A FRAMEWORK FOR FULLY AUTONOMOUS DESIGN OF MATERIALS VIA MULTIOBJECTIVE OPTIMIZATION AND ACTIVE LEARNING: CHALLENGES AND NEXT STEPS

Anonymous authors
Paper under double-blind review

ABSTRACT

In order to deploy machine learning in a real-world self-driving laboratory where data acquisition is costly and there are multiple competing design criteria, systems need to be able to intelligently sample while balancing performance tradeoffs and constraints. For these reasons, we present an active learning process based on multiobjective blackbox optimization with continuously-updated machine learning models. This workflow is built upon open-source technologies for real-time data streaming and modular multiobjective optimization software development. We demonstrate a proof-of-concept for this workflow through the autonomous operation of a continuous-flow chemistry laboratory, which identifies ideal manufacturing conditions for the electrolyte 2,2,2-trifluoroethyl Methyl carbonate.

1 INTRODUCTION

Data-driven automated laboratories, also called self-driving laboratories, can significantly accelerate molecular synthesis and materials discovery. A key technical challenge of fully autonomous and AI-assisted laboratory design is to effectively collect and utilize data from multiple complex processes in order to inform future experimentation. One long-standing template for utilizing experimental and simulation data is the multi-response surface methodology Myers et al. (2016), whereby initial data sets are gathered through design-of-experiments and then statistical models are built for each quantity of interest, analyzed, and hypothesis-tested iteratively.

In modern scientific and engineering settings, there are several common paradigms that could be considered as specific implementations of this discovery framework, the most common of which are active learning and model-based optimization techniques, such as Bayesian optimization. In order to account for multiple competing criteria, we utilize an active learning framework based in multiobjective optimization, which utilizes surrogates (such as Gaussian processes), optimization solvers, and multicriteria data acquisition in a closed feedback loop.

In this paper, we present our approach to integrate streaming experimental data from a real-world, fully automated continuous-flow chemistry setup into a customizable machine learning (ML) and multiobjective optimization framework for steering the design of battery electrolytes. In the remaining sections, we briefly summarize the problem background, focusing on relevant techniques and challenges. In Section 2 we describe some related work and the overall challenges that we face. Section 3 introduces our framework for addressing these challenges, which is based on integration between the multiobjective optimization library ParMOO Chang & Wild (2023) and the data streaming platform for Manufacturing Data and Machine Learning (MDML) Elias et al. (2020). In Section 4 we present our results from the synthesis of the battery electrolyte 2,2,2-trifluoroetheyl Methyl carbonate (TFMC) with an integrated automated feed, continuous-flow reactor (CFR), and nuclear magnetic resonance (NMR) setup. This experiment does not constitute novel material discovery, but we discuss how our framework will allow for extension to the discovery domain in Section 5.

2 BACKGROUND AND KEY CHALLENGES

There is no shortage of options when it comes to multiobjective optimization solvers and libraries. Several notable techniques include genetic algorithms (GAs) Blank & Deb (2020); search-based methods Le Digabel (2011); and Bayesian optimization Balandat et al. (2020). It is worth noting that we are not the first paper to take a multiobjective approach to chemical synthesis Häse et al. (2018); Shields et al. (2021). However, setting these solvers up to integrate with scientific computing and laboratory environments is a nontrivial task, where each optimization software must be integrated as an optimization service Raghunath et al. (2017) or function Chang et al. (2020) in the broader library. This approach is appropriate for integrating with computer simulation environments, but to successfully integrate with heterogeneous data sources, a more flexible approach is needed Barba-González et al. (2018).

Another challenge of applying optimization solvers for autonomous material discovery is exploiting structures to address problem complexity. Optimization solvers tend to perform well over continuous input spaces, but when some of the design variables are complex (such as chemical networks), the problem becomes combinatorial for classical solvers unless they are carefully tailored to the problem at hand. In the context of material discovery, the molecular structure can be embedded using either its molecular descriptors Shields et al. (2021) or problem-specific latent-space representations Hoffman et al. (2022). Similarly, after performing experiments, the raw outputs of these chemical processes may consist of large volumes of time-series data or spectral measurements, and the majority of this information is lost when the data is post-processed to derive meaningful objectives. However, modern simulation optimization solvers can take advantage of these raw simulation outputs, as a method for exploiting the physical structure in the problem Wild (2017).

