Preprints.org

Article

Not peer-reviewed version

Reverse Engineering of Radical Polymerizations by Multi-objective Optimization

Jelena Fiosina * , Philipp Sievers , Gavaskar Kanagaraj , Marco Drache , Sabine Beuermann *

Posted Date: 5 February 2024

doi: 10.20944/preprints202402.0199.v1

Keywords: polymerization reverse engineering; clustering; multi-objective optimization

Article

Reverse Engineering of Radical Polymerizations by Multi-Objective Optimization

Jelena Fiosina *, Philipp Sievers, Gavaskar Kanagaraj, Marco Drache and Sabine Beuermann *

J. Fiosina, G. Kanagaraj: Institute of Informatics, Clausthal University of Technology, Julius-Albert-Str. 4, Clausthal-Zellerfeld, Germany; Email Address: jelena.fiosina@tu-clausthal.de
P. Sievers, M. Drache, S. Beuermann; Institute of Technical Chemistry, Clausthal University of Technology, Arnold-Sommerfeld-Strasse 4, Clausthal-Zellerfeld, Germany; Email Address: sabine.beuermann@tu-clausthal.de
* Correspondence: jelena.fiosina@tu-clausthal.de (J.F.); sabine.beuermann@tu-clausthal.de (S.B.)

Abstract: Reverse engineering is applied to identify optimum polymerization conditions for the synthesis of polymer with pre-defined properties. The proposed approach uses multi-objective optimization (MOO) and provides multiple candidate polymerization procedures to reach the targeted polymer property. The objectives for optimization include maximal similarity of molar mass distributions (MMD) compared to the target MMD, minimal reaction time, and maximal monomer conversion. The method is tested for vinyl acetate radical polymerizations and can be adopted to other monomers. The data for the optimization procedure is generated by an in-house developed kinetic Monte-Carlo (kMC) simulator for a selected recipe search space. The proposed reverse engineering algorithm comprises several steps: kMC simulations for the selected recipe search space to derive initial data, performing MOO for a targeted MMD, and identification of the Pareto optimal space. The last step uses a weighted sum optimization function to calculate the weighted score of each candidate polymerization conditions. To decrease the execution time clustering of the search space based on MMDs is applied. The performance of the proposed approach was tested for various target MMDs. The suggested MOO-based reverse engineering provides multiple recipe candidates depending on competing objectives.

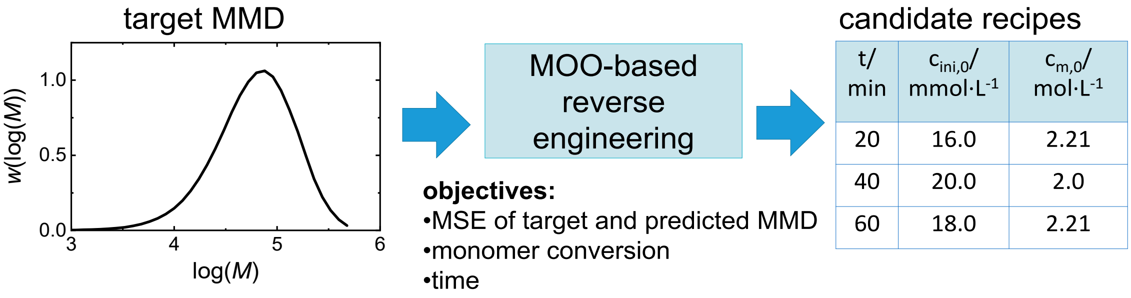
Keywords: polymerization reverse engineering; clustering; multi-objective optimization

