Reverse engineering of vinyl acetate polymerizations by genetic algorithm-based multi-objective optimization

Jelena Fiosina[a;*], Philipp Sievers[b], Marco Drache[b] and Sabine Beuermann[b;]**

[a]Institute of Informatics, Clausthal Technical University, Julius-Albert-Str. 4, Clausthal-Zellerfeld, Germany
[b]Institute of Technical Chemistry, Clausthal Technical University, Arnold-Sommerfeld-Str. 4, Clausthal-Zellerfeld, Germany
ORCiD ID: Jelena Fiosina https://orcid.org/0000-0002-4438-7580,
Philipp Sievers https://orcid.org/0000-0003-1143-4413, Marco Drache https://orcid.org/0000-0001-8841-8812,
Sabine Beuermann https://orcid.org/0000-0003-4903-5717

Abstract. This work proposes a multi-objective optimization (MOO) approach for reverse engineering of vinyl acetate polymerization processes. Our method leverages machine learning (ML) models trained on data from kinetic Monte Carlo (kMC) simulations to replace expensive laboratory experiments. We employ a genetic algorithm (GA) as the MOO optimizer, considering reaction time, monomer conversion, and molar mass distribution (MMD) similarity as objectives. The trained ML models assist the optimization process and predict key polymer properties for candidate recipes generated by the GA, enabling rapid fitness function evaluation.

The proposed framework involves: (1) training ML models for monomer concentration and MMD prediction using kMC simulation data; (2) performing GA-based MOO to identify optimal recipes (Pareto front) for a target MMD (3) selecting the most suitable recipe based on user priorities from the resulting Pareto front, considering user-defined weights for each objective (reaction time, conversion, MMD).

Our experiments demonstrate that the GA, coupled with simulation-supported ML, efficiently identifies optimal recipes with high accuracy. Notably, the ML models achieve good performance even with limited training data. This approach offers a rapid and cost-effective solution for reverse engineering of vinyl acetate polymerization processes.

1 Introduction

Polymerization processes play a crucial role in material science, but their intricate mechanisms and the resulting statistical variations within polymers pose significant challenges for design and prediction. These variations significantly impact the final product's properties, making customized polymer development a resource-intensive endeavor.

Traditional laboratory testing, while valuable, can be time-consuming and expensive. Thankfully, computational modeling offers powerful alternative techniques like kinetic Monte-Carlo (kMC) simulations [10], differential equations [28], and machine learning (ML) models [14] have emerged as valuable tools for modeling polymer behavior.

* Corresponding Author. Email: jelena.fiosina@tu-clausthal.de
** Corresponding Author. Email: sabine.beuermann@tu-clausthal.de

The ability to predict polymer characteristics opens doors to tackling complex problems like polymerization reverse engineering (PRE) (Figure 1).

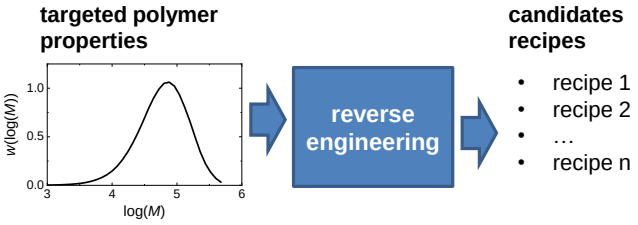

Figure 1. Polymerization reverse engineering

PRE aims to identify a polymerization process that yields a polymer with desired properties. Traditionally, the "trial and error" approach is impractical due to the sheer number of potential processes. kMC simulations, while valuable, cannot perform simulations backward and require extensive experimentation.

This work proposes a novel approach that formulates PRE as a ML-assisted multi-objective optimization (MOO) problem [15]. We couple MOO with data-driven machine learning (ML) methods trained on a limited amount of data from kMC simulations to accelerate the optimization process.

ML methods have gained widespread popularity in recent years, demonstrating effectiveness across diverse domains [34]. However, applying ML to polymerization processes has been limited due to the typically small data sets available compared to other fields with readily available large data sets. To address this challenge, we leverage powerful ML techniques like random forest and kernel density regression, particularly well-suited for tasks with limited data [13]; [14]; [23].

MOO [33] is a powerful optimization technique for maximizing or minimizing multiple objectives simultaneously while considering constraints. It has applications in various scientific fields, including engineering [19], economics [9], and logistics [20], where optimal decision-making requires balancing trade-offs between conflicting

objectives. Among various optimization methods, we employ a genetic algorithm (GA) for its effectiveness in both single-objective and MOO problems [22].

In this study, we leverage MOO to solve the PRE problem. Our objective is to find a "recipe" (polymerization process) that produces a polymer with a target molar mass distribution (MMD). We consider three optimization objectives: maximizing the similarity between the predicted MMD and the target MMD, minimizing reaction time, and maximizing monomer and initiator conversion.

Our proposed approach, combining ML and MOO, offers several advantages. First, it provides multiple potential polymerization procedures to achieve the desired polymer properties. Second, ML models trained on a limited amount of kMC simulation data enable efficient prediction and optimization.

The proposed framework consists of three key steps:

Train ML models for predicting monomer concentration and MMD using kMC simulation data [14]. Perform GA-based MOO to identify optimal recipes (Pareto front) for a target MMD. Select the most suitable recipe from the Pareto front based on user priorities and user-defined weights for each objective (reaction time, conversion, MMD). We demonstrate the effectiveness of our method through its application to the vinyl acetate (VAc) polymerization system.

2 Related work

Early in the new millennium, a few supervised learning data-driven techniques to polymerization process modeling (PPM) begin to emerge [6]; [11]; [39].

