# MOFL/D: A Federated Multi-objective Learning Framework with Decomposition


**Maria Hartmann**
SnT, University of Luxembourg
Luxembourg
maria.hartmann@uni.lu

**Grégoire Danoy**
FSTM/DCS, University of Luxembourg
SnT, University of Luxembourg
Luxembourg
gregoire.danoy@uni.lu

**Mohammed Alswaitti**
SnT, University of Luxembourg
Luxembourg
mohammed.alwaitti@uni.lu

**Pascal Bouvry**
FSTM/DCS, University of Luxembourg
SnT, University of Luxembourg
Luxembourg
pascal.bouvry@uni.lu



## Abstract

Multi-objective learning problems occur in all aspects of life and have been studied for decades, including in the field of machine learning. Many such problems also exist in distributed settings, where data cannot easily be shared. In recent years, joint machine learning has been made possible in such settings through the development of the Federated Learning (FL) paradigm. However, there is as of now very little research on the general problem of extending the FL concept to multi-objective learning, limiting such problems to non-cooperative individual learning. We address this gap by presenting a general framework for multi-objective FL, based on decomposition (MOFL/D). Our framework addresses the *a posteriori* type of multi-objective problem, where user preferences are not known during the optimisation process, allowing multiple participants to jointly find a set of solutions, each optimised for some distribution of preferences. We present an instantiation of the framework and validate it through experiments on a set of multi-objective benchmarking problems that are extended from well-known single-objective benchmarks.


## 1 Introduction

Federated Learning (FL) is a distributed machine learning paradigm that is effective in settings where training data originates in distribution and cannot be shared. Reasons for such an isolation of datasets are manifold, including privacy concerns, proprietary interests or technological constraints limiting communication between learners. Federated Learning circumvents these constraints to allow participants to nevertheless cooperate in training by exchanging data only about the training process, typically in the form of model or gradient updates. Since its inception in 2017, Federated Learning has advanced in leaps and bounds, quickly finding use in applications across the industry [23].


International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023 (FL@FM-NeurIPS'23). This workshop does not have official proceedings and this paper is non-archival.

However, as the scope of potential FL applications expands and new machine learning paradigms become established, combining these advances poses new algorithmic challenges. Multi-objective problems arise naturally in many real-world settings, from resource management to mobility control problems, and may in many cases be tackled through the use of machine learning approaches such as reinforcement learning.Where such problems exist in a distributed setting and are subject to constraints on information sharing, applying Federated Learning appears natural; yet no extension of the FL paradigm to the solution of general multi-objective problems appears to have been proposed up to now.

In this work, we present a novel systematic treatment of multi-objective learning in a federated setting, considering the class of problems where user preferences are not defined at computing time. This class is known as the class of *a posteriori* problems. We propose a first general multi-objective federated learning framework based on decomposition (MOFL/D) to solve such problems, combining ideas from the fields of multi-objective optimisation and federated learning. We demonstrate one possible instantiation of this framework and validate its performance on a number of multi-objective variants of well-established single-objective benchmarks, with additional results shown in the appendix of this work. In doing so, we open up the field of federated learning to a new class of learning problems.

## 2 Related Work

The main contribution, a first general framework for federated multi-objective learning, combines elements of federated learning (FL) and multi-objective optimisation (MOO), specifically multi-objective optimisation with decomposition (MOO/D).

Significant previous research exists on the topic of multi-objective machine learning (MOML), with the large majority of contributions focused on optimising the hyperparameters of a machine learning algorithm alongside an underlying single-objective problem[13][2][20]. Other works tackle the extension of specific algorithms to the multi-objective case, e.g. [10]and [24]. However, despite the prevalence of such problems and the existing research on MOML, there appears to be no previous research on the integration of multi-objective learning into the FL paradigm. Therefore, we begin this direction of research by formulating a framework that utilises concepts from the field of MOO itself, to allow a later systematic integration of existing approaches from related fields.
The problem of multi-objective optimisation has been studied for decades [19]. Problems can be classified into those where user preferences are known at the time of optimisation, providing an ordering between objectives, known as *a priori* problems, and *a posteriori* problems, where preferences are unknown. One approach of solving such multi-objective problems is by decomposition (MOO/D) – a common decomposition method is to scalarise the set of objectives to obtain a single-objective problem, with different scalarisations producing different subproblems. Here we choose a linear scalarisation approach.

Some previous literature exists on the application of multi-objective concepts to federated learning. These works fall into one of two categories: treating the performance of each client as a separate objective, aiming to optimise the system for 'fairness' in some sense [7], [3], [12]; or considering multiple objectives on the global system, such as maximising performance of the global model while minimising model size [28], [9]. Finally, a recent work by Yang et al. [25] is, to the best of our knowledge, the only other existing work to tackle federated multi-objective learning in a general way. Their work, however, differs from ours in several important respects: First, their setting assumes that the knowledge of each client is permanently limited to a subset of all relevant objectives. In our work, we assume that all objectives are known to all clients, and that clients are capable of modifying their preferences over these objectives. Second, their framework is designed to find only a single global solution to the multi-objective problem. Our work generates a set of solutions representing different trade-offs between the objectives, allowing for the selection of solutions based on different priorities without the need to recompute. Third, their framework is based on multi-gradient descent, whereas we rely on a decomposition approach.

## 3 Description of the MOFL/D framework

In this section, we shall first introduce and formalise relevant concepts from Federated Learning and multi-objective optimisation; then we will present the general MOFL/D framework.