1. Introduction

Radical polymerizations are known to be very robust and to provide access to a wide range of polymers with largely differing properties, which are defined by the process conditions. The strong correlation between production process and material properties is due to the complex reaction mechanism consisting of a large number of elemental reactions even for homopolymerizations with a single monomer [1]. The kinetics of the elemental reactions are strongly dependent on the process conditions. Therefore, the prediction of a suitable radical polymerization process to obtain a polymer with targeted properties is challenging. To allow for on demand polymer synthesis, at first sight it appears highly attractive to apply simulations of polymerization processes, e.g., employing differential equations [2–4] or kinetic Monte-Carlo (kMC) methods [5–8]. Simulations are particularly valuable, because detailed information on polymer microstructure at each time moment is available, which is not accessible from polymerization processes. However, this type of simulations cannot be run backwards and on demand suggestion of polymerization conditions to obtain a pre-defined polymer is not feasible. To overcome this issue, reverse engineering has the potential to provide several solutions, as opposed to a single one to one relation between polymerization variables and microstructural properties [9]. From the input variables, a polymerization process model predicts the concentration vs. time profiles and the polymer properties. Inverse modeling, on the other hand, is more difficult and calls for optimization strategies. In order to determine the optimal input values for systems with complex reaction mechanisms that provide pre-defined reaction outputs (such as pre-set conversion, yield, and/or other product properties), it was suggested to intelligently explore the reaction condition search space [10]. Further, it was proposed to solve reverse engineering problems using machine learning (ML) based prediction [11], in which ML regression models based on the random forest algorithm and a multivariate and multi-target regression problem [12] were applied. The model took a targeted MMD and predicted the initial polymerization conditions to produce polymer with this targeted property, minimizing the errors in predicted recipes. However, the prediction of the recipe was based on the MMD only, and not optimized with respect to multiple objectives like reaction time or conversion.

Pareto or multi-objective optimization (MOO) [13] is the process of maximizing or minimizing many objective functions while taking a set of constraints into consideration. Numerous scientific domains, such as engineering [14], economics [15], and logistics [16], need the use of MOO when making optimum decisions facing trade-offs between two or more competing objectives. An increasing use of MOO has been seen in chemical engineering [17,18]. In 2009, Fiandaca et al. [19] used genetic algorithm-based MOO to optimize a pressure swing adsorption process based on the maximization of two objectives: nitrogen recovery and nitrogen purity. In 2013, Ganesan et al. [20] carried out MOO of the combined carbon dioxide reforming and partial-oxidation of methane with respect to three objectives. MOO utilized a gravitational search algorithm and particle swarm optimization to tackle the problem. With respect to technical application it appears highly important to solve the reverse engineering problem as an optimization task with multiple contradicting objectives [13]. Various ML-based optimization strategies were addressed for the purpose of reverse engineering polymerization processes [21–23]. A genetic algorithm-based optimizer is proposed by Mohammadi et al. [9] to generate a variety of polymerization recipes at random and to send them to the kMC simulator for error evaluation.

This study provides a MOO-based reverse engineering approach, not only ensuring that the targeted MMD is obtained by means of minimizing the mean squared error (MSE), but also providing minimal reaction time and maximal monomer conversion. Frequently, there are several recipes for obtaining similar MMDs, which are referred to as candidate recipes in the following. The solution of the MOO approach is referred to as polymerization recipe, which includes temperature, reaction time as well as initial monomer and initiator concentrations. In the proposed MOO approach, the data for the selection of the optimal recipes from the search space is based on kMC simulations, as previously reported for training ML models [11]. The simulation provides the dependence of monomer conversion and the corresponding MMD on the polymerization conditions and reaction time. The proposed reverse engineering algorithm consists of several steps. First, the kMC simulations are run for the selected recipe search space to derive the MMDs and monomer concentrations as an input for the MOO step. Then, MOO is applied for a given target MMD and the Pareto optimal space is found on the base of the search space. In the last step, a weighted sum optimization function is used to calculate the weighted score of each candidate recipe, which is used for evaluating the solutions. The best candidate has the smallest score. To accelerate the MOO procedure an additional search space clustering on the basis of MMDs is considered. The approach pursued is illustrated in Figure 1. The method is tested for the following model system: chemically initiated vinyl acetate (VAc) radical polymerization model system at 60 °C using tert-butyl peroxypivalate as initiator.



objectives:
- MSE of target and predicted MMD
- monomer conversion
- time

candidate recipes

t/ min	$c_{ini,0}$/ mmol·L^{-1}	$c_{m,0}$/ mol·L^{-1}
20	16.0	2.21
40	20.0	2.0
60	18.0	2.21

Figure 1. Illustration of the polymerization reverse engineering approach by means of MOO.

