduced in Section 2.3. Works in this section can accordingly be divided further into top-level and federation-level methods, and will be discussed as such in the following sections. FMOL, in contrast, corresponds to the lowest layer in Fig. 3, and extends the "standard" FL scenario, where Federated Learning is used to solve an arbitrary learning problem in distribution, to include multi-objective learning problems.

3.1 Multi-objective federated learning at top level

Methods at the top level of a federated system, as defined in Fig. 3, are decoupled from the federated learning and aggregation process and can treat the federated algorithm as a black-box system. As such, this class of algorithms is arguably the least specific to the FL context, since modifications at this level require no particular adaptation to the federated setting. Current work can largely be divided into two major applications: multi-objective neural architecture search (MO-NAS), focused on optimising the architecture of a neural network with respect to multiple objectives, and more general multi-objective hyperparameter tuning, where other hyperparameters of the federated system are tuned. Both types typically employ population-based multi-objective strategies, known to offer effective search space exploration.

Offline hyperparameter tuning

Multi-objective hyperparameter tuning can find algorithm parameters for additional requirements beyond the utility of the

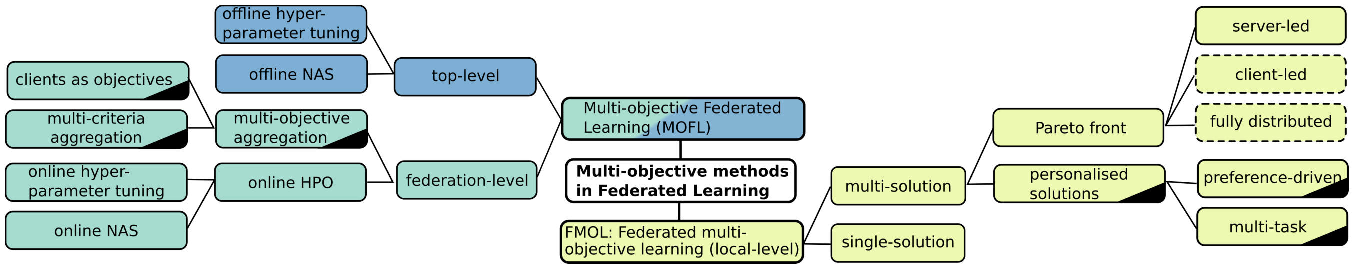

Figure 4: Proposed taxonomy. Colours denote the level of the federated system where MO methods are integrated (see Fig. 3 and Sec. 2.3). Some categories arise from the unique properties of the FL setting; these are marked by a shaded corner. Categories in dashed boxes are currently unexplored in the literature.

global model. Depending on the use case, FL systems may face challenges such as privacy restrictions, resource limitations, or malicious attacks. This approach allows users to explicitly model such requirements and make informed choices about the trade-offs inherent to different solutions.

[Kang *et al.*, 2024b] assert that optimising hyperparameters solely for model performance may expose the federation to a risk of data leakage. The proposed mitigation approach optimises the three objectives of model performance, training cost and privacy leakage simultaneously. This algorithm, derived from NSGA-II [Deb *et al.*, 2002], a well-known population-based baseline algorithm, is designed to find a Pareto front of possible configurations representing different trade-offs between these objectives. [Morell *et al.*, 2024] also introduce a second objective in addition to the model accuracy, based on the mean amount of data transmitted and received by clients. This approach is designed to optimise a large number of hyperparameters and algorithmic choices, including the number of local training steps, the number of bits used to encode local updates, and whether clients submit gradient or weight updates. All variables are optimised using a hybrid of NSGA-II and an estimated distribution-based algorithm (EDA). [Geng *et al.*, 2024] formulate a similar strategy, also using NSGA-II, but considering the four objectives of minimising global model error rate, the variance of model accuracy, the communication cost, and a privacy budget.