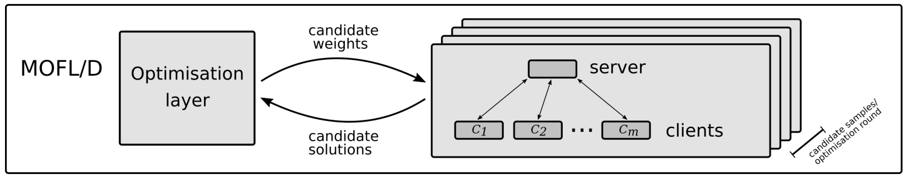



Figure 1: A high-level depiction of the theoretical MOFL/D framework.

### 3.1 Background

In the FL setting, a set of $n$ training samples $\mathcal{P}$ is partitioned into $m$ subsets $\mathcal{P}_1, \ldots \mathcal{P}_m$, with each $\mathcal{P}_i$ privately owned by a client $\mathcal{C}_i$. Each dataset cannot be shared outside of the client that owns it. Let $|\mathcal{P}_i| = n_i$ be the size of the $i$-th training set. In this work, we consider the classical horizontal FL setting as defined in [26], where all clients observe the same features and client model architectures are homogeneous. Though classical Federated Learning[11] was formulated to learn a global model $\omega$ that optimises a single objective function, here we assume instead that each client is optimising a *vector* of objectives $\vec{f}_i$. In the spirit of the assumptions made in the horizontal FL setting, we assume that all $m$ clients optimise the same set of objectives; so the formulation of the FL problem becomes

$$\vec{f}(\omega) = \sum_{i=0}^{m} \frac{n_i}{n} \vec{F}_i(\omega), \text{ where } \vec{F}_i(\omega) := \frac{1}{n_i} \sum_{p \in \mathcal{P}_i} \vec{f}_p(\omega). \tag{1}$$

In the absence of a pre-defined hierarchy of objectives, the set of solutions to this problem is a partially ordered set, as the value of different objectives is not comparable in terms of overall optimality. In such cases, with preferences unknown during the optimisation process, a common MOO approach is to find a set of solutions, each representing an optimal trade-off between objectives. We say that a solution $v$ *Pareto dominates* another solution $u$ iff it improves the value of at least one objective while matching or improving all others. In formal terms, we hold for a maximisation problem:

$$v \succ_p u \iff \exists i : f_i(v) > f_i(u) \land \forall j : f_j(v) \geq f_j(u).$$

The *Pareto front* $\mathcal{PF}$ of a set of solutions $\mathcal{S}$ is then defined as the subset of all solutions that are not Pareto dominated by any other solution:

$$\mathcal{PF}(\mathcal{S}) := \{\, v \in \mathcal{S} \,|\, \neg\exists u \in \mathcal{S} : u \succ_p v \,\}.$$

### 3.2 The MOFL/D framework

The overall goal of our MOFL/D framework is to find a set $\mathcal{M}$ of solution models, using the federated system, that together approximate the Pareto front of the objective space. In abstract terms, this may be modelled as shown in Figure 1: A federated system consists of multiple participants, coordinated by a server, with each participant learning to optimise an MO learning problem as defined in Eq. 1, using a given scalarisation. An optimisation layer is added on top and given control of the federated system in order to manage the overall optimisation process. This optimisation layer carries out three tasks: (1) decomposing the MOL problem into candidate sub-problems by generating scalarisation weights, (2) managing the federated system to compute candidate solutions to each scalarisation provided by the optimisation layer and (3) maintaining a set of optimal solutions out of the candidate solutions returned by the federated system.

At the beginning of each round, the optimisation layer generates a set of scalarisation weights to map the multi-objective problem to single-objective subproblems. The choice of candidate weights is governed by a metaheuristic method, making inferences from the results of previous optimisation rounds. (Note that this framework places no restrictions on the choice of multi-objective solver; any suitable method from conventional MOO may be used as a drop-in replacement.)

To solve the candidate problems generated thus, the optimisation layer invokes the federated system. A candidate weight is passed to the federated system, which executes a full FL cycle, computing a candidate solution to the scalarised problem. Once the federated system converges, the resulting model is passed back to the optimisation layer. This process is repeated for all candidates. For the sake of simplicity, we take a naïve approach in this first work, re-initialising the entire federated system for each subproblem and solving all subproblems in sequence. However, the question of how

to use the federated system more effectively is a natural next step to continue our research.
Finally, the optimisation layer updates the current set of Pareto-optimal solutions discovered thus far, incorporating the results obtained from this most recent candidate generation. Depending on the choice of metaheuristic, a spearate set of 'generating solutions' may also be maintained and updated at this stage, used to generate new candidate solutions or base models for initialisation. This optimisation cycle is repeated until a termination condition defined by the metaheuristic is met.

In addition to this main approach of generating scalar weights, we also propose the possibility of generating an initial base model for each single-objective problem, used to warm-start the federated training process. Previous works[18], [14] have shown that FL tolerates, and may benefit from, initialisation with a pre-trained model chosen with sufficient care. We suggest that a base model could be derived from the solution obtained for a previous subproblem that is 'sufficiently close' to the current problem - a straightforward approach to quantifying problem similarity in this framework is to use the distance between the respective scalar weights used to generate each subproblem.

**Practical considerations on the federated system**
Translating the abstract MOFL/D framework into an implementation requires us to make two practical choices with respect to the federated system. The first choice is the implementation of the optimisation layer. In the preceding theoretical discussion of the framework, we have treated the high-level optimisation aspects of the algorithm as a fully separate layer; however, we note that in practice the optimisation layer may be integrated with the server functionality of the FL system.
A second point to consider is the evaluation of candidate solutions. In a classical federated system, training samples are typically only available to clients; in this case the final evaluation of any candidate solution would need to be performed on the client-side. This approach has the advantages of preserving data privacy and spreading the computational load of evaluation. However, the resulting estimate may not be representative of the system if the distribution of client data is skewed, and the self-reporting of solution values places a level of trust in clients that may be exploited by a malicious participant. Another approach also taken by some previous works, e.g. [12], is to require a validation dataset to be known to the server; we follow this approach in our demonstration of the framework.