2. Reverse Engineering Modeling Approach

2.1. Model Development

This section provides a formal description of solving a reverse engineering problem with an MOO approach. State of the art MOO methods [24] like genetic algorithms require to generate and evaluate new recipes in each step. Due to the high number of required steps, it is very demanding to perform on-line kMC simulations in loop [9].

Instead, here a recipe search space $\mathbf{R}$ is selected, where each recipe $\mathbf{r} \in \mathbf{R}$ consists of reaction time t, the initial monomer concentration $c_{\mathrm{m,0}}$, and the initial initiator concentration $c_{\mathrm{ini,0}}$. Then, the corresponding monomer concentration $c_{\mathrm{m}}(\mathbf{r})$ and molar mass distribution $MMD(\mathbf{r})$ are obtained for each $\mathbf{r} \in \mathbf{R}$ via kMC simulation. In the MOO approach, the input is a target MMD, MMD^{target}, and the output is a set of optimal candidate recipes $\mathbf{R}^*$: $\mathbf{r}_1^*$, $\mathbf{r}_2^*$, …,$\mathbf{r}_n^*$. $\mathbf{R}^*$ is a subset of the recipe search space $\mathbf{R}$, $\mathbf{R}^* \subset \mathbf{R}$ with MMD($\mathbf{r}_i^*$) being close to MMD^{target} as evaluated on the basis of the MSE as well as the maximal conversion and minimal time.

The optimization variables are presented in Table 1. The lower and upper limits for the variables $c_{\mathrm{mon,0}}$, $c_{\mathrm{ini,0}}$, and t are defined by the simulated data. In Table 2 for a specific recipe $\mathbf{r}$, the simulated values $c_{\mathrm{m}}(\mathbf{r})$ and $MMD(\mathbf{r})$ are used to calculate the values of three optimization objective functions: objective for the reaction time, $f_{\mathrm{t}}(\mathbf{r})$, and objective for the mean squared error (MSE) between MMD^{target} and the predicted $MMD(\mathbf{r})$, f_{MSE}. $f_{\mathrm{cm}}(\mathbf{r})$ is used to turn the maximization problem of monomer conversion $f_{\mathrm{conv}}(\mathbf{r}) = (c_{\mathrm{m,0}} - c_{\mathrm{m}}(\mathbf{r})) / c_{\mathrm{m,0}}$ into a minimization problem, in which $1 - f_{\mathrm{conv}}(\mathbf{r})$ is minimized (Table 2).

Table 1. Optimization variables.

variables	description	restrictions
$c_{\mathrm{m,0}}$	initial monomer concentration	$c_{\mathrm{m,0(min)}} \leq c_{\mathrm{m,0}} \leq c_{\mathrm{m,0(max)}}$
$c_{\mathrm{ini,0}}$	initial initiator concentration	$c_{\mathrm{ini,0(min)}} \leq c_{\mathrm{ini,0}} \leq c_{\mathrm{ini,0(max)}}$
t	reaction time	$t_{\mathrm{min}} \leq t \leq t_{\mathrm{max}}$
$\mathbf{r}$=[$c_{\mathrm{m,0}}$, $c_{\mathrm{ini,0}}$, t]	initial recipe	

Table 2. Optimization objective functions and their calculation on the basis of simulated data.

objectives	description
$\min\limits_{\mathbf{r}} f_{\mathrm{MSE}}(\mathbf{r}) = \min\limits_{\mathbf{r}}\ [\mathrm{MSE}(MMD^{target}, MMD(\mathbf{r}))]$	minimal MSE, where $MMD(\mathbf{r})$ is simulated
$\min\limits_{\mathbf{r}} f_{\mathrm{cm}}(\mathbf{r}) = \min\limits_{\mathbf{r}}[1 - f_{\mathrm{conv}}(\mathbf{r})] = \min\limits_{\mathbf{r}} \frac{c_{\mathrm{m}}(\mathbf{r})}{c_{\mathrm{m,0}}}$	minimal relative monomer concentration
$\min\limits_{\mathbf{r}} f_{\mathrm{t}}(\mathbf{r}) = \min\limits_{\mathbf{r}}\ t$	minimal reaction time (directly from $\mathbf{r}$)