Offline neural architecture search

Neural architecture search aims to optimise the structure of a neural network for given objectives. Federated NAS can be seen as an inherently multi-objective problem [Zhu *et al.*, 2021], as changes to the model structure impact not only the model utility, but also other aspects of the federated system, such as the communication and training cost. One of the first works on multi-objective federated neural architecture search [Zhu and Jin, 2020] proposes an offline federated NAS algorithm that constructs models with the two objectives of minimising the validation error obtained by the model, and the cost of communicating the model. Solutions are once again generated using NSGA-II. The same problem is tackled in [Chai *et al.*, 2022], but with the use of a multi-objective evolutionary algorithm (MOEA) instead of NSGA-II to improve the exploration of the multi-objective search space.

Federated split learning is a related problem, where partial blocks of the global model are assigned to clients, with blocks of different size assigned to clients depending on the available resources. [Yin *et al.*, 2023] propose to optimise this splitting decision, along with communication bandwidth and computing resource allocation, as a multi-objective problem, minimising training time and energy consumption of the system. The proposed algorithm yields a Pareto front of solutions using a hybrid of NSGA-III and a generative adversarial network trained to identify configurations generating Pareto-dominated solutions. Research on offline MO-NAS algorithms for FL is arguably more advanced than other areas of MOFL, as existing approaches can be applied to the federated setting without change. The main challenge remains the high computational cost of these methods.

3.2 Multi-objective federated learning at federation-level

MOO methods can also be integrated with FL at the server-level to solve challenges inherent to the FL paradigm – a brief overview of representative works from the literature is presented in Table 1. The majority of existing works focus on one of two design aspects of a federated system: the aggregation strategy used on the federated server, and the selection of relevant hyperparameters for the FL algorithm. We discuss both separately, beginning with multi-objective aggregation.

Multi-objective aggregation

The aggregation of local model updates by the server can be modelled as a MOO problem, permitting the use of more than one criterion for computing the global model. This multi-objective version of federated aggregation can be formulated in general terms as follows:

$$min_\theta(f_1(\theta), \dots, f_n(\theta)),^T \tag{3}$$

where θ is the global model and f_i is the loss function of the i-th objective. Solving this problem typically translates to finding optimal aggregation weights λ_i to compute the global model from the local models:

$$min_{\lambda_1 \dots, \lambda_n}(f_1(\theta), \dots, f_n(\theta)),^T \text{ , with } \theta = \sum_i^n \lambda_i \theta_i. \tag{4}$$

The literature on FL algorithms with multi-objective aggregation can be categorised based on the nature of the objectives [Kang *et al.*, 2024a]. One line of work derives objectives from the performance of individual clients; the other uses objectives that describe the federation as a whole. This distinction is significant, as the different mathematical properties of these variants permit the use of different multi-objective methods. The following sections discuss both types in detail.

Reference	Taxonomy label	System level	MOO method	Objectives
[Hu *et al.*, 2022]	Clients as objectives	federation-level	MGDA	Local model utilities
[Ju *et al.*, 2024]	Clients as objectives	federation-level	dynamic preferences	Fairness, convergence
[Mehrabi *et al.*, 2022]	Multi-criteria aggregation	federation-level	obj.-contribution scoring	Arbitrary system objs.
[Zhu and Jin, 2022]	online MO-NAS	federation-level	NSGA-II	Global model utility, evaluation speed
[Kang *et al.*, 2024b]	offline MO-HPO	top-level	NSGA-II	Model utility, training cost, privacy leakage

Table 1: Comparison of selected MO-FL algorithms. Each row lists the level of the federated system where multi-objective notions are introduced, as well as the method used to solve the multi-objective problem.

Clients as objectives. These algorithms consider the performance of individual clients and the global model as separate objectives. In client-heterogeneous settings, this approach can balance the interests of both the clients and the general system. This perspective enables explicit fairness guarantees for selfish participants, ensuring that the performance of individual clients is not sacrificed for that of the system in computing the global model. Crucially, performance criteria in this class of MOFL problems are tied directly to the client models and thus differentiable with respect to model parameters. As such, they can be solved efficiently using gradient-based multi-objective algorithms such as the classical multi-gradient descent algorithm (MDGA) [Désidéri, 2012], established in the field of MOO.