**Input:** Number of iterations $n_i$, number of samples $n_s$, number of federated clients $n_c$
Pareto front $\mathcal{PF}_0 \leftarrow \{\}$;
Pareto front models $\mathcal{PFM}_0 \leftarrow \{\}$;
$t \leftarrow 0$;
**while** $t < n_i$ **do**
    $\mathcal{W}_t \leftarrow$ generate $n_s$ candidate weights;
    $\mathcal{V}_t, \mathcal{M}_t \leftarrow \{\}, \{\}$;
    **foreach** $w \in \mathcal{W}_c$ **do**
        $\theta_0^w \leftarrow$ generate initial candidate model;
        `/* Train federated system to completion to obtain global model */`
        $\theta^w \leftarrow$ run Fed-Server with $\theta^w, w$;
        $\vec{v} \leftarrow$ evaluate $\theta^w$ for all objectives;
        append $\theta^w, \vec{v}$ to $\mathcal{M}_t, \mathcal{V}_t$;
    **end**
    $\mathcal{PF}_{t+1} \leftarrow \mathcal{PF}_t \cup \mathcal{V}_t$;
    $\mathcal{PFM}_{t+1} \leftarrow$ models generating $\mathcal{PF}_{t+1}$;
    $t \leftarrow t + 1$;
**end**

**Algorithm 1:** MOFL/D

## 4 Experiments

In this section, we demonstrate an experimental validation of our MOFL/D framework on a number of multi-objective reinforcement learning (MORL) problems. We begin by providing a brief overview of the state of the art in the field of federated reinforcement learning; then we detail our choices regarding the instantiation and implementation of the framework. Finally, we discuss the design of the experiments performed and analyse our results.

Table 1: Instantiation and implementation choices for the experimental validation of MOFL/D.

| Component | Instantiation | Implementation resources |
|---|---|---|
| Federated Algorithm | DQNAvg [8] | morl-baselines [5]<br>stable-baselines3 [17] |
| Learning Problems | Deep-Sea Treasure (DST) [21]<br>Deterministic Minecart (DMC) [1]<br>Multi-objective Lunar Lander (MOLL) | mo-gymnasium [6] |
| Metaheuristic | Pareto Simulated Annealing [4] | None used |

A number of recent works study the application of FL to single-objective reinforcement learning[15]. Zhuo et al. [29] propose an algorithm that learns a secondary model to approximate the Q-network values of all clients without exposing their true networks. In [27], multiple clients with different fixed preferences perform federated learning to obtain a generalised critic for carrying out local actor-critic reinforcement learning. While this work is one of the few where each client attempts to optimise multiple objectives, the proposed algorithm does not yield a Pareto front. Each client joining the learning process must train its own actor model from scratch. Furthermore, it is not clear how this approach to federalising the training would generalise to other types of RL or non-RL algorithms.
Finally, Jin et al. [8] propose two algorithms, QAvg and PAvg, that extend the vanilla federated averaging (FedAvg)[11] for use with Q-networks and policy networks, respectively.

### 4.1 MOFL/D Instantiation and implementation

We evaluate our framework on a number of multi-objective reinforcement learning problems, as this class of problems represents a natural type of MO problem well-suited to application,possesses a number of well-defined benchmarks, and existing literature provides a straightforward FL algorithm for single-objective reinforcement learning problems[8].
Where possible, we make choices that resemble as closely as possible the equivalent baselines commonly chosen for demonstrations in the respective field of research; otherwise we choose methods based on their simplicity and ease of reproduction. A comprehensive overview of instantiation choices and applicable libraries used in the implementation is given in Table 1. The complete set of parameters chosen for all experiments is reported in the appendix. Noting that few reference parameterisations for MORL algorithms exist in the literature, we have, where available, tested parameterisations for the related single-objective problems from the rl-baselines3-zoo [16] benchmarking project; however, these did not always prove suitable to the MO extension of the problem. Where no suitable parameterisation could be derived from the literature, parameters were tuned manually.

### 4.2 Experiment design

We focus on investigating the impact of varying parameters of the federated system on the overall performance of the framework. We run experiments on each environment with two, three, and five clients in federation. In addition, we run the algorithm with the same configuration on single-client systems with no communication to obtain a baseline performance of the non-federated system. We also investigate the impact of the duration of the local training phase in the federated system, comparing runs with a local training phase duration of $2000, 5000$ and $10000$ iterations. Finally, we contrast the performance of the algorithm on a federated system using pre-trained models and a federated system following the conventional approach of training models from scratch. We repeat experiments multiple times for each parameter combination, using different random seeds. All experiments on two- and three-client systems are repeated ten times, with the number of runs reduced to five for five-client systems in deference to the high computational cost of these experiments. Detailed information about the choice of random seeds for all experiments may be found in the appendix. We evaluate the performance of our framework using three common multi-objective metrics [30]: the hypervolume defined by our non-dominated solution set, the sparsity of the solution set, and the inverted generational distance (IGD), using the set of all solutions to approximate the true Pareto front. These metrics quantify different properties of the solution set: sparsity measures the diversity of the solution set, IGD the convergence and hypervolume size a combination of both. A desirable set of solutions would have low sparsity and IGD values and a large hypervolume.