The final decision takes user preferences into account by assigning specific weights to the objectives applying the weighted sum method [24,25]. Thus, the multi-objective function can be represented in a single-objective way. Then, the values of this function are calculated for each candidate recipe and the recipes with minimal values of the objective function are selected as a set of optimal solutions. A weight w_i is assigned to each normalized objective function f_i as follows:

$$\min_{\boldsymbol{r}} f(\boldsymbol{r}) = \min_{\boldsymbol{r}} \sum_i w_i \cdot f_i(\boldsymbol{r}) \tag{1}$$

where $\sum_i w_i = 1$, $i \in \{\text{MSE, cm, } t\}$, $\mathbf{r} \in \mathbf{R}$, and $\mathbf{R}$ is a polymerization recipe space. For clarity of presentation, the weight w_{cm} of the objective f_{cm} is also referred to as the weight of the conversion objective. If required the number of the objectives can be increased, e.g. the conversion of initiator can be added.

The steps of the proposed algorithm are presented in Figure 2 as direct approach. First, a search space $\mathbf{R}^{\text{S}} \subset \mathbf{R}$ is selected (for details see Section 2.2) and for each $\mathbf{r} \in \mathbf{R}^{\text{S}}$ MMD($\mathbf{r}$) and $c_{\text{m}}(\mathbf{r})$ are obtained via kMC simulations. Then, MOO is performed over $\mathbf{R}^{\text{S}}$. First, the objective function values are calculated as follows: the values for the objective $f_{\text{t}}(\mathbf{r})$ are already included in $\mathbf{r}$ as t, the values of the objective $f_{\text{cm}}(\mathbf{r})$ are calculated in advance for all possible $\mathbf{r}$ according to Table 2, and the values of the objective f_{MSE} are specified by MMD^{target}. Further, based on the calculated objective function values, the Pareto front points $\mathbf{R}^{\text{par}} \subset \mathbf{R}^{\text{S}}$ leading to MMD^{target} are identified. For this, the points from $\mathbf{R}^{\text{S}}$ are represented in the Pareto optimal space, with coordinates specified by the three objectives. The Pareto front points are found in this Pareto optimal space, such that one value of the objective function cannot be improved without downgrading the value of another objective function. Finally, for the Pareto front points $\mathbf{R}^{\text{par}}$, the weights of each objective function are defined and a set of the best recipe candidates $\mathbf{R}^{*} \subset \mathbf{R}^{\text{par}}$ according to Eq. 1 is selected.