The FedMGDA+ algorithm [Hu *et al.*, 2022] leverages this insight, defining the performance of each participating client as a separate objective. Using MGDA yields aggregation weights for a common descent gradient for all clients, thus guaranteeing that no client suffers a reduced performance by participating in an aggregation step. An added constraint on the divergence of aggregation weights serves as protection against false updates by malicious participants. The FedMC+ algorithm [Shen *et al.*, 2025] is also designed to reconcile individual client updates and the global model in the presence of heterogeneous data. A secondary objective, minimising conflict between the global and local gradients, is introduced during the aggregation step and solved by transformation into a convex optimisation problem. [Cui *et al.*, 2021] formulate the aggregation step as a parameterised min-max optimisation problem. Fairness constraints serve to optimise model utility for the single worst-performing client while ensuring that (i) the utility of all clients improves, and (ii) no client improves much less than another. The solution obtained from this formulation is optimised further to guarantee Pareto-stationarity, a prerequisite for local optimality [Ye and Liu, 2022].

The three methods have different implications for the ultimate balance of client models. While both [Hu *et al.*, 2022] and [Cui *et al.*, 2021] (in its pure form) guarantee that all clients improve during an aggregation step, only the latter considers the magnitude of gradients in the calculation. Thus, [Cui *et al.*, 2021] may force a greater balance between clients, to the potential detriment of overall performance in highly heterogeneous settings. In contrast, [Shen *et al.*, 2025] may sacrifice an outlier for the benefit of the system. Though undesirable to selfish clients, the latter could offer a defence against intentionally divergent updates submitted by a malicious client.

Multi-criteria aggregation. These algorithms perform aggregation based on multiple metrics that describe different characteristics of the federated system, such as the accuracy of the global model and fairness between clients. Such criteria are not generally differentiable with respect to the model, and thus cannot be optimised using gradient-based methods [Kang *et al.*, 2024a]. Solution approaches rely instead on heuristic insights or the formulation of the aggregation step into a mathematically solvable optimisation problem.

[Mehrabi *et al.*, 2022] propose an algorithm that can incorporate multiple arbitrary system objectives, including fairness metrics, on the server. Aggregation is accomplished by assigning weighted ranking scores to each client for its contribution to optimising each objective, calculated using a validation dataset possessed by the server. These scores are used to compute aggregation weights. In contrast, [Ju *et al.*, 2024] formulate fairness-controlled FL as a dynamic multi-objective problem, where the optimisation problem consists of a linear combination of client losses, with weights adjusted dynamically to balance the progress of all component objectives. This approach yields different trade-off solutions between fairness and convergence depending on the value chosen for a fairness parameter. The idea of optimising a weighted linear combination of objectives in the federated aggregation step was proposed before in [Li *et al.*, 2020], generalising ideas from [Mohri *et al.*, 2019]; but neither work explicitly acknowledges a multi-objective view of the problem. Both aggregation strategies have different strengths and weaknesses. [Mehrabi *et al.*, 2022] offers transparent server-side evaluation of clients, including the potential to automatically recognise low-quality or malicious clients. However, the need for a validation dataset on the server may violate the privacy requirements of clients, and renders the method vulnerable to data poisoning attacks. Conversely, [Ju *et al.*, 2024] offers mathematical fairness guarantees, but little transparency in the aggregation process. In addition, this algorithm may be vulnerable to malicious client participation.

Online multi-objective hyperparameter optimisation

Algorithms that use MOO to optimise hyperparameters for the federated system may run off-line or on-line. In on-line algorithms, the optimisation process is integrated into the federated algorithm, i.e. parameters are changed during the runtime of the FL process. On-line candidate generation is typically integrated on the federated server at the aggregation step, with local training rounds used for evaluation. Existing works on online MO-HPO in FL can again be divided into hyperparameter tuning and neural architecture search.