### 4.3 Selected Results and Discussion

Numerical results are shown in Table 2. For all three learning environments, we consistently observe that the MOFL/D algorithm run with a federated system matches, and for the more complex problems outperforms, the same heuristic run with a non-federated system. This demonstrates both the general potential of federating the training of MO learning problems, and the validity of our framework. In more detailed terms, we observe a significantly increased hypervolume value along with a decreased sparsity in the results generated by running MOFL/D on multi-client systems, compared to a single client, on the two more complex MO-Lunar Lander (MOLL) and Deterministic Minecart (DMC) environments - see Figure 2 for an example of the observed hypervolume evolution (Fig. 2a and associated solutions(Fig. 2b). The ultimate hypervolume values obtained for the Deep-Sea Treasure (DST) environment are similar for all federated systems and the non-federated system; this can likely be explained by the simplicity of the environment, with its very limited number of optimal solutions. On the more complex environments we also observe a tendency for systems with a higher number of clients to find solution sets with greater hypervolume and lower sparsity. The impact of length of local training phase appears dependent on both the complexity of problem and number of clients in the federated system, with differing qualitative results for different environments. Finally, we observe no clear result on the benefits of re-using results to warm-start new training rounds: the ultimate performance of the system relative to non-pre-trained models differs across environments, with improvements in some and reduced performance in other cases.

## 5 Conclusion and Outlook

In this work, we have presented a novel general framework to solve inherently multi-objective problems in a Federated Learning setting. To the best of our knowledge, this is the first work to consider the general case of federated multi-objective learning and present a systematic approach to solving it using decomposition. We have discussed instantiation choices for the framework and shown one such instantiation. Using this instantiation, we have performed experiments on three well-founded benchmarking problems from the MORL field, showing the validity of our framework and investigating the effect of several variable parameters related to the federated system. Potential further work includes investigating other, more complex possible instantiations and application to different types of multi-objective problems.

## Acknowledgements

This work is partially funded by the joint research programme UL/SnT–ILNAS on Technical Standardisation for Trustworthy ICT, Aerospace, and Construction.
The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [22] – see `https://hpc.uni.lu`.


(a) Hypervolume evolution compared for systems with variable numbers of federated clients.

(b) Sample Pareto fronts obtained with the same configuration across variable numbers of clients.

Figure 2: Impact of different parameters of the federated system on hypervolume development for MOFL/D run on the MO-Lunar Lander environment. Experiments run with 10000 local steps per federated round and without pre-trained models.

Table 2: Numerical results of experiments on each benchmarking environment. Hypervolume and sparsity metrics are reported here; see the appendix for the corresponding values of the IGD metric. Each entry reports the mean value of the respective metric, with the variance in parentheses. Higher hypervolume values and lower sparsity values, respectively, correspond to better performance.

| Conf. | Hypervolume | | | Sparsity | | |
|---|---|---|---|---|---|---|
| $n_c/n_i^f/ws$ | DST | DMC | MOLL ($\cdot 10^{-7}$) | DST | DMC | MOLL ($\cdot 10^{-1}$) |
| $2/2k/T$ | 992.3(2.4) | 896.8(33.4) | 403.7(8.3) | 17.9(3.0) | 1.0(0.5) | 353.5(634.3) |
| $2/2k/F$ | 970.8(39.9) | 932.9(17.9) | 404.6(8.2) | 21.8(9.0) | 1.5(2.9) | 50.9(85.1) |
| $2/5k/T$ | 973.6(33.1) | 867.6(58.2) | 399.7(8.0) | 22.3(11.8) | 1.5(0.8) | 95.6(166.0) |
| $2/5k/F$ | 990.8(3.3) | 936.9(11.6) | 405.2(11.3) | 19.6(3.3) | 0.5(0.2) | 30.6(23.5) |
| $2/10k/T$ | 990.3(4.5) | 854.6(58.1) | 405.4(7.5) | 19.8(3.9) | 1.6(0.8) | 141.8(338.8) |
| $2/10k/F$ | 985.8(14.7) | 932.0(11.5) | 404.0(10.7) | 22.1(10.4) | 0.5(0.2) | 30.7(20.0) |
| $3/2k/T$ | 984.5(12.2) | 869.0(60.0) | 410.2(15.1) | 26.1(12.8) | 1.4(0.9) | 108.3(247.2) |
| $3/2k/F$ | 986.6(10.7) | 940.5(7.1) | 405.2(8.3) | 24.2(11.4) | 0.4(0.1) | 52.5(55.1) |
| $3/5k/T$ | 990.8(3.1) | 893.6(49.7) | 402.9(9.3) | 20.0(4.9) | 1.1(0.7) | 210.1(565.0) |
| $3/5k/F$ | 974.8(40.0) | 935.7(9.0) | 406.0(6.0) | 23.0(12.0) | 0.5(0.1) | 20.9(7.6) |
| $3/10k/T$ | 987.3(10.7) | 819.2(44.8) | 407.9(15.0) | 22.0(7.6) | 2.1(0.6) | 104.2(134.0) |
| $3/10k/F$ | **993.9** (1.3) | 908.2(39.0) | 412.3(11.7) | **15.8** (1.2) | 0.9(0.6) | 51.3(66.0) |
| $5/2k/T$ | 974.8(28.4) | 908.0(1.9) | **425.0** (6.8) | 46.1(46.1) | 0.9(0.0) | 68.9(104.2) |
| $5/2k/F$ | 988.2(9.1) | **941.1** (7.2) | 408.1(6.7) | 21.3(8.5) | **0.4** (0.1) | 14.9(5.3) |
| $5/5k/T$ | 985.5(10.7) | 890.0(47.1) | 420.5(14.0) | 27.4(13.9) | 1.1(0.7) | 57.4(67.1) |
| $5/5k/F$ | 991.4(2.6) | 936.5(16.3) | 411.2(11.7) | 19.0(2.9) | 0.5(0.2) | **12.9** (1.8) |
| $5/10k/T$ | 989.8(3.8) | 886.4(44.8) | 413.3(14.1) | 22.1(5.8) | 1.2(0.6) | 207.4(276.3) |
| $5/10k/F$ | 992.1(3.3) | 923.1(13.5) | 421.4(7.8) | 17.8(3.5) | 0.7(0.2) | 23.2(16.9) |
| Non-fed. | 983.1(39.2) | 879.8(73.3) | 388.7(8.5) | 35.7(116.9) | 1.3(0.9) | 108.5(147.9) |

## References


[1] Axel Abels et al. *Dynamic Weights in Multi-Objective Deep Reinforcement Learning*. 13th May 2019. arXiv: 1809.07803[cs, stat]. URL: http://arxiv.org/abs/1809.07803 (visited on 08/09/2023).