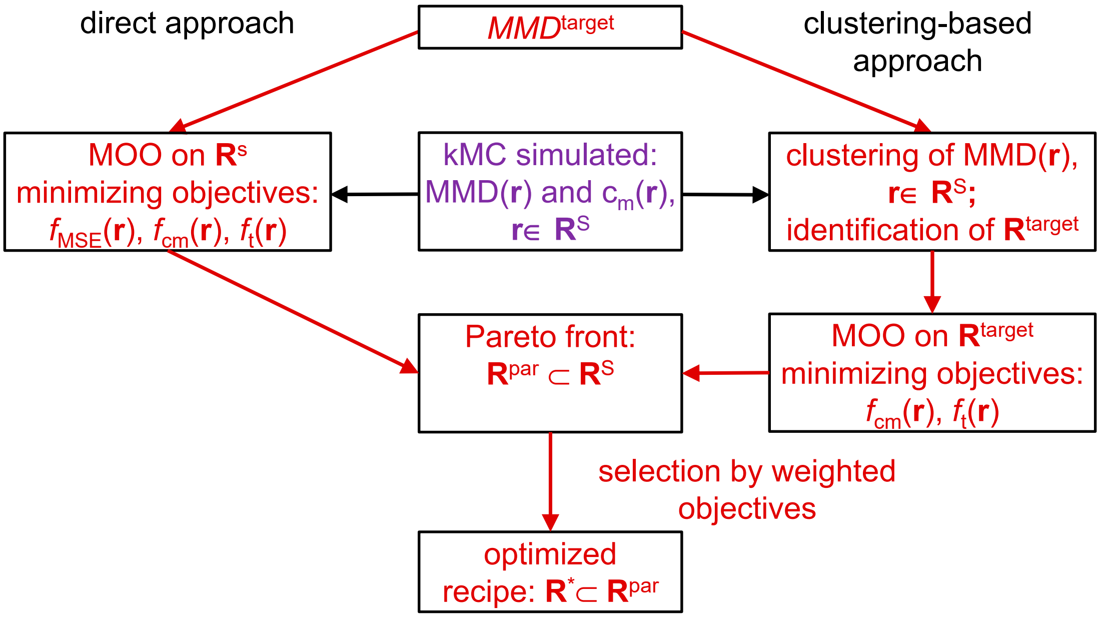

direct approach
MMD^target
clustering-based approach
MOO on R^s minimizing objectives: f_MSE(r), f_cm(r), f_t(r)
kMC simulated: MMD(r) and c_m(r), r∈ R^S
clustering of MMD(r), r∈ R^S; identification of R^target
Pareto front: R^par ⊂ R^S
MOO on R^target minimizing objectives: f_cm(r), f_t(r)
selection by weighted objectives
optimized recipe: R*⊂ R^par

Figure 2. Direct Pareto optimization vs. clustering supported Pareto optimization.

The above-described algorithm is improved with respect to the optimization time by means of clustering the search space, which is illustrated in Figure 2 as the clustering-supported approach. Clustering divides the search space $\mathbf{R}^{\text{S}}$ into a number of clusters as illustrated in Figure 3. First, the search space $\mathbf{R}^{\text{S}}$ is clustered on the base of MMD($\mathbf{r}$), which allows for selecting a cluster $\mathbf{R}^{\text{target}} \subset \mathbf{R}^{\text{S}}$ containing the MMDs, which are the closest to MMD^{target}. In general, a larger number of clusters leads to a smaller number of MMDs per cluster gaining higher similarity of the distributions. However, there is less space for optimization regarding other objectives, e.g., such as polymerization time and monomer conversion. For this reason, an appropriate trade-off between the number of clusters and its size has to be identified. Upon appropriate clustering a cluster for the target MMD (Figure 2, red arrows) $\mathbf{R}^{\text{target}}$ is found. Then, by MOO the search space is reduced to the number of Pareto front points $\mathbf{R}^{\text{par}} \subset \mathbf{R}^{\text{target}}$. Finally, after defining objective weights, the best recipe candidates $\mathbf{R}^{*} \subset \mathbf{R}^{\text{par}}$ are found according to Eq. 1. Since MOO is applied to a single cluster $\mathbf{R}^{\text{target}} \subset \mathbf{R}^{\text{S}}$, which is considerably smaller than $\mathbf{R}^{\text{S}}$, the optimization time is significantly reduced.

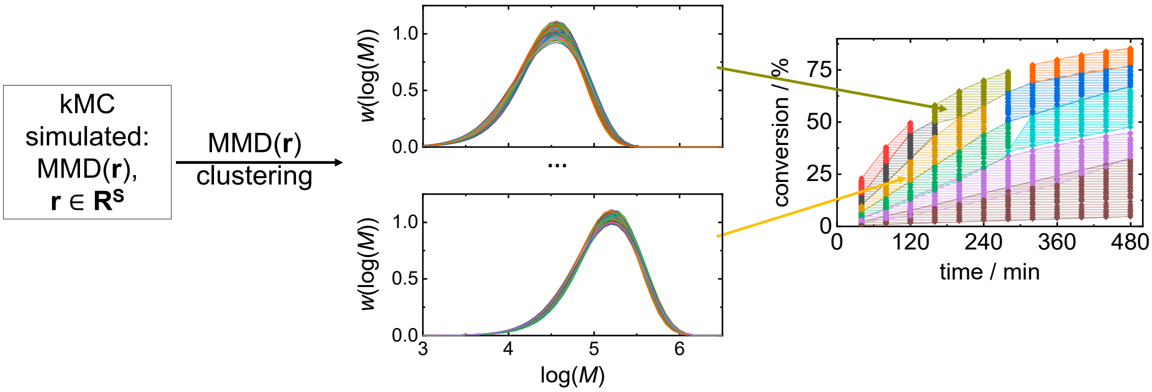


Figure 3. Illustration of the clustering approach of the search space. In the middle two exemplary clusters of MMDs are presented, on the right all clusters defined are given.