Thus, our challenge is to exploit problem-specific structures and embeddings, in a way that generalizes to a variety of applications while modeling data from multiple chemical processes. This requires a multiobjective optimization framework that is flexible enough to utilize customized problem definitions, while still leveraging state-of-the-art modeling techniques, and can be coupled with a flexible data streaming service.

3 A FRAMEWORK FOR AUTONOMOUS EXPERIMENTATION

To address our challenges, we use the multiobjective optimization library ParMOO Chang & Wild (2023) to manage multiple surrogate models, exploit domain knowledge, and utilize problem embeddings through its customizable modular framework. The data streaming service MDML Elias et al. (2020) provides an event-driven architecture with publishers and subscribers to pass data between laboratory workstations, servers, supercomputers, and more. By utilizing MDML as ParMOO's simulation distribution back-end, we are able to stream scientific data from multiple sources, process that data, and issue requests for new experiments in a closed loop. Additionally, both of these tools are open-source, lightweight Python libraries, which can be installed on a typical laboratory workstation. For more information on how ParMOO facilitates exploiting problem structure, see Appendix A.1. For more information on how the MDML handles data streaming from heterogeneous sources, see Appendix A.2.

To integrate ParMOO with the MDML, we have extended ParMOO's `MOOP` class, overwriting the `solve` method to produce Kafka requests for experimental data via the MDML. This means that the MDML platform handles data streaming and recording through its usual methods and is responsible for issuing experiments or simulations, logging data, and storing results.

4 CHEMICAL SYNTHESIS WITH PARMOO AND MDML

In this section, we explore electrolyte production in an autonomously operated CFR using the framework from Section 3. Compared to traditional batch processing, CFRs can provide a means to rapidly test reagent combinations, experiment with high reaction temperatures, and measure yield and specificity when equipped with online characterization. However, manual operation can be very expensive, time-consuming, and inefficient. This makes automating all aspects of the experimentation process attractive to maximize equipment utilization and minimize labor. Furthermore when coupled with ML algorithms, the total number of experiments required to reach optimal parame-

ters is also minimized, in comparison to a traditional exhaustive search and experimental design techniques.

For this example, we are interested in identifying ideal manufacturing conditions for the battery electrolyte TFMC in a CFR, using predetermined reagents and a limited range of values for the manufacturing conditions. Specifically, we must select an optimal flow rate, reaction temperature, and equivalence ratio for the two reagents. To reduce the reaction time and increase production speed, we would like the reaction to be carried out at high temperatures. However, high temperatures can also activate a side reaction, which produces an unwanted byproduct and reduces the purity of the product. Therefore, we are seeking chemical mixtures and conditions that produce large amounts of TFMC but small amounts of the byproduct trifluoroethanol (TFE).

To summarize our self-driving laboratory setup, a MDML client process was created running a modified ParMOO solver, as described in Section 3, which produced requests for experiments and consumed experimental results. Experiment requests were consumed by a laboratory workstation running Labview, which operates a multiport VICI valve controlling the flow of reagents into a Vaportec CFR. Outputs from the CFR were shunted into a Magritek NMR, which recorded peak areas for both the product (TFMC) and byproduct (TFE). This data was recorded in Labview, and returned to ParMOO via a MDML producer. Additional details are given in Section B.

4.1 Experiment: Optimal Manufacturing Conditions

Although it is possible to experiment with different options, we have fixed the solvent and base for this experiment to a predetermined pairing that has shown promising results. Instead, we are focused on identifying optimal manufacturing conditions, such as flow rates, reaction times, and reaction temperatures. In order to ensure safe operations of the CFR, upper and lower bound constraints were given for each of these values, as shown in Table 1.

Parameter	Lower bound	Upper bound
Temperature (degrees C)	35	150
Reaction time (seconds)	60	150
Equivalence ratio (no units)	1	2

Table 1: Design variables and bound constraints for initial experiment.

One of the key challenges for this problem is to achieve convergence on an extremely limited budget. Since we are running real experiments on a CFR, each experiment has significant costs in terms of financial cost, resources, and time. As examples, the raw materials (reagents, solvents, and bases) used in these experiments must have high purity, which comes at a significant financial cost; the machinery (CFR and NMR) have operating costs; and the experiments themselves are time-consuming, requiring up to 10 minutes each. Therefore, we cannot afford to perform hundreds of experiments in this setting, as would be typical in many active learning/Bayesian optimization feedback loops.