Subsequently, an ML model based on deep learning and an offline kMC simulator was developed by Mohammadi et al [30]. It was proposed to obtain more training data combining kMC simulation with ML models. Other more accurate predictions for PPM were obtained with deep learning models in [26]. Ensemble-learning methods (including random forest and XGBoost) were evaluated for prediction of polymer and process properties such as conversion and MMD [7]; [8]; [17]; [26]; [38], [40].

In [15] we used ML methods to train models with data from the kMC simulator. However, here the ML approach is universal, which allows to model the polymerization processes not only based on the simulated results, but also on laboratory experiments. We successfully predicted polymer and process properties such as monomer concentration, molar mass averages, reaction time, MMD, using ML decision tree ensemble methods such as random forest, XGBoost, and CatBoost. These results can be successfully used in this study for the PRE problem.

One of the major challenges in constructing PRE models is the existence of multiple valid solutions for a given set of desired outputs. Unlike traditional models with a one-to-one link between polymerization variables and microstructural properties, PRE models can yield various possibilities due to the dependence of polymer and process properties on the specific reaction pathway taken, even with the same input variables [30]. Traditional PPMs excel at predicting properties from input variables. However, identifying the ideal input conditions to achieve specific pre-defined outputs (e.g., conversion, yield) is more complex and requires optimization techniques. For systems with intricate reaction mechanisms, intelligently exploring the vast search space of possible reaction conditions becomes crucial [29]. Evolutionary algorithms like GAs have proven successful for optimization in various domains [1], [30]. Machine Learning (ML) based optimization techniques have also been explored for PRE models [16].

Our approach ensures continuous improvement by employing ML models for real-time result evaluation during optimization. This offers a significant speed advantage compared to traditional methods that rely on computationally expensive Kinetic Monte Carlo (kMC) simulations. Additionally, our ML-based approach has the potential to handle more complex output structures. While Mohammadi et al. [30] proposed a GA-based optimizer that generates random recipes and feeds them to a kMC simulator, our approach leverages the speed and flexibility of ML for faster and potentially more adaptable optimization in PRE.

We already used a random forest ML method formulated as a multivariate and multi-target regression problem [37] for PRE [14]. The model took a targeted MMD and produced multiple responses as parts of the required procedure, giving the initial conditions to produce the polymer with the targeted properties. The optimization focused solely on minimizing variations in polymerization procedure parameters.

MOO has become increasingly popular in chemical engineering. Fiandaca and Fraga [18] employed multi-objective GAs to optimize a pressure swing adsorption process, maximizing both nitrogen recovery and purity. This demonstrates MOO's ability to handle problems with conflicting objectives. Similarly, Ganesan et al. [16] used MOO for the combined reforming and partial-oxidation of methane, optimizing methane conversion, carbon monoxide selectivity, and the hydrogen-to-carbon monoxide ratio. MOO's applications extend to diverse areas like chemical extraction [2] and bioethanol production [32]. Even in advanced material design, Kim et al. [24] applied GA-based MOO to create polymers with both high bandgap and high glass transition temperature. These examples showcase the growing importance and versatility of MOO in chemical engineering.

In our previous paper, we solved PRE problem for the VAc polymerization system, using a kMC simulated search space [15]. PRE was formulated as a MOO problem and we compared direct and MMD clustering based optimization approaches for finding the most relevant recipes for a target MMD. The importance of each objective was estimated using weights and the problem was translated to single objective optimization. Our approach allowed to find multiple recipe candidates with given targeted properties and objective weights.

3 Proposed approach

This section details our proposed approach for solving the PRE problem: an MOO framework that leverages both ML and GAs.

The approach takes two key inputs:

- Target MMD defines the desired properties of the final polymer.
- Data set for training ML models is generated by a kMC simulator and used in the optimization process.

Then, the MOO framework processes these inputs and generates the desired output as multiple optimal candidate recipes. These recipes represent potential polymerization processes that can achieve the target MMD in the most efficient way possible. "Efficiency" in this context can be considered as a combination of factors depending on user priorities, such as minimizing reaction time, maximizing monomer conversion, or achieving the closest possible match to the target MMD.

3.1 PRE problem formulation as MOO

This subsection details the formulation of the MOO problem for PRE, based on the work presented in [15]. Here, we aim to identify a

polymerization recipe that optimizes multiple objectives simultaneously.

The recipe for the polymerization process is defined as a vector of polymerization variables $\mathbf{r} = [c_{m,0}, c_{ini,0}, t]$, where $c_{m,0}$ is the initial concentration of monomer, $c_{ini,0}$ is the initial concentration of initiator, and t is reaction time.

The lower and upper bounds for each variable are established based on the data obtained from the kMC simulations. For this study, in contrast to [15], the simulated values of $c_m(\mathbf{r})$ and $MMD(\mathbf{r})$ were predicted with the corresponding ML models [14].

The MOO framework considers three primary objectives:

1. Minimizing mean squared error (MSE): This objective aims to achieve the closest possible match between the predicted and target MMD. The ML models trained on the kMC simulation data are used for this prediction (as described in [14]):

$$\min_{\mathbf{r}} f_{MSE}(\mathbf{r}) = \min_{\mathbf{r}} MSE(MMD^{target}, MMD(\mathbf{r})), \quad (1)$$

   where $MMD(\mathbf{r})$ is predicted by the corresponding ML model;
2. minimizing relative monomer concentration: This objective prioritizes minimizing the concentration of monomer:

$$\min_{\mathbf{r}} f_{c_m}(\mathbf{r}) = \min_{\mathbf{r}} \frac{c_m(\mathbf{r})}{c_{m,0}}, \quad (2)$$

   where $c_m(\mathbf{r})$ is predicted with the corresponding ML model. Alternatively, the monomer conversion function can be expressed as $f_{conv}(\mathbf{r}) = 1 - f_{c_m}(\mathbf{r})$ and then maximized.
3. minimizing reaction time: This objective focuses on achieving the desired polymer properties in the shortest possible reaction time:

$$\min_{\mathbf{r}} f_t(\mathbf{r}) = \min_{\mathbf{r}} t, \quad (3)$$

   where t is directly obtained from $\mathbf{r}$.