Different methods can be applied for clustering the search space on the basis of MMD. Clustering of distributions and their representation in histograms is an important topic, which attracted a lot of attention, because of specific metrics, which should be used to compare the distributions. One of the most popular and fast clustering methods is the kMeans method [26]. A modified kMeans clustering algorithm was applied to the clustering of histograms [27]. Further, a novel non-parametric clustering algorithm of empirical probability distributions was proposed [28]. Here, the classical kMeans clustering method was used. This algorithm starts with a random separation of the MMDs into clusters. At each step, it recalculates the centroids of each cluster and relocates the data points to the new centroids. The clustering process finishes when the clusters are stable or the given number of iterations is reached. In this study, the simplest Euclidean distances are used for calculation of the distances between multi-dimensional data points, while specific metrics for clustering of distributions are also available [27,29].

There are different strategies for the data generation for the MOO procedure: use of exclusively in advance generated data from kMC simulation (Figure 2, blue arrows), on demand kMC-generated data, ML-generated data, kMC-based and ML-generated hybrid data sets, etc. Currently, as a first step, the focus is exclusively on the use of kMC-simulated data.

2.2. Data acquisition and Processing

The in-house developed kMC simulator mcPolymer was used to carry out the simulations required for the generation of polymerization data according to the search space [30]. The simulator allows for exporting the concentration profiles of all reactants and products as well as microstructural data like MMD, chain composition and branching of all polymeric species involved in the process. The simulator output was adapted to be well-machine readable. The data were filtered, further abstracted, logically connected, and stored in the well-structured no-SQL database MongoDB. The kMC simulated MMDs and monomer concentrations were obtained for the selected search space $\mathbf{R}^{\text{S}}$, allowing for MOO and subsequent finding of the weighted optimal solution.

The kMC simulations were performed for radical polymerizations with VAc as monomer, tert. butyl peroxypivalate as initiator, and methanol as solvent. The simulations are based on a full kinetic model for the VAc radical polymerization containing all elemental reactions [31]. The following polymerization conditions were used: constant temperature of 60 °C, $c_{\text{ini,0}}$ in the range of

1.0 to 20.0 mmol·L^{-1}, and $c_{\mathrm{m,0}}$ in the range of 2.0 to 5.0 mol·L^{-1} with uniformly distributed grid size of $c_{\mathrm{ini,0}}$ (geometrically scaled grid points) and $c_{\mathrm{m,0}}$ (arithmetic scaled grid points), resulting in 225 simulations of the process. The geometric scale was selected for $c_{\mathrm{ini,0}}$ to put more attention on the small values of this parameter. The polymerization process was simulated for a constant reaction time of 6 hours and the properties of interest were recorded every 20 minutes, thus, obtaining in total 18 data points at different time moments for each investigated property. Thus, the data set contains 4050 different MMDs.

For training the ML prediction models for single-objective reverse engineering the obtained data set was divided into training and test set in proportion of 80:20. The same data was used for MOO, again taking 80 % of the data as training set for the search space $\mathbf{R}^{\mathrm{S}}$ and 20 % of the data as test set $\mathbf{R}^{\mathrm{test}}$. The test set $\mathbf{R}^{\mathrm{test}}$ contains 810 recipes, which correspond to a set $\mathbf{MMD}^{\mathrm{test}}$ consisting of 810 kMC-simulated MMDs. The evaluation of all optimization approaches was performed with $\mathbf{MMD}^{\mathrm{test}}$ with each element serving as MMD^{target}. In order to test the MMO approach a single MMD is selected from $\mathbf{MMD}^{\mathrm{test}}$ and used as MMD^{target} for the MMO approach. The performance of the MMO is evaluated by testing it with every MMD from $\mathbf{MMD}^{\mathrm{test}}$.