Online hyperparameter tuning. The work by [Badar *et al.*, 2024] performs on-line hyperparameter optimisation for clients, generating and transmitting new parameters during each aggregation step. These parameters, a fairness constraint regularisation parameter and the learning rate designed to enforce fairness locally, are recomputed on the server-side by using multi-objective Bayesian optimisation. Finally, [Banerjee *et al.*, 2022] propose a multi-objective on-line device selection approach to speed up the learning process in the presence of stragglers. The selection algorithm is designed to maximise the available computing and communication resources on selected clients, using NSGA-II.

Online neural architecture search. NAS algorithms may be designed run on-line, modifying during the execution of the federated algorithm the structure of the neural network to be trained by each client. Such a strategy could significantly reduce the computational cost of the search, at the price of complicating the training and aggregation process by introducing dynamic parameters. The only such algorithm currently existing in the MOFL literature dynamically optimises the accuracy and evaluation speed of federated model training [Zhu and Jin, 2022]. The NSGA-II algorithm is used during each aggregation step to generate partial samples of the full model to assign to clients for training. On-line MO-NAS presents a difficult challenge and is currently underexplored in the literature, but could offer significant efficiency benefits.

3.3 Federated multi-objective learning

In federated multi-objective learning, the solving of a multi-objective learning problem (MOLP) is the ultimate goal, and FL acts as an auxiliary tool to facilitate learning in distribution. A major challenge compared to the class of MOFL algorithms is that in this setting, there is no control or information about the compatibility of the objectives involved in the problem, whereas in MOFL the objectives were designed to suit the federated setting. Note also that FL techniques have largely been developed for neural networks, so the focus in this setting is on MO-algorithms that train such models. Compared with the application of MO techniques to FL algorithms, the federated solving of MOLPs has received very little attention so far. Here we aim to offer a classification of the few existing works, and extrapolate the open challenges and problems that remain to be solved. See also Table 2 for a representative overview of existing works. On the most fundamental level, algorithms in this category can be separated by the number of solutions they are designed to find: one single solution to the MOLP, or multiple solutions representing different trade-offs between the underlying objectives.

Methods finding a single solution

FMOL algorithms designed to find a single solution aim to find an arbitrary Pareto-stationary solution. The advantage of such approaches is a relatively quick convergence, e.g. by exploiting gradients to locate the nearest solution. The main disadvantage is a lack of control over which solution out of all possible ones is found, and thus a lack of choice for potential users. One of the earliest such works [Yang *et al.*, 2023] once again extends the MGDA algorithm to the federated setting, this time with respect to client objectives. Local training sequentially updates client models with respect to each component objective. Then, clients submit a gradient vector for aggregation to the server, where MGDA yields optimal aggregation weights to update the global model. This algorithm is shown to converge to a Pareto-stationary solution. A subsequent work [Askin *et al.*, 2024] points out a risk of local drift in this approach, as well as a high communication load caused by transmitting separate gradient updates for all objectives. The algorithm proposed to mitigate these issues is also based on server-side MGDA, but clients reduce communication cost by transmitting a compressed matrix of all objective gradients. Local drift is avoided via a similar modification: client updates are computed from a linear combination of all objective gradients rather than a series of single-objective updates. Tackling a different use case, [Kinoshita *et al.*, 2024] discuss data-driven MOO problems, where a federated server attempts to solve a multi-objective problem, e.g. clustering, using only indirect information from distributed clients. In this unsupervised setting, no gradient-based strategies are possible; the server instead utilises a MOEA to solve the problem.

Methods finding multiple solutions

Federated algorithms designed to find multiple solutions have one of two goals: they either attempt (1) to find a full Pareto front, i.e. a set of trade-off solutions, or (2) to find a personalised model for each participant. For both variants, participants may have different preferences over the same objective functions, or may even be solving entirely disjoint tasks.