It is important to note that the number of objectives can be extended to incorporate additional user-defined priorities.

To facilitate the optimization process, we utilize the weighted sum method [27]; [25] to convert the final Pareto front of the MOO problem into a single-objective result. This method assigns specific weights (w_i) to each normalized objective function (f_i). The MOO function is represented as a single-objective one:

$$\min_{\mathbf{r}} f = \min_{\mathbf{r}} \sum_i w_i f_i(\mathbf{r}), \quad (4)$$

$$\sum_i w_i = 1, i \in \{MSE, c_m, t\}.$$

Function f is calculated for each candidate recipe. The recipe with the smallest value of function f is then selected as the optimal solution based on the user-defined weights assigned to each objective.

3.2 *Coupling MOO and ML*

Figures 2 illustrates the key steps of our proposed algorithm and Figure 3 the interaction between ML models and genetic algorithms.

1. **ML model training**: We begin by constructing and training ML models using data generated by the kMC simulator. These models enable the prediction of crucial polymerization properties like MMD and monomer concentration, essential for the subsequent optimization stage.

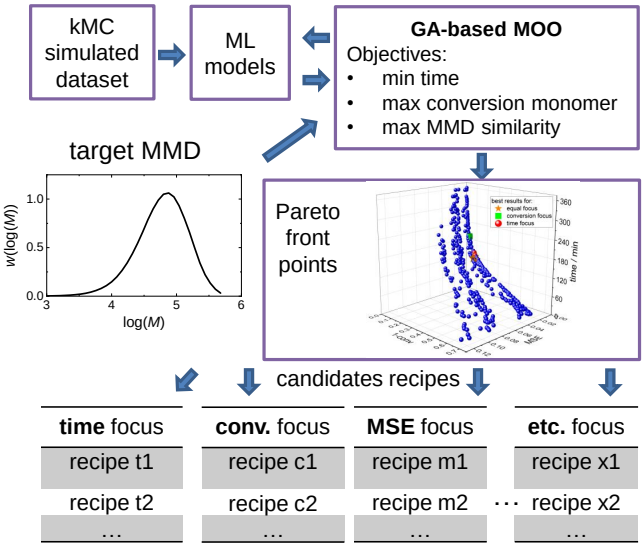

Figure 2. ML assisted GA-based MOO approach for reverse engineering

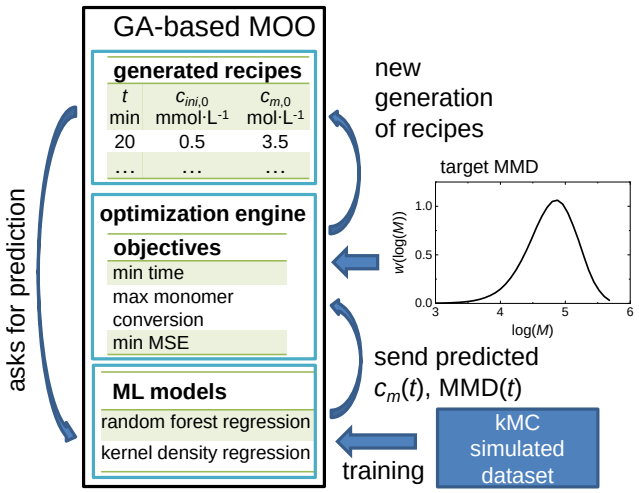

Figure 3. Collaboration background of GA-based MOO and ML methods

2. **Optimization process**: First, the user defines a target MMD representing the desired polymer properties. Then, the algorithm seeks to identify the optimal recipe that achieves this target. This optimization is an iterative process. During each iteration each objective is calculated for candidate recipes generated by an GA according equations (1-3). The GA dynamically communicated with the ML models, allowing for the prediction of polymer properties (MMD and monomer concentration) for each candidate recipe. By calculating these objectives (considered as "fitness functions" in the context of the GA), we establish a Pareto optimal search space for the MOO process. This front represents a set of optimal

solutions that offer trade-offs between the defined objectives.The GA iterates through a series of steps, progressively refining the candidate recipes. At the next step, the GAs methods reproduction/crossover, and mutation mechanisms are utilized, which results in the new population of recipes.
3. **Identifying optimal recipes:** Once the optimal solutions remain stable (no further improvement), the algorithm provides the user with a set of optimal candidate recipes based on user-defined objective weights according to equation 4.

3.3 Data acquisition

Here we describe the process of generating and preparing the data used to train the ML models. An in-house developed kMC simulator, mcPolymer, was utilized to generate the training data. This simulator employs a comprehensive kinetic model for vinyl acetate radical polymerization, encompassing all essential elemental reactions [12]. A grid search approach was employed to explore a meaningful range of chemical reaction conditions, including vinyl acetate and initiator concentration.

The simulated data serves as the foundation for training the ML models used within our optimization framework. These models predict crucial polymer properties like MMD and monomer concentration for candidate recipes generated by the genetic algorithm during the optimization stage.