3. Results and Discussion

In order to compare results obtained via the previously described ML modeling-based reverse engineering strategy [11] with data from the MOO approach introduced in this work, the technique reported for butyl acrylate was adopted for the polymerization of vinyl acetate. Previously, it was described how an ML regression model provides a recipe $\mathbf{r} = (c_{\mathrm{m,0}},\ c_{\mathrm{ini,0}},\ T)$ for a fixed time for a given MMD^{target}. $c_{\mathrm{m,0}}$ and $c_{\mathrm{ini,0}}$ are initial concentrations of the monomer and the initiator, respectively, and T is the polymerization temperature. The prediction was performed with the random forest method for a target MMD to minimize the errors in the predicted recipe $\mathbf{r}$(MMD). In the case of VAc polymerizations, the temperature is kept constant at T = 60 °C and the reaction time t was predicted by the model: $\mathbf{r} = (c_{\mathrm{m,0}},\ c_{\mathrm{ini,0}},\ t)$. The reverse engineering result for a sample MMD^{target} presented in Figure 4 corresponds to the recipe ($c_{\mathrm{m,0}}$; $c_{\mathrm{ini,0}}$; t)/ (2.0 mol/L; 1.9 mmol/L; 140 min). Only very small differences between the target and predicted MMD as well as the predicted recipe are seen. The overall performance of the developed model was evaluated using the R^2 determination coefficient, which is a measure for the quality of the prediction by the ML model. Keeping in mind that R^2=1 indicates perfect prediction, it is remarkable to note that a rather low value of 0.78 for R^2 is obtained while visual inspection of the MMDs for the representative example given in Figure 4 indicates only minor differences. However, this ML-based approach does not consider multiple contradicting optimization objectives and does not provide multiple alternative solutions as proposed in the Pareto optimization.

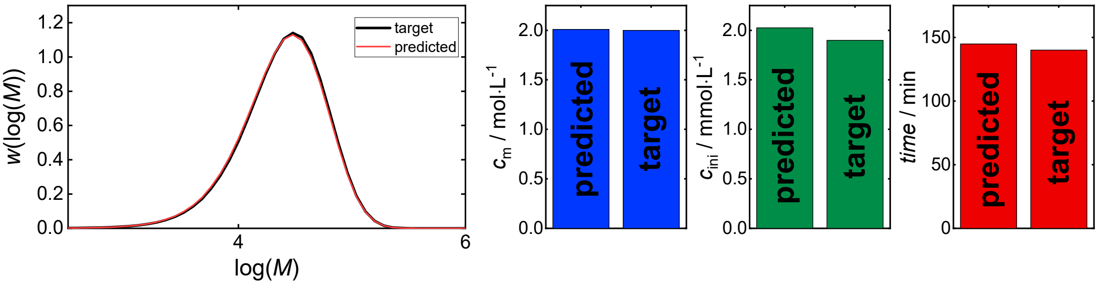


Figure 4. An example of reverse engineering with ML modeling comparing the MMD (left) and the recipe (right) for a single MMD^{target}, with MMD^{target} being an element of $\mathbf{MMD}^{\mathrm{test}}$. The target recipe is the recipe associated with MMD^{target}.

3.1. Direct Pareto Optimization

Figure 5A shows all Pareto front points with three objectives f_i, $i \in$ {MSE, cm, t}. After definition of objective weights, the optimal solution from the Pareto front points is selected, which satisfies the weighted score calculated with Eq. 1 best. Several combinations of objective weights are considered: time focused, conversion focused, MSE focused. For example, in the time focused case, the weight of time is higher than the other weights. To identify the optimal recipe leading to the target MMD, the MSE should have a higher weight than the other objectives. To reduce the number of Pareto front points an MSE limit is chosen. As an example, the data in Figure 3 was obtained with an MSE limit of 2·10^{-3}.