Finding a Pareto front. Algorithms that aim to find a Pareto front of solutions must explore a wide range of the search space to identify a diverse spread of trade-off solutions. In the distributed setting, this may happen at different levels of the federated system: *server-led* exploration sees the federated server managing the exploration and constructing a Pareto front. A first framework for such a scenario has been proposed in [Hartmann *et al.*, 2023], utilising a metaheuristic on the federated server to decompose the multi-objective problem in into single-objective candidate subproblems. This approach bears similarities to some of the top-level algorithms discussed in Section 3.2, in that each candidate is evaluated separately by a full federated system. Unlike those approaches, however, the full system is not strictly required for an effective evaluation. Thus, the efficiency of the evaluation could be improved by the use of an algorithm that can federate candidates with different objective preferences. To the best of our knowledge, such an algorithm has not yet been proposed in the literature. Future contributions may be able to leverage client-specific solution algorithms in combination with server-led Pareto exploration strategies.

In contrast, *client-led* exploration would have each client attempting to find a Pareto front, e.g. in cases where the server is untrusted or lacks computing resources. This scenario has, to the best of our knowledge, not yet been addressed in the the

Reference	Taxonomy label	Local MOO method	Global MOO method	Objectives
[Yang *et al.*, 2023]	single-solution	successive single-obj. updates	MGDA	arbitrary
[Askin *et al.*, 2024]	single-solution	linearised objectives	MGDA	arbitrary
[Hartmann *et al.*, 2023]	server-led	linearised objectives	offline metaheuristic	arbitrary
[Sen and Borcea, 2024]	multi-task	multi-task layer	similarity-based partial aggregation	arbitrary separable tasks
[Hartmann *et al.*, 2024]	preference-driven	linearised preferences	similarity-based aggregation+clustering	arbitrary

Table 2: Comparison of selected Federated Multi-objective Learning (FMOL) algorithms. Note that all algorithms are dedicated to handling local multi-objective learning. As noted in Section 2.3, this requires modifications at several levels of the federated system.

literature, but would carry its own challenges and opportunities inherent to the federated setting, most importantly a shift of control from server to clients, and the alignment of local Pareto fronts. Possibly related is the *fully-distributed* setting, where no server is involved in the training process and aggregation is decentralised across the client network.

Finding client-specific solutions. Here, the goal of the algorithm is to find a solution for each client in the system, based on different local requirements. Crucially, and in contrast to single-solution algorithms, this approach yields a different model for each client, matching that client's objectives, instead of finding a global model that generalises over all clients. This variant is known as Personalised FL, and is typically used in highly heterogeneous settings where the focus is on individual client performance [Tan *et al.*, 2023]. Note that this type of algorithm is arguably unique to the federated setting, arising from its properties that participants in FL are heterogeneous and may have different, independent interests.

In a *preference-driven* setting, client heterogeneity is induced by different preference weights assigned by each client to the same underlying multi-objective problem [Hartmann *et al.*, 2024]. Formally, the objectives of the i-th client are weighted by that client's unique preference weights w^i:

$$\vec{f^i}(x) := \vec{w}^i \odot \vec{f}(x) = (w_1^i f_1(x), \ldots, w_n^i f_n(x))^T \qquad (5)$$

Where objective components are conflicting, learning trajectories of clients could diverge even on the same underlying model; the PFL approach is intended to embrace this diversity instead of counteracting it. Only a handful of works so far have considered a personalised approach to objective heterogeneity. In the first such work [Hartmann *et al.*, 2024], client preferences are assumed to be private, and local training is performed on a weighted linear combination of the objectives. The challenge in this setting is to aggregate clients whose current training trajectory is compatible, and separate clients where it is not. As little direct information about the mutual compatibility of clients is available on the server, many classical MOO methods cannot be applied. Instead, the proposed algorithm performs clustering and weighted aggregation based on the similarity of model updates.

Federated *multi-task* learning is an edge case scenario where clients solve mutually different subsets of tasks (i.e. objectives). A number of works in the FL literature, e.g. [Ghosh *et al.*, 2020] and [Huang *et al.*, 2023], have addressed a simplified setting where each client is assigned a single task[1] without acknowledging a multi-objective perspective. To the best of our knowledge, only one work currently considers the problem where each client is assigned a set of several tasks [Sen and Borcea, 2024]. Similarly to other works on FMOL, this task assignment is private. Under the proposed algorithm, clients jointly train a block of shared model parameters plus a separate parallel model layer for each task to be solved by the client. Once again, clients are aggregated based on a model similarity score, computed here based both on the shared parameters and a matching of task-specific layers.