3.4 ML models

This section discusses the selection and optimization of ML models used for property prediction within our framework. We build upon the initial exploration presented in [14], where various methods were compared for predicting diverse polymer and process properties. Our framework utilizes two primary ML models for prediction both monomer concentration and MMD:

1. **Random forest (RF) regression [3]**: This model offers a good balance between performance and computational efficiency. While other ensemble methods like XGBoost and CatBoost exhibited similar performance, the random forest's faster training time made it the preferred choice for our application. RF regression supports multiple outputs, required for the multi-target regression used for MMD prediction.
2. **Kernel density (KD) regression [21]**: This non-parametric method demonstrated superior performance compared to tree-based ensemble methods, particularly for data points outside the training grid. KD regression is known for its versatility across various tasks and domains [13]. Although its training can be time-consuming, especially for multivariate data sets, the relatively small size of our data sets (3 features and 225 experiments) made it computationally feasible in this case. KD regression does not support multiple outputs, thus an ensemble of single-target KD regressions was applied.

The proposed ML approach had the following challenges:

- selection of optimal parameters for KD regression: Due to the high dimensionality of the MMD output (100 intervals), standard cross-validation techniques were not applicable for finding optimal bandwidth parameters for the KD regression model. To address this, we employed Scott's rule [36] to obtain approximate values.
- generalization through time integration: Compared to the models discussed in [14], we introduce a key modification to enhance generalization. Time is included as an input variable for the monomer concentration prediction model. This eliminates the need for multi-target regression for prediction of monomer conversion and allows for flexibility in predicting values across a broader range of input parameters, including time.

The proposed approach provides the following benefits for the application of ML to generate the search space:

- reduced gap size: Leveraging ML predictions the search space is refined by filling the gaps between kMC simulation data points, which leads to a more precise optimization solution.
- computational efficiency: ML model predictions are significantly faster (seconds) compared to kMC simulations, which can require hours of computation. This efficiency is a key advantage for the overall optimization process.

4 Experimental results

4.1 Generation of training data and experimental design of ML models

The kMC simulations investigated radical polymerizations of vinyl acetate (VAc) as monomer, tert-butyl peroxypivalate as initiator, and methanol as solvent. A comprehensive kinetic model for VAc radical polymerization, encompassing all elemental reactions, was used for the simulations. This validated model has been shown to accurately describe a large set of experimental data [12]. The simulations were conducted under the following conditions: constant temperature of 60 °C, $c_{ini,0}$ ranging from 1.0 to 20.0 mmol·L^{-1}, and $c_{m,0}$ ranging from 2.0 to 5.0 mol·L^{-1}. A total of 200 simulations were performed using uniformly and randomly chosen parameters within these concentration ranges. This random selection strategy, along with a planned random selection of test data, ensures a well-balanced distribution for training and testing data sets.

The simulations captured property data every 20 minutes for 6 hours, resulting in a comprehensive data set (3600 data points) that covers various reaction conditions. Thus, the data set contained 3600 different MMDs, which were selected in a way that the relevant technical reaction conditions are sufficiently covered. The number of data is reasonable in view of the simulation time. The simulations took 9 h with 128 CPU cores (2 AMD EPYC 7H12) and 2 TB of RAM. Moreover, this number of 3600 MMDs allowed for the construction of ML models for reverse engineering and MMD prediction with good performance [14]. The data set is available as Supplementary Material.

This data was used to train ML models for polymer reaction engineering. The data was first split into training (80%) and testing (20%) sets. R-squared metric and cross-validation were used to evaluate the performance of the trained models. Grid search was employed to optimize the models' hyperparameters.

Two types of ML models were used: RF for predicting both monomer concentration c_m and MMD, and KD regression models for both c_m and MMD prediction. For the RF model of c_m, the number of trees in the forest is equal to 100 and the maximum depth of the tree is equal to 30. For the RF model of the MMD, the values are 50 and 20, respectively.

The values for the bandwidth or smoothing parameters for multivariable KD regression are selected for each variable and for both c_m and MMD models equal to [0.195, 0.0012, 1310.56] for $c_{m,0}$ in mmol·L^{-1}, $c_{ini,0}$ in mol·L^{-1} and time t in seconds, respectively.

Smoothing parameters scale the width of the kernel. This means placing a smooth function at the location of each data point and then summing up the results. We use Gaussian kernels, and the initial values for smoothing parameters are selected according to the Scott's rule [35].

We considered the explainability of the ML methods. For RF we applied a model-specific built-in feature explainability method, which is based on "mean decrease in impurity" [4]. For KD regression, as it is a black-box ML method, there is no build-in explainability functions, and therefore, a model-agnostic explainability method "permutation importance" [31] was applied. The method was selected, because of its simplicity and previous experience [14], where the model explainability results were similar compared to "Shapley values" [5], another popular explainability method.

4.2 Prediction of monomer concentration and MMD

The analysis focused on three key questions for each property prediction using machine learning models:

- **Best Model Selection:** Which ML model provides the most accurate predictions for a specific property?
- **Impact of Training Data:** How does the model's accuracy decrease as the training data size shrinks? How much data is needed for reliable predictions?
- **Explainability and Chemistry:** How do polymerization parameters influence the model's predictions? Are the explanations from the chosen methods chemically sound?