[2] Stamatios-Aggelos Alexandropoulos et al. 'Multi-Objective Evolutionary Optimization Algorithms for Machine Learning: A Recent Survey'. In: May 2019, pp. 35–55. ISBN: 978-3-642-54671-6. DOI: 10.1007/978-3-030-12767-1_4.

[3] Sen Cui et al. 'Addressing Algorithmic Disparity and Performance Inconsistency in Federated Learning'. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 26091–26102. URL: https://proceedings.neurips.cc/paper/2021/hash/db8e1af0cb3aca1ae2d0018624204529-Abstract.html (visited on 13/07/2023).

[4] Piotr Czyzżak and Adrezej Jaszkiewicz. 'Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization'. In: *Journal of Multi-Criteria Decision Analysis* 7.1 (1998). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291099-1360%28199801%297%3A1%3C34%3A%3AAID-MCDA161%3E3.0.CO%3B2-6, pp. 34–47. ISSN: 1099-1360. DOI: 10.1002/(SICI)1099-1360(199801)7:1<34::AID-MCDA161>3.0.CO;2-6. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-1360%28199801%297%3A1%3C34%3A%3AAID-MCDA161%3E3.0.CO%3B2-6 (visited on 02/08/2023).

[5] Florian Felten and Lucas N. Alegre. *MORL-Baselines: Multi-Objective Reinforcement Learning algorithms implementations*. https://github.com/LucasAlegre/morl-baselines. 2022.

[6] Florian Felten et al. ‘A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning’. en. In: *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*. 2023. (Visited on 22/09/2023).

[7] Zeou Hu et al. ‘Federated Learning Meets Multi-Objective Optimization’. In: *IEEE Transactions on Network Science and Engineering* 9.4 (July 2022). Conference Name: IEEE Transactions on Network Science and Engineering, pp. 2039–2051. ISSN: 2327-4697. DOI: 10.1109/TNSE.2022.3169117.

[8] Hao Jin et al. ‘Federated Reinforcement Learning with Environment Heterogeneity’. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. ISSN: 2640-3498. PMLR, 3rd May 2022, pp. 18–37. URL: https://proceedings.mlr.press/v151/jin22a.html (visited on 28/02/2023).

[9] Yan Kang et al. *Optimizing Privacy, Utility and Efficiency in Constrained Multi-Objective Federated Learning*. 9th May 2023. DOI: 10.48550/arXiv.2305.00312. arXiv: 2305.00312[cs]. URL: http://arxiv.org/abs/2305.00312 (visited on 13/07/2023).

[10] S. Liu and L. N. Vicente. ‘The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning’. In: *Annals of Operations Research* (17th Mar. 2021). ISSN: 1572-9338. DOI: 10.1007/s10479-021-04033-z. URL: https://doi.org/10.1007/s10479-021-04033-z (visited on 18/09/2023).

[11] Brendan McMahan et al. ‘Communication-Efficient Learning of Deep Networks from Decentralized Data’. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1273–1282. URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

[12] Ninareh Mehrabi et al. *Towards Multi-Objective Statistically Fair Federated Learning*. 24th Jan. 2022. DOI: 10.48550/arXiv.2201.09917. arXiv: 2201.09917[cs]. URL: http://arxiv.org/abs/2201.09917 (visited on 13/07/2023).

[13] Alejandro Morales-Hernández, Inneke Van Nieuwenhuyse and Sebastian Rojas Gonzalez. ‘A survey on multi-objective hyperparameter optimization algorithms for machine learning’. In: *Artificial Intelligence Review* 56.8 (1st Aug. 2023), pp. 8043–8093. ISSN: 1573-7462. DOI: 10.1007/s10462-022-10359-2. URL: https://doi.org/10.1007/s10462-022-10359-2 (visited on 18/09/2023).

[14] John Nguyen et al. ‘Where to Begin? On the Impact of Pre-Training and Initialization in Federated Learning’. In: Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022). 22nd Nov. 2022. URL: https://openreview.net/forum?id=zE9ctlWm5lx (visited on 07/02/2023).

[15] Jiaju Qi et al. *Federated Reinforcement Learning: Techniques, Applications, and Open Challenges*. 24th Oct. 2021. DOI: 10.48550/arXiv.2108.11887. arXiv: 2108.11887[cs]. URL: http://arxiv.org/abs/2108.11887 (visited on 28/02/2023).

[16] Antonin Raffin. *RL Baselines3 Zoo*. https://github.com/DLR-RM/rl-baselines3-zoo. 2020.

[17] Antonin Raffin et al. ‘Stable-Baselines3: Reliable Reinforcement Learning Implementations’. In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.

[18] Felix Sattler, Klaus-Robert Müller and Wojciech Samek. ‘Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints’. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.8 (Aug. 2021). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 3710–3722. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2020.3015958.

[19] Shubhkirti Sharma and Vijay Kumar. ‘A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future’. In: *Archives of Computational Methods in Engineering* 29.7 (1st Nov. 2022), pp. 5605–5633. ISSN: 1886-1784. DOI: 10.1007/s11831-022-09778-9. URL: https://doi.org/10.1007/s11831-022-09778-9 (visited on 18/09/2023).