ParMOO was instructed to maximize the product and minimize the byproduct subject to the constraints in Table 1. After a 15-point initial design-of-experiments, ParMOO was run for 9 additional iterations with a batch size of three experiments per iteration (i.e., 27 additional iterative experiments driven by the optimization framework). Due to convergence, not all experiments were carried to completion. In total, 41 experiments were completed and the corresponding data is shown in Figure 1. Note that the clustering of late (high index) runs in the bottom-right corner indicates convergence to optimal manufacturing conditions with an extremely limited budget.

5 Path Forward

In Section 4, we were successful in optimizing the manufacturing conditions for a pre-determined material, but additional work is needed in order to broaden our scope to the original goal of material *discovery*. In particular, for this work we did not utilize the most powerful features of ParMOO and the MDML, such as customized latent-space embeddings, exploiting problem structure, and utilization of heterogeneous data sources.

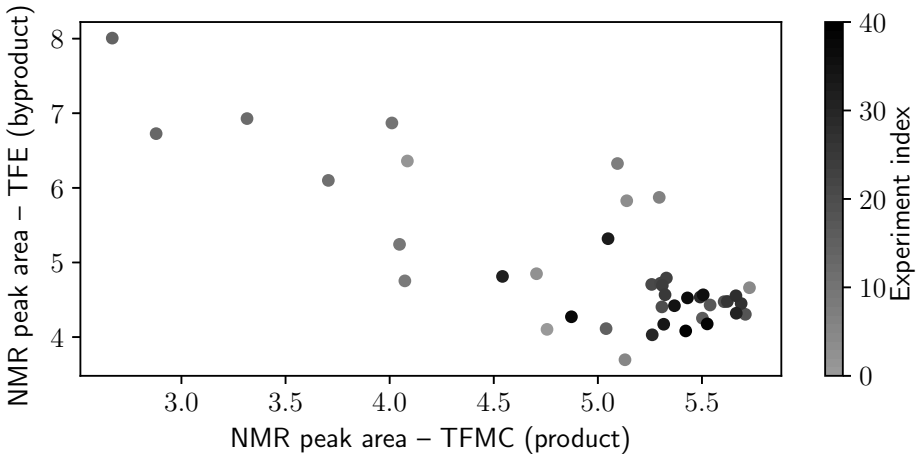

Figure 1: Product (TFMC) vs. byproduct (TFE) areas recorded by NMR for 41 continuous-flow chemistry experiments steered by ParMOO using the MDML. Point colors are coded by experiment index, with larger indices indicating later runs.

In order to extend to discovery, we will search the much larger and complex space of possible reagents through a latent-space embedding, similarly as in Shields et al. (2021). Then, to make optimization more efficient and identify latent structures, we will utilize the full spectrum of NMR data during surrogate-modeling, similarly as in Wild (2017). Additionally, in order to perform material discovery at scale, we will need to utilize heterogeneous data sources including both simulation (for early exploration) followed by experimental data (for late-stage testing), thus leveraging the full capabilities of the MDML Elias et al. (2020).

REFERENCES

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21524–21538. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf`.

Cristóbal Barba-González, José García-Nieto, Antonio J. Nebro, José A. Cordero, Juan J. Durillo, Ismael Navas-Delgado, and José F. Aldana-Montes. jMetalSP: A framework for dynamic multiobjective big data optimization. *Applied Soft Computing*, 69:737–748, 2018. doi: https://doi.org/10.1016/j.asoc.2017.05.004.

Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in Python. *IEEE Access*, 8:89497–89509, 2020. doi: 10.1109/ACCESS.2020.2990567.

Tyler H. Chang and Stefan M. Wild. ParMOO: Python library for parallel multiobjective simulation optimization. Technical Report Version 0.1.0, Argonne National Laboratory, Lemont, IL, USA, 2022. URL `https://parmoo.readthedocs.io/en/latest`.

Tyler H. Chang and Stefan M. Wild. ParMOO: A Python library for parallel multiobjective simulation optimization. *Journal of Open Source Software*, 8(82):4468, 2023. doi: 10.21105/joss.04468.