Figure 4 shows the predictions for the variation of c_m with time (A-C) and MMD (D-E) obtained with the RF and KD ML models. Figure 4A,D illustrate a typical prediction for one specific experiment. The simulated (ground truth) black line trajectory is compared with predicted trajectories of the ML models. For this particular example, the models provide good predictions for both properties. KD regression performs best with an almost perfect overlap of the predicted and the kMC simulation-derived c_m and MMD. This finding is also generalized for the full test set by the performance metric R^2, whose results are shown in Figure 4B,E. Thus, for the 100 % training set and c_m, both KD and RF regression are associated with the best R^2 score of 0.995. However, for the MMD prediction and for 100 % training set size, KD regression outperforms the RF model, which is demonstrated with R^2 values of 0.996 and 0.964, respectively. The average performance of each model with different sizes of the training set is evaluated. As expected, the performance reduces with decreasing size of the training set as shown in Figure 4B,E. Remarkably, even with only 10 % of the training data the KD model is associated with a value of R^2 = 0.965 for c_m prediction and of R^2 = 0.954 for MMD prediction. Thus, the number of training data can be significantly reduced and ML models can be trained with only 10 % of training data as an optimal trade-off between the performance and scalability. Investigations into the explainability of the considered models are given in Figure 4C,F. It is seen that for c_m prediction, time t is the most decisive input parameter, accounting for around 80 % of the result. The importance of c_{ini} is lower with a contribution of around 16 % followed by $c_{m,0}$ with a contribution of around 4 %. In contrast to this, for MMD prediction, $c_{m,0}$ is the most decisive input parameter for each model, accounting for around 74 % and 42 % of the RF and KD result, respectively. The importance of time is lower with a contribution of around 23 % and 37 % for RF and KD regression, respectively followed by $c_{ini,0}$ with a contribution of around 3 % and 20 % for RF and KD models, respectively. The explainability results for MMD differs for RF and KD models, however the order of the feature importance remains the same. The differing importances for the prediction of c_m and MMDs is chemically reasonable: The monomer concentration decreases throughout the reaction with time, thus time is the most decisive factor. On the contrary, the MMD is strongly dependent on $c_{m,0}$ and does not change to large degree with time during radical polymerizations.

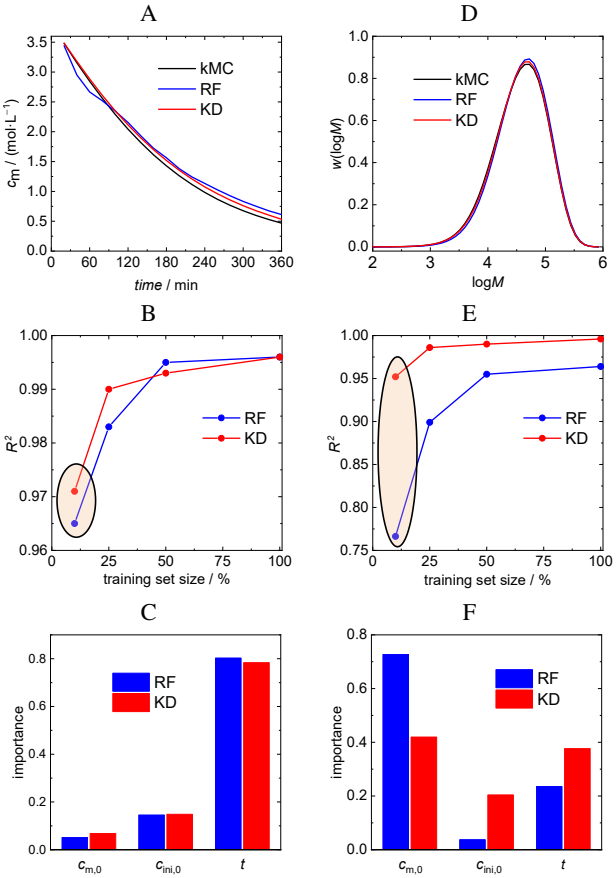

Figure 4. RF and KD prediction models for c_m (A-C) and MMD (D-E): example predictions for c_m (A) and MMD (D), where $c_{m,0}$=3.832 mol $\cdot L^{-1}$; $c_{ini,0}$=1.824 mmol$\cdot L^{-1}$; model performance for a reduced training set sizes for c_m (B) and MMD (E), where original data set size: 200 (simulations)$\cdot$ 18(time points)=3600; feature importance of the RF model for c_m(C) and for MMD (F)

4.3 Optimization

We employed ML models constructed in Section 4.2 within a Genetic Algorithm Multi-Objective Optimization (GA MOO) framework to identify candidate recipes for achieving a target MMD. The performance of our GA MOO approach was compared to a direct optimization approach, utilizing only simulated data as in [15].

A grid search was conducted to identify optimal hyperparameters for the GA, including population size, number of generations, as well as crossover and mutation rates (Table 2). Convergence analysis (Figure 5A) indicated that 15 generations were sufficient training time with our chosen hyperparameters, as evidenced by the stabilization of the number of Pareto points.

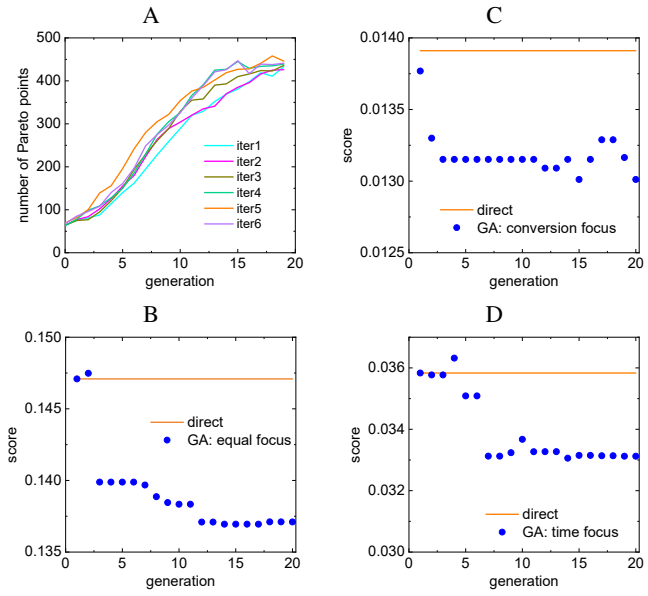

Figure 5. GA learning from different perspectives, (A): dependency of the number of Pareto front points from the number of generations; weighted scores dependency from the number of generations for conversion focus (B), equal focus (C) and time focus (D), for a specific target MMD: $c_{m,0}$ = 2.995 mol ·L^{-1}; $c_{ini,0}$ = 17.84 mmol·L^{-1}, t = 260 min , 10 % training data