4 Conclusion and perspectives

In this work, we have presented the first comprehensive survey on the use of multi-objective methods in connection with Federated Learning. We have proposed a novel taxonomy to classify existing works in the literature, and offered a perspective on recent trends, open challenges and possible approaches. Existing work demonstrates that MOO is a promising tool to improve transparency and effectiveness of FL techniques when navigating real-world problems. As in the wider field of FL, further work remains to be done. Open avenues of research in MO-FL include, most prominently, (i) effective defence against malicious attackers in multi-objective aggregation; (ii) the use of MOO methods specifically to recognise low-quality clients; (iii) enhancing transparency and control of MO-preferences for users, e.g. by generating multiple different Pareto-optimal solutions, and (iv) exploring more sophisticated MOO techniques, e.g. to replace the baseline NSGA-II algorithm that is currently used in many of the works discussed here. The area of FMOL, enabling the federated solving of multi-objective learning problems, remains largely open. Initial contributions to the field could include, for example, (v) improving the efficiency of server-led strategies finding a Pareto front; (vi) exploring the effect of preference heterogeneity on convergence in single- and multi-solution algorithms; (vii) exploring the cumulative effect of data heterogeneity on FMOL problems; (viii) considering variant FMOL settings, e.g. where client preferences are not private.

[1] Note that the 'multi-task' label is assigned inconsistently in the existing FL literature, referring variously to clients with heterogeneous datasets or objectives.

Acknowledgements

This work is partially funded by the joint research programme UL/SnT–ILNAS on Technical Standardisation for Trustworthy ICT, Aerospace, and Construction. The authors would like to thank Gwen Maudet for proofreading this work.

References

[Askin *et al.*, 2024] Baris Askin, Pranay Sharma, Gauri Joshi, and Carlee Joe-Wong. Federated communication-efficient multi-objective optimization, 2024.

[Badar *et al.*, 2024] Maryam Badar, Sandipan Sikdar, Wolfgang Nejdl, and Marco Fisichella. Fairtrade: Achieving pareto-optimal trade-offs between balanced accuracy and fairness in federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):10962–10970, March 2024.

[Banerjee *et al.*, 2022] Sourasekhar Banerjee, Xuan-Son Vu, and Monowar Bhuyan. Optimized and adaptive federated learning for straggler-resilient device selection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2022.

[Brik *et al.*, 2020] Bouziane Brik, Adlen Ksentini, and Maha Bouaziz. Federated learning for uavs-enabled wireless networks: Use cases, challenges, and open problems. *IEEE Access*, 8:53841–53849, 2020.

[Chai *et al.*, 2022] Zheng-yi Chai, Chuan-dong Yang, and Ya-lun Li. Communication efficiency optimization in federated learning based on multi-objective evolutionary algorithm. *Evolutionary Intelligence*, 16(3):1033–1044, April 2022.

[Che *et al.*, 2023] Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor Sheng, Huaiyu Dai, and Dejing Dou. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7888, Singapore, December 2023. Association for Computational Linguistics.

[Cui *et al.*, 2021] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 26091–26102. Curran Associates, Inc., 2021.

[Deb *et al.*, 2002] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[Désidéri, 2012] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012.

[Geng *et al.*, 2024] Daoqu Geng, Shouzheng Wang, and Yihang Zhang. Multi-objective federated averaging algorithm. *Expert Systems*, November 2024.

[Ghosh *et al.*, 2020] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Hartmann *et al.*, 2023] Maria Hartmann, Grégoire Danoy, Mohammed Alswaitti, and Pascal Bouvry. MOFL/D: A federated multi-objective learning framework with decomposition. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[Hartmann *et al.*, 2024] Maria Hartmann, Grégoire Danoy, and Pascal Bouvry. Fedpref: Federated learning across heterogeneous multi-objective preferences. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, December 2024. Just Accepted.