[20] Peter Súkeník and Christoph H. Lampert. *Generalization In Multi-Objective Machine Learning*. 2022. arXiv: 2208.13499 [cs.LG].

[21] Peter Vamplew et al. 'Empirical evaluation methods for multiobjective reinforcement learning algorithms'. In: *Machine Learning* 84.1 (1st July 2011), pp. 51–80. ISSN: 1573-0565. DOI: 10.1007/s10994-010-5232-5. URL: https://doi.org/10.1007/s10994-010-5232-5 (visited on 11/09/2023).

[22] S. Varrette et al. 'Management of an Academic HPC & Research Computing Facility: The ULHPC Experience 2.0'. In: *Proc. of the 6th ACM High Performance Computing and Cluster Technologies Conf. (HPCCT 2022)*. Fuzhou, China: Association for Computing Machinery (ACM), 2022. ISBN: 978-1-4503-9664-6.

[23] Jie Wen et al. 'A survey on federated learning: challenges and applications'. In: *International Journal of Machine Learning and Cybernetics* 14.2 (1st Feb. 2023), pp. 513–535. ISSN: 1868-808X. DOI: 10.1007/s13042-022-01647-y. URL: https://doi.org/10.1007/s13042-022-01647-y (visited on 13/07/2023).

[24] Fangjie Yang et al. 'PMDRL: Pareto-front-based multi-objective deep reinforcement learning'. In: *Journal of Ambient Intelligence and Humanized Computing* 14.9 (1st Sept. 2023), pp. 12663–12672. ISSN: 1868-5145. DOI: 10.1007/s12652-022-04232-x. URL: https://doi.org/10.1007/s12652-022-04232-x (visited on 18/09/2023).

[25] Haibo Yang et al. *Federated Multi-Objective Learning*. 15th Oct. 2023. arXiv: 2310.09866[cs]. URL: http://arxiv.org/abs/2310.09866 (visited on 30/10/2023).

[26] Qiang Yang et al. 'Federated Machine Learning: Concept and Applications'. In: *ACM Trans. Intell. Syst. Technol.* 10.2 (2019). ISSN: 2157-6904. DOI: 10.1145/3298981. URL: https://doi.org/10.1145/3298981.

[27] Fangyuan Zhao et al. 'Federated multi-objective reinforcement learning'. In: *Information Sciences* 624 (1st May 2023), pp. 811–832. ISSN: 0020-0255. DOI: 10.1016/j.ins.2022.12.083. URL: https://www.sciencedirect.com/science/article/pii/S002002552201578X (visited on 13/02/2023).

[28] Hangyu Zhu and Yaochu Jin. 'Multi-Objective Evolutionary Federated Learning'. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.4 (Apr. 2020). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 1310–1322. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2019.2919699.

[29] Hankz Hankui Zhuo et al. *Federated Deep Reinforcement Learning*. 9th Feb. 2020. DOI: 10.48550/arXiv.1901.08277. arXiv: 1901.08277[cs]. URL: http://arxiv.org/abs/1901.08277 (visited on 28/02/2023).

[30] Eckart Zitzler et al. 'Performance Assessment of Multiobjective Optimizers: An Analysis and Review'. In: *Evolutionary Computation, IEEE Transactions on* 7 (May 2003), pp. 117–132. DOI: 10.1109/TEVC.2003.810758.

# A Complete experimental parameters

All experiments with two or three clients were repeated $10$ times each, with respective seeds $5, 11, 17, 176, 462, 488, 3011, 6543, 9347, 675234$. Experiments with five clients were repeated only 5 times due to the high computing cost; these experiments were run with seeds $5, 17, 488, 3011, 6543$. The number of runs on the non-federated system is matched to the total number of clients involved in all repetitions of the federated system, so e.g. $2 \cdot 10 = 20$ to compare with a federated system with two clients repeated $10$ times. Note that our implementation uses multi-threading to model individual federated participants; therefore the experiments are not deterministic and will not reproduce precisely the same numerical results.

# B Computing details

The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [22] – see https://hpc.uni.lu. The computing time equates to approximately $1450$ hours (i.e., more than 60 days) for a single HPC node. The technical specifications of a cluster compute node are given in Table 4

Table 3: The full set of hyperparameters for all experiments presented in this paper. Left to right: Deep-Sea Treasure (DST), Multi-objective Lunar Lander (MO-LL) and Deterministic Minecart (DMC).

| Parameter name | **DST** | **MO-LL** | **DMC** |
|---|---|---|---|
| **Metaheuristic** (Pareto Simulated Annealing) | | | |
| Iterations | 20 | 25 | 25 |
| Samples per round | 5 | 10 | 10 |
| **Federated Learning** (FedAvg) | | | |
| Total iterations | $10^5$ | $10^5$ | $1.5 \cdot 10^5$ |
| Iterations/local round | $(2/5/10) \cdot 10^3$ | $(2/5/10) \cdot 10^3$ | $(2/5/10) \cdot 10^3$ |
| Number of clients | $2/3/5$ | $2/3/5$ | $2/3/5$ |
| **Reinforcement Learning** (DQN) | | | |
| Train frequency | 16 | 4 | 32 |
| Gradient steps | 8 | $-1$ | 32 |
| Gamma | 0.98 | 0.99 | 0.99 |
| Exploration fraction | 0.2 | 0.12 | 0.8 |
| Exploration final episode | $7 \cdot 10^{-2}$ | 0.1 | $5 \cdot 10^{-2}$ |
| Target update interval | 600 | 250 | 750 |
| Buffer size | $10^4$ | $5 \cdot 10^4$ | $5 \cdot 10^4$ |
| Batch size | 128 | 64 | 64 |
| Learning rate | $4 \cdot 10^{-3}$ | $6.3 \cdot 10^{-4}$ | $2 \cdot 10^{-4}$ |
| Network | $[256, 256]$ | $[256, 256]$ | $[256, 256]$ |
| Reference point | $(0, -50)$ | $(-200, -200, -200, -200)$ | $(-1, -1, -200)$ |