Three distinct optimization focuses were explored, reflected by varying weights assigned to the objective functions (Equation 4, Table 1). A high weight (90 %) was maintained for MSE in case of conversion and time focuses due to the inherent importance of achieving the target MMD shape. Figures 5B-D illustrate the consistent improvement/decreasing of scores for each focus with increasing generations within the GA MOO approach (blue points), outperforming the direct solution (orange line) across all focuses.

Table 1. User defined objective focuses

description	w_{MSE}	w_{c_m}	w_t
equal focus	1/3	1/3	1/3
conversion focus	0.9	0.08	0.02
time focus	0.9	0.02	0.08

Table 2. GA hyperparameters

parameter	value
population size	400
crossover rate	0.2
mutation rate	0.2
local search rate	0.3
coordinate displacement during local search	[0.02 0.004, 1000]
number of iterations	20

Figure 6A depicts the Pareto front for a specific target MMD, showcasing the trade-offs between objectives based on user-defined weights (colors). This allows researchers to select the optimal solution from the available options within the Pareto space. Figure 6B displays the significantly smaller Pareto front obtained from the direct approach for the same target MMD. Both figures highlight optimal solutions for each focus.

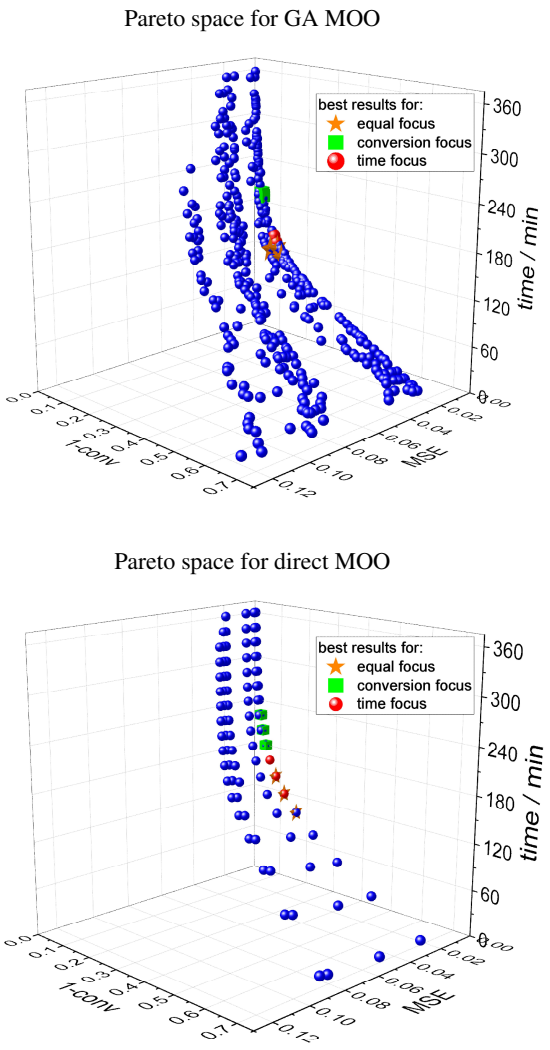

Figure 6. Pareto space for GA and direct MOO with different highlighted focuses, for a specific target MMD: $c_{m,0}$ = 2.995 mol ·L^{-1}; $c_{ini,0}$ = 17.84 mmol·L^{-1}, t = 260 min , 10 % training data

Figure 7 presents the best candidate recipes identified by the GA MOO approach (blue points). Only solutions exceeding the performance of the best direct solution are included. The original recipe of the target MMD (green point) and the best direct solution (red point) are also shown for reference. Notably, the GA MOO approach offers a wider range of recipes to achieve the target MMD. Furthermore, several predicted recipes exhibit shorter reaction times compared to the recipe of the target MMD.

Figure 8 compares the MMDs of candidate solutions with the target MMD. Due to the multi-objective nature (where MSE is not the sole focus), perfect similarity is not always achieved, particularly in the equal focus case, where MSE, conversion, and time have equal weight. However, for other focuses with a significant MSE contribution, the predicted MMDs closely resemble the original MMD.