Tyler H. Chang, Jeffrey Larson, Layne T. Watson, and Thomas C. H. Lux. Managing computationally expensive blackbox multiobjective optimization problems using libEnsemble. In *Proc. 2020 Spring Simulation Conference (SpringSim 2020), the 28th High Performance Computing Symposium (HPC '20)*, pp. Article No. 31. SCS, 2020. doi: 10.22360/SpringSim.2020.HPC.001.

Jakob R. Elias, Ryan Chard, Joseph A. Libera, Ian T. Foster, and Santanu Chaudhuri. The manufacturing data and machine learning platform: Enabling real-time monitoring and control of scientific experiments via IoT. *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pp. 1–2, 2020. doi: 10.1109/WF-IoT48130.2020.9221078.

Florian Häse, Loïc M Roch, and Alán Aspuru-Guzik. Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories. *Chemical science*, 9(39):7642–7655, 2018. doi: 10.1039/C8SC02239A.

Samuel C. Hoffman, Vijil Chenthamarakshan, Kahini Wadhawan, Pin-Yu Chen, and Payel Das. Optimizing molecules using efficient queries from property evaluations. *Nature Machine Intelligence*, 4(1):21–31, 2022. doi: 10.1038/s42256-021-00422-y.

Sébastien Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):Article No. 44, 2011. doi: 10.1145/1916461.1916468.

Raymond H. Myers, Douglas C. Montgomery, and Christine M. Anderson-Cook. *Response Surface Methodology: Process and Design Optimization Using Designed Experiments*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 4 edition, 2016. ISBN 9781118916032.

Chaitra Raghunath, Tyler H. Chang, Layne T. Watson, Mohamad Jrad, Rakesh K. Kapania, and Raymond M. Kolonay. Global deterministic and stochastic optimization in a service oriented architecture. In *Proc. 2017 Spring Simulation Conference (SpringSim 2017), the 25th High Performance Computing Symposium (HPC '17)*, pp. Article No. 7, Virginia Beach, VA, USA, 2017. SCS. doi: 10.22360/springsim.2017.hpc.023.

Benjamin J. Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I. M. Alvarado, Jacob M. Janey, Rryan P. Adams, and Abigail G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021. doi: 10.1038/s41586-021-03213-y.

Stefan M. Wild. Solving derivative-free nonlinear least squares problems with POUNDERS. In Tamas Terlaky, Miguel F. Anjos, and Shabbir Ahmed (eds.), *Advances and Trends in Optimization with Engineering Applications*, pp. 529–540. SIAM, 2017. doi: 10.1137/1.9781611974683.ch40.

A Software Details

A.1 ParMOO

ParMOO is a framework for customizing and deploying multiobjective simulation optimization solvers Chang & Wild (2023). The key distinction in ParMOO is the difference between a *simulation* and an *objective*, as depicted in Figure A.1. This abstraction and the utilization of custom latent-space embedding tools are the primary mechanisms by which ParMOO exploits problem structure.

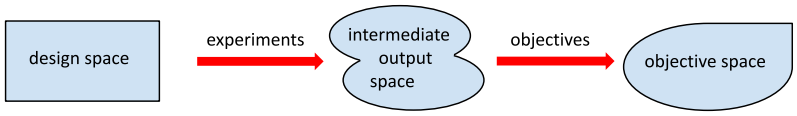

Figure 2: ParMOO models complex processes separately from objectives, in order to exploit physical structure in how the objectives are defined.

To create a ParMOO instance, users must first specify all design variables, data sources (e.g., experiments and simulations), problem constraints, and problem objectives. For complex design variable types, ParMOO is capable of automatically generating latent-space embeddings or utilizing customized embedding tools. Next, users provide

- any pre-existing data or a technique for generating an initial design-of-experiment;
- surrogate models for learning each of the simulation outputs;

- optimization solvers for the surrogate problem; and
- one or more data acquisition functions for determining how ParMOO will blend objectives.

When run, ParMOO will evaluate the initial experimental design, then iteratively generate batches of new experiments to be distributed for evaluation across all data sources, by following the standard response surface methodology. For further information on this process, see ParMOO's online documentation Chang & Wild (2022).