Table 4: Hardware specifications of the cluster nodes employed for experiments.

| | |
|---|---|
| CPU | 2 AMD Epyc ROME 7H12 @ 2.6 GHz [64c/280W] |
| RAM | 256GB |

# C Additional Results

## C.1 Impact of local training phase

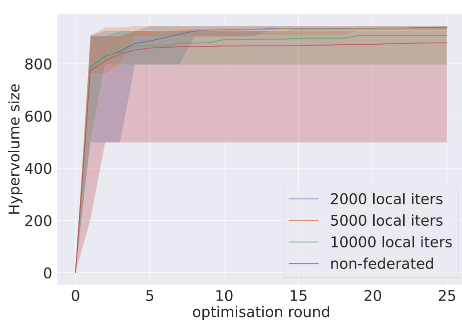
We observe that the duration of the local training phase during federated learning has a notable impact on the overall performance of the MOFL/D algorithm. This matches previous experiences with optimising the performance of federated learning system outside of a higher-level framework. The optimal choice of the federated learning phase differs between the three experimental environments we consider, as is to be expected for problems of differing complexity. For the Lunar Lander environment, the longest tested local training phase (10000 iterations) ultimately produces the most optimal solution set, whereas shorter training phases tend to be more successful in the other two, less complex, environments tested here. An inspection of the solutions obtained e.g. for the Lunar Lander environment clearly shows the impact of local training phase duration on the diversity of the solution set - see the projections of the solution sets shown in Figures 4a, 4b. The diversity of solutions obtained with a shorter local training phase is much lower for this environment, indicating that the federated system likely converges too quickly to a local optimum to adequately explore the solution space.

## C.2 Number of federated clients

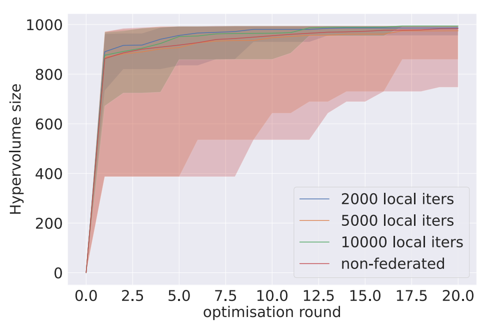
We observe that, in general, an increased number of federated clients leads to an increased performance of the MOFL/D algorithm - see e.g. the hypervolume evolution for the Deterministic Lunar Lander and Minecart, shown in Figures 2a, 5a; compare also Table 2. While this is not the case for the Deep-Sea Treasure environment (see Figure 5b, a higher number of clients in this case still matches

Table 5: Numerical results of experiments on each benchmarking environment. The IGD metric is reported here; for the corresponding hypervolume and sparsity values see Table 2. Each entry reports the mean value of the respective metric, with the variance in parentheses. Lower values of the IGD metric correspond to better performance.

| Conf. | IGD | | |
|---|---|---|---|
| $n_c/n_i^f/ws$ | DST | DMC | MOLL |
| $2/2k/T$ | $0.113(0.1)$ | $0.135(0.1)$ | $24.729(2.0)$ |
| $2/2k/F$ | $0.393(0.5)$ | $0.097(0.1)$ | $24.030(1.7)$ |
| $2/5k/T$ | $0.399(0.5)$ | $0.257(0.2)$ | $25.017(2.1)$ |
| $2/5k/F$ | $0.189(0.2)$ | $0.079(0.0)$ | $24.693(1.6)$ |
| $2/10k/T$ | $0.218(0.2)$ | $0.276(0.2)$ | $23.850(2.4)$ |
| $2/10k/F$ | $0.287(0.4)$ | $0.098(0.1)$ | $24.290(2.5)$ |
| $3/2k/T$ | $0.426(0.5)$ | $0.237(0.2)$ | $24.452(4.0)$ |
| $3/2k/F$ | $0.367(0.5)$ | $0.087(0.0)$ | $23.617(2.1)$ |
| $3/5k/T$ | $0.189(0.2)$ | $0.169(0.1)$ | $22.834(2.0)$ |
| $3/5k/F$ | $0.412(0.5)$ | $0.054(0.0)$ | $23.351(1.8)$ |
| $3/10k/T$ | $0.273(0.3)$ | $0.388(0.1)$ | $23.920(2.5)$ |
| $3/10k/F$ | **0.024** $(0.1)$ | $0.158(0.1)$ | $22.019(1.8)$ |
| $5/2k/T$ | $0.748(1.0)$ | $0.106(0.0)$ | $22.411(2.7)$ |
| $5/2k/F$ | $0.271(0.4)$ | **0.054** $(0.0)$ | $22.138(1.5)$ |
| $5/5k/T$ | $0.366(0.4)$ | $0.157(0.2)$ | $22.462(2.5)$ |
| $5/5k/F$ | $0.151(0.1)$ | $0.088(0.1)$ | $22.780(2.1)$ |
| $5/10k/T$ | $0.308(0.3)$ | $0.168(0.1)$ | $21.549(2.3)$ |
| $5/10k/F$ | $0.082(0.1)$ | $0.086(0.0)$ | **20.221** $(1.5)$ |
| Non-fed. | $0.335(1.1)$ | $0.209(0.2)$ | $27.912(2.0)$ |