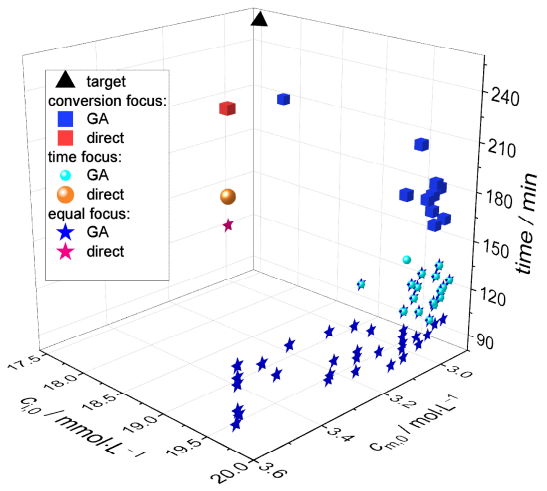

Figure 7. Recipe candidates for each focus, for a specific target MMD: $c_{m,0}$ = 2.995 mol ·L^{-1}; $c_{ini,0}$ = 17.84 mmol·L^{-1}, t = 260 min , 10 % training data

On average, the optimization procedure identified approximately 18 candidate recipes for each MMD within the test set, across all focus configurations. The GA MOO approach outperforms the direct approach in roughly 80 % of cases, providing a superior set of candidate solutions. However, in the remaining 20 % of cases, the direct approach might yield a better solution. Interestingly, combining both approaches can potentially capture the best candidates from each method for every target MMD in the test set.

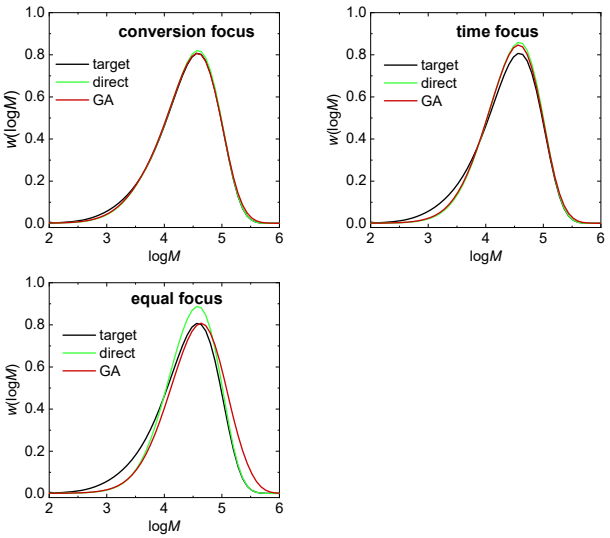

Figure 8. MMDs of the best recipe candidates for each focus, for a specific target MMD: $c_{m,0}$ = 2.995 mol ·L^{-1}; $c_{ini,0}$ = 17.84 mmol·L^{-1}, t = 260 min , 10 % training data

5 Conclusion

This article presents a novel multi-objective optimization approach using a genetic algorithm to optimize recipes for reverse engineering polymerization processes. This method effectively balances multiple objectives, includes achieving a target molar mass distribution, minimizing reaction time, and maximizing monomer conversion. The integration of pre-trained machine learning models significantly reduces the number of required experiments while maintaining high accuracy. The MOO approach offers a set of strong candidate recipes, allowing researchers to select the most suitable option based on their specific priorities. Furthermore, the generalizability of this framework paves the way for its application in optimizing various polymerization processes for different types of polymers. Future work will focus on validating this approach with other polymer systems.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – <466601458> – within the Priority Programme “SPP 2331: Machine Learning in Chemical Engineering”.

References

[1] H. Afanasyeva, 'Fuzzy learning classifiers systems for classification task', *Transport and Telecommunication*, **3**(3), 43–51, (2002).

[2] V. Bhaskar, Santosh Gupta, and Ajay Ray, 'Applications of multiobjective optimization in chemical engineering', *Reviews in Chemical Engineering*, **16**, (01 2000).

[3] L. Breiman, 'Random forests', *Machine Learning*, **45**(1), 5–32, (Oct 2001).

[4] L. Breiman, Jerome H. Friedman, Richard A. Olshen, and C. J. Stone, 'Classification and regression trees', *Biometrics*, **40**, 874, (1984).

[5] Javier Castro, Daniel Gómez, and Juan Tejada, 'Polynomial calculation of the shapley value based on sampling', *Computers & Operations Research*, **36**(5), 1726–1730, (2009).

[6] Silvia Curteanu, 'Direct and inverse neural network modeling in free radical polymerization', *Open Chemistry*, **2**(1), 113–140, (2004).

[7] Silvia Curteanu, Florin Leon, Andra-Maria Mircea-Vicoveanu, and Doina Logofătu, 'Regression methods based on nearest neighbors with adaptive distance metrics applied to a polymerization process', *Mathematics*, **9**(5), 547, (2021).

[8] Lucas Dall Agnol, Heitor Luiz Ornaghi, Francisco Monticeli, Fernanda Trindade Gonzalez Dias, and Otávio Bianchi, 'Polyurethanes synthetized with polyols of distinct molar masses: Use of the artificial neural network for prediction of degree of polymerization', *Polymer Engineering & Science*, **61**(6), 1810–1818, (2021).

[9] Michalis Doumpos and Constantin Zopounidis, 'Multi-objective optimization models in finance and investments', *J. of Global Optimization*, **76**(2), 243–244, (feb 2020).

[10] Marco Drache and Georg Drache, 'Simulating controlled radical polymerizations with mcpolymer—a monte carlo approach', *Polymers*, **4**(3), 1416–1442, (2012).

[11] Fabiano Fernandes and Liliane Lona, 'Neural network applications in polymerization processes', *Brazilian Journal of Chemical Engineering*, **22**, (07 2005).

[12] Andreas Feuerpfeil, Marco Drache, Laura-Alice Jantke, Timo Melchin, Jessica Rodríguez-Fernández, and Sabine Beuermann, 'Modeling semibatch vinyl acetate polymerization processes', *Industrial & Engineering Chemistry Research*, **60**(50), 18256–18267, (2021).