[Hu *et al.*, 2022] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9(4):2039–2051, July 2022.

[Huang *et al.*, 2023] Zhi-An Huang, Yao Hu, Rui Liu, Xiaoming Xue, Zexuan Zhu, Linqi Song, and Kay Chen Tan. Federated multi-task learning for joint diagnosis of multiple mental disorders on mri scans. *IEEE Transactions on Biomedical Engineering*, 70(4):1137–1149, April 2023.

[Ju *et al.*, 2024] Li Ju, Tianru Zhang, Salman Toor, and Andreas Hellander. Accelerating fair federated learning: Adaptive federated adam. *IEEE Transactions on Machine Learning in Communications and Networking*, 2:1017–1032, 2024.

[Kairouz *et al.*, 2021] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[Kang *et al.*, 2024a] Yan Kang, Hanlin Gu, Xingxing Tang, Yuanqin He, Yuzhu Zhang, Jinnan He, Yuxing Han, Lixin Fan, Kai Chen, and Qiang Yang. Optimizing privacy, utility, and efficiency in a constrained multi-objective federated learning framework. *ACM Trans. Intell. Syst. Technol.*, 15(6), December 2024.

[Kang *et al.*, 2024b] Yan Kang, Ziyao Ren, Lixin Fan, Linghua Yang, Yongxin Tong, and Qiang Yang. Hyper-parameter optimization for SecureBoost via constrained multi-objective federated learning, 2024.

[Karimireddy *et al.*, 2019] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 2019.

[Kinoshita *et al.*, 2024] Takato Kinoshita, Naoki Masuyama, and Yusuke Nojima. A federated data-driven multiobjective evolutionary algorithm via continual learnable clustering. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, page 1–7. IEEE, June 2024.

[Li *et al.*, 2020] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.

[Mehrabi *et al.*, 2022] Ninareh Mehrabi, Cyprien de Lichy, John McKay, Cynthia He, and William Campbell. Towards multi-objective statistically fair federated learning, 2022.

[Mohri *et al.*, 2019] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR, 09–15 Jun 2019.

[Morell *et al.*, 2024] José Ángel Morell, Zakaria Abdelmoiz Dahi, Francisco Chicano, Gabriel Luque, and Enrique Alba. A multi-objective approach for communication reduction in federated learning under devices heterogeneity constraints. *Future Generation Computer Systems*, 155:367–383, June 2024.

[Sen and Borcea, 2024] Pritam Sen and Cristian Borcea. Fedmtl: Privacy-preserving federated multi-task learning. In *ECAI 2024*. IOS Press, October 2024.

[Shen *et al.*, 2025] Yuhao Shen, Wei Xi, Yunyun Cai, Yuwei Fan, He Yang, and Jizhong Zhao. Multi-objective federated learning: Balancing global performance and individual fairness. *Future Generation Computer Systems*, 162:107468, January 2025.

[Talbi, 2009] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.

[Tan *et al.*, 2023] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, December 2023.

[Yang *et al.*, 2023] Haibo Yang, Zhuqing Liu, Jia Liu, Chaosheng Dong, and Michinari Momma. Federated multi-objective learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Ye and Liu, 2022] Mao Ye and Qiang Liu. Pareto navigation gradient descent: a first-order algorithm for optimization in pareto set. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 2246–2255. PMLR, 01–05 Aug 2022.

[Yin *et al.*, 2023] Benshun Yin, Zhiyong Chen, and Meixia Tao. Predictive gan-powered multi-objective optimization for hybrid federated split learning. *IEEE Transactions on Communications*, 71(8):4544–4560, August 2023.

[Zhu and Jin, 2020] Hangyu Zhu and Yaochu Jin. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1310–1322, April 2020.

[Zhu and Jin, 2022] Hangyu Zhu and Yaochu Jin. Real-time federated evolutionary neural architecture search. *IEEE Transactions on Evolutionary Computation*, 26(2):364–378, 2022.

[Zhu *et al.*, 2021] Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, 7(2):639–657, January 2021.