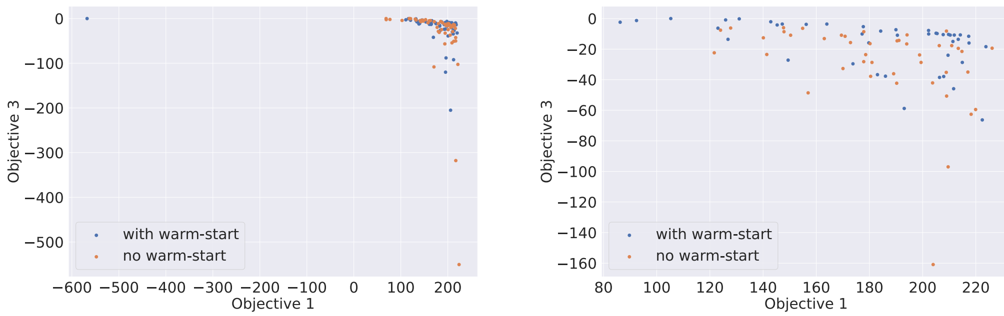



(a) Results for the Deterministic Minecart environment.




(b) Results for the Deep-Sea Treasure environment.

Figure 3: Hypervolume evolution compared for different durations of the local training phase in federated training. Experiments run with 3 federated clients and without pre-trained models.




(a) Results for 2000 iterations per local training round. (b) Results for 10000 iterations per local training round.

Figure 4: Solutions obtained for the Lunar Lander environment compared for different durations of the local training phase in federated training. Experiments run with 3 federated clients and without pre-trained models. (Solution vectors projected into the plane to show objectives 1 and 3. It is clear that the duration of the local training phase has a significant impact on solution diversity.

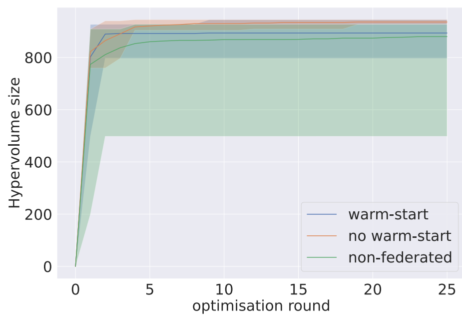

(a) Results for the Deterministic Minecart environment. (b) Results for the Deep-Sea Treasure environment.

Figure 5: Hypervolume evolution compared for variable numbers of federated clients. Experiments run with 2000 local steps per federated round and without pre-trained models.

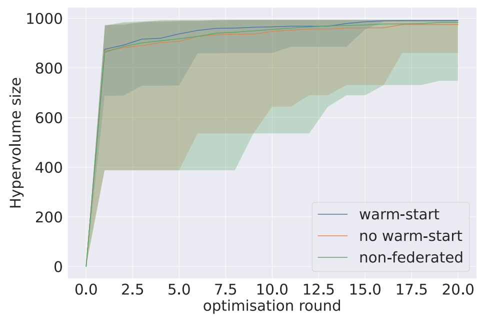
the performance of other systems. The lack of improvement for higher numbers of clients is very likely due to the limited complexity of the problem.

### C.3 Using pre-trained models

The experiments do not offer conclusive results for or against the use of pre-trained models, obtained earlier in the optimisation process, to initialise new federated learning runs.
In some cases, e.g. in the results for the Deep-Sea Treasure environment and the Lunar Lander environment shown in Figure 6b and Figure 7a, respectively, the results of the algorithm run with pre-trained models seem to match or at times during the optimisation process even outperform the algorithm run without pre-trained models. Also notable in some cases, e.g. in the results shown for the DST environment, is the significantly reduced variance of the hypervolume obtained by the system with pre-trained models in the initial stages of convergence, as well as the slightly faster increase of the hypervolume. However, when comparing the corresponding values of the sparsity metric in Table 2, it becomes apparent that these are significantly higher when pre-trained models are used. This indicates that this instantiation of the algorithm tends to find more solutions that are in close proximity to ones already discovered, leading to a high number of solutions, but with low diversity. This observation also serves to explain the reduced performance on the Deterministic Minecart environment, as optimal solutions in this environment are sparse. Therefore, any attempt to exploit the neighbourhood of a previous solution is less likely to be successful.

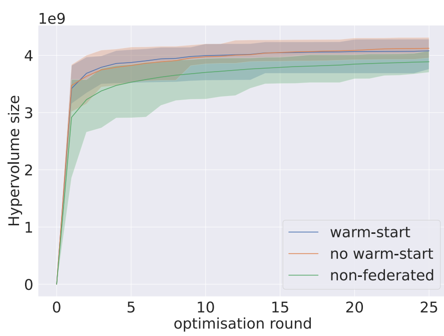



(a) Results for the Deterministic Minecart environment.

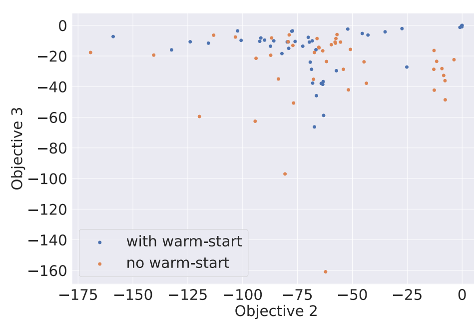



(b) Results for the Deep-Sea Treasure environment.

Figure 6: Hypervolume evolution compared for experiments run with and without pre-trained models. The duration of the local training phase in federation was fixed at $5000$ iterations; the number of federated clients was fixed at $3$.




(a) The evolution of hypervolumes obtained using pre-trained models and those obtained without pre-trained models are quite similar.




(b) Solutions obtained for objectives 2 and 3 (solution vectors projected into the plane). Solutions obtained using pre-trained models are clustered closer together than those obtained without pre-trained models.

Figure 7: Results for experiments run on the Lunar Lander environment with and without pre-trained models. The duration of the local training phase in federation was fixed at $10000$ iterations; the number of federated clients was $3$.