A.2 THE MDML

The Manufacturing Data and Machine Learning (MDML) platform provides cyberinfrastructure to standardize the research and operational environment to act on scientific data streams and integrate ML in the loop to steer or optimize experiments. The MDML has been designed to meet the needs of in-situ measurements for accelerating scalable materials manufacturing while providing capabilities that can be easily adapted to any scientific domain. The MDML enables users to construct rich, data-oriented analysis pipelines that span disparate computational environments.

Underlying the MDML is the widely-used, distributed event streaming platform named Apache Kafka. This allows scientists using the MDML to follow an event-driven architecture. Clients connected to the MDML's Kafka service are able to publish and subscribe to one or many data channels that they have created. Clients here include researchers themselves, software programs controlling/monitoring experiments, ML models, and more. Given this architecture, any relevant experiment data can be streamed anywhere and fed into client processes running ML models (such as ParMOO) in a unified format. Additionally, when the experimental data collection process can be fully automated (as with our integrated feed, CFR, and NMR setup), this enables a user-free feedback loop.

B EXPERIMENTAL METHODS

To perform autonomous experimentation on the CFR using ParMOO and the MDML in a closed feedback loop, a MDML client process was created running a modified ParMOO solver, as described in Section 3. Then, an additional experiment-running client was created on a laboratory workstation that directly controlled the CFR. Finally, the MDML host was run on a remote laboratory mainframe to broker requests and collect data.

From the ParMOO client side, a ParMOO solver was created and configured to use a 15-point Latin hypercube design-of-experiments, Gaussian process surrogate model, generalized pattern search (GPS) optimizer, and three acquisition functions (two using the epsilon-constraint method for Pareto front exploration, and the third using a fixed weighting of the two objectives). Following the active learning framework, the ParMOO client generated an initial design-of-experiments then posted Kafka producer requests for these designs, listened for results data through a Kafka consumer, used those results to update its Gaussian models, optimized those models to target new experimental designs, then posted new producer requests in an iterative feedback loop. In order to reduce reaction times on the CFR, experiment requests within a single batch were sorted by the reaction temperature. This was beneficial since it takes additional time to stabilize temperatures when adjusting reaction temperatures.

For the experiment client, a lab computer running the Labview control program (LVCP) was configured to consume Kafka requests from the ParMOO client. The LVCP consumer polled for a producer request in the form of a valid experiment parameter test set. Whenever a valid experiment parameter set was received from the ParMOO client, an experiment was set in motion by automated deployment of the provided set on the CFR and NMR setup, described in the next paragraph. After completion of a test, the LVCP posted the result as a Kafka producer and awaited the next data point as a consumer.

To perform the physical experiments, a CFR apparatus mfd by Vaportec was used to carry out continuous-flow reactions. Inputs to the A and B channels of the Vaportec unit were provided by multiport VICI valves connecting up to 15 possible reagents to the CFR. The output from the CFR was shunted through a Magritek NMR equipped with a flow-through cell. The selector valves, Vaportec reactor and Magritek NMR were fully controlled using a Labview control program. To

calibrate the equipment, the reactor was manually set up and NMR spectra tests were run on pure solvents. Operating parameter ranges were also set. Whenever a valid experiment parameter set was received, the experiment was initiated from LVCP. Steady state was determined by monitoring the integrated NMR peak areas provided by the Magritek software, and test completion was determined by the integrated peak area falling below a standard deviation criteria set in advance. To conclude a single experiment, data acquisition by the NMR required stopping of flow though its flow-through cell, whereas it was not possible to stop the flow in the CFR. Therefore a bypass loop was employed on the CFR in order to terminate each experiment.

C OPEN ACCESS STATEMENT

We acknowledge that it will not be possible for researchers without access to highly specialized laboratory equipment to reproduce our results. However, to facilitate open access and reproducibility of results as much as possible, the source code for using ParMOO to create MDML consumer/producer services and a variation of our script used for Section 4.1 (demonstrating solver settings, but not running the experiments) is publicly available at `https://anonymous.4open.science/r/cfr-materials-F232` [1]. We have also included a copy of the raw MDML-recorded outputs (a JSON file) from our experiment, as well as a Python script for post-processing these results to produce the plot in Figure 1.

[1] This anonymous repository will be replaced with a public GitHub repo, if accepted