[13] Jelena Fiosina and Maksims Fiosins, 'Chapter 1: Cooperative regression-based forecasting in distributed traffic networks', in *Distributed Network Intelligence, Security and Applications*, ed., Q. A. Memon, 3–37, CRC Press, Taylor and Francis Group, (2013).

[14] Jelena Fiosina, Philipp Sievers, Marco Drache, and Sabine Beuermann, 'Polymer reaction engineering meets explainable machine learning', *Computers & Chemical Engineering*, **177**, 108356, (2023).

[15] Jelena Fiosina, Philipp Sievers, Gavaskar Kanagaraj, Marco Drache, and Sabine Beuermann, 'Polymerization reverse engineering for vac by multi-objective optimization', *Polymers*, **16**(7), 945, (2024).

[16] T. Ganesan, I. Elamvazuthi, Ku Zilati Ku Shaari, and P. Vasant, 'Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production', *Applied Energy*, **103**, 368–374, (2013).

[17] Luciana Ghiba, Elena Niculina Drăgoi, and Silvia Curteanu, 'Neural network–based hybrid models developed for free radical polymerization of styrene', *Polymer Engineering & Science*, **61**(3), 716–730, (2021).

[18] Eric S. Fraga Giovanna Fiandaca and Stefano Brandani, 'A multi-objective genetic algorithm for the design of pressure swing adsorption', *Engineering Optimization*, **41**(9), 833–854, (2009).

[19] Nyoman Gunantara, 'A review of multi-objective optimization: Methods and its applications', *Cogent Engineering*, **5**(1), 1502242, (2018).

[20] Srikant Gupta, Ahteshamul Haq, Irfan Ali, and Biswajit Sarkar, 'Significance of multi-objective optimization in logistics problem for multi-product supply chain network under the intuitionistic fuzzy environment', *Complex & Intelligent Systems*, **7**, 2119 – 2139, (2021).

[21] W. Härdle, *Applied Nonparametric Regression*, Cambridge University Press, Cambridge, 2002.

[22] John H Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press, 1992.

[23] Clifford M. Hurvich and Chih-Ling Tsai, 'Regression and time series model selection in small samples', *Biometrika*, **76**(2), 297–307, (1989).

[24] Chiho Kim, Rohit Batra, Lihua Chen, Huan Tran, and Rampi Ramprasad, 'Polymer design using genetic algorithm and machine learning', *Computational Materials Science*, **186**, 110067, (2021).

[25] Abdullah Konak, David W. Coit, and Alice E. Smith, 'Multi-objective optimization using genetic algorithms: A tutorial', *Reliability Engineering & System Safety*, **91**(9), 992–1007, (2006). Special Issue - Genetic Algorithms and Reliability.

[26] Haichen Li, Christopher R. Collins, Thomas G. Ribelli, Krzysztof Matyjaszewski, Geoffrey J. Gordon, Tomasz Kowalewski, and David J. Yaron, 'Tuning the molecular weight distribution from atom transfer radical polymerization using deep reinforcement learning', *Mol. Syst. Des. Eng.*, **3**, 496–508, (2018).

[27] R. Timothy Marler and Jasbir S. Arora, 'Beuermann for multi-objective optimization: new insights', *Structural and Multidisciplinary Optimization*, **41**(6), 853–862, (2010).

[28] Z.G. Meszena and A.F. Johnson, 'Modelling and simulation of polymerisation processes', *Computers & Chemical Engineering*, **23**, S375–S378, (1999). European Symposium on Computer Aided Process Engineering.

[29] Yousef Mohammadi and Alexander Penlidis, 'Polymerization data mining: A perspective', *Advanced Theory and Simulations*, **2**, (12 2018).

[30] Yousef Mohammadi, Mohammad Reza Saeb, Alexander Penlidis, Esmaiel Jabbari, Florian J. Stadler, Philippe Zinck, and Krzysztof Matyjaszewski, 'Intelligent machine learning: Tailor-making macromolecules', *Polymers*, **11**(4), (2019).

[31] Christoph Molnar, *Interpretable Machine Learning*, 2 edn., 2022. Online Book: https://christophm.github.io/interpretable-ml-book.

[32] Chitra Murugan and Sutha Subbaian, 'Multi-objective optimization for enhanced ethanol production during whey fermentation', in *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, pp. 1–7, (2022).

[33] P. Ngatchou, A. Zarei, and A. El-Sharkawi, 'Pareto multi objective optimization', in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pp. 84–91, (2005).

[34] Iqbal H. Sarker, 'Machine learning: Algorithms, real-world applications and research directions', *SN Comput. Sci.*, **2**(3), (mar 2021).

[35] David Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, 03 2015.

[36] David W. Scott, *Multivariate Density Estimation and Visualization*, 549–569, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[37] Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and I. Vlahavas, 'Multi-target regression via input space expansion: treating targets as inputs', *Machine Learning*, **104**, (07 2016).

[38] Jin Da Tan, Balamurugan Ramalingam, Swee Liang Wong, Jayce Cheng, Yee-Fun Lim, Vijila Chellappan, Saif A. Khan, Jatin Kumar, and Kedar Hippalgaonkar, 'Machine learning predicts conversion and molecular weight distributions in computer controlled polymerization', *ChemRxiv*, (2022). DOI: 10.26434/chemrxiv-2022-tlz53.

[39] Jie Zhang and Nikos Pantelelis, 'Modelling and control of reactive polymer composite moulding using bootstrap aggregated neural network models', *Chemical Product and Process Modeling*, **6**, (09 2011).

[40] Haifan Zhou, Yue Fang, and Hanyu Gao, 'Using active learning for the computational design of polymer molecular weight distributions', *ACS Engineering Au*, **4**(2), 231–240, (2024).