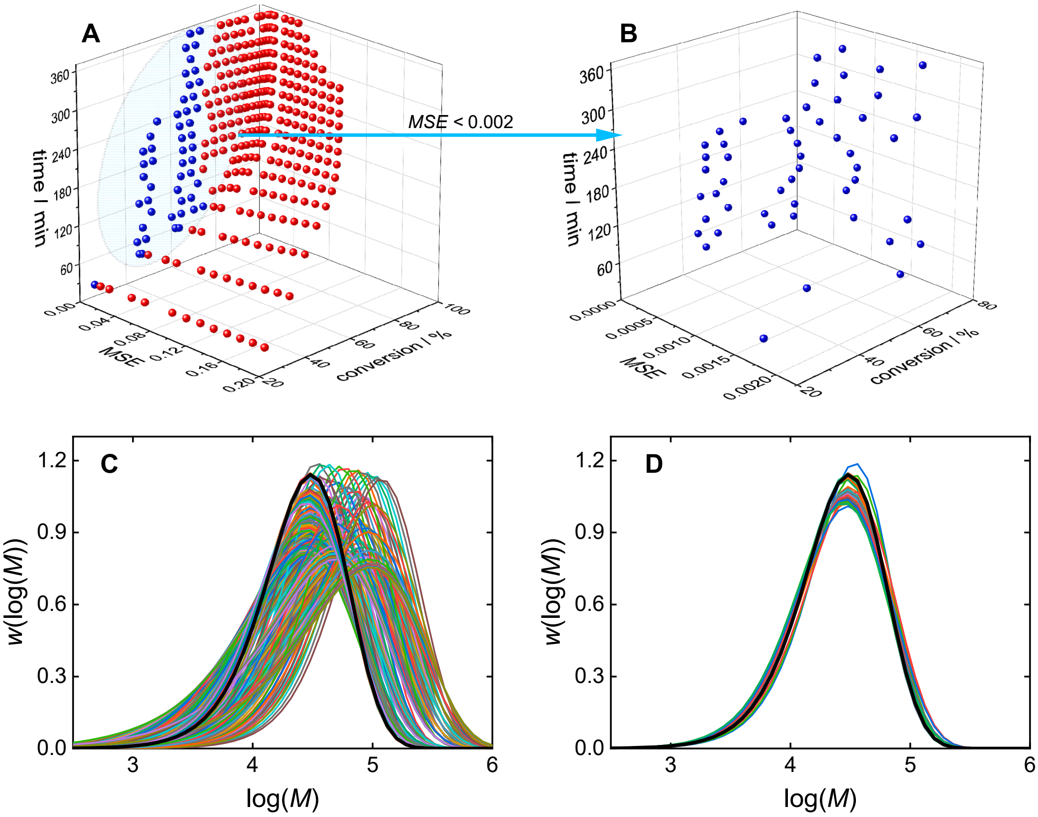


Figure 5. Pareto front points (A), filtered Pareto front points (B), MMDs for all Pareto front points (C), and MMDs for filtered Pareto front points (D). The target MMD is given in bold black line, the MSE limit was set to 2·10^{-3}.

The optimization procedure is illustrated in Figure 5 for an example target MMD, which corresponds to the recipe ($c_{m,0}$; $c_{ini,0}$; t) / (2.0 mol/L; 1.9 mmol/L; 140 min). Rather than presenting the objective f_{cm}, which represents the minimized monomer concentration, the conversion is shown in Figure 5A,B. Figure 5B demonstrates the reduction of the number of Pareto front points from 340 points (A) to 46 points (B) by filtering with the MSE limit of 2·10^{-3}. The MMDs of all Pareto front points and of the filtered Pareto front points are given in Figure 5C,D, respectively. The filtering approach allows for considering only MMDs of similar shape. Figure 6 shows the impact of different MSE limits on the number of Pareto front points. Even with an MSE limit of 10^{-3} the average number of points is around 30, which is a relevant number for consideration of other objectives. The minimal number of filtered Pareto points for the MSE limit of 10^{-3} is about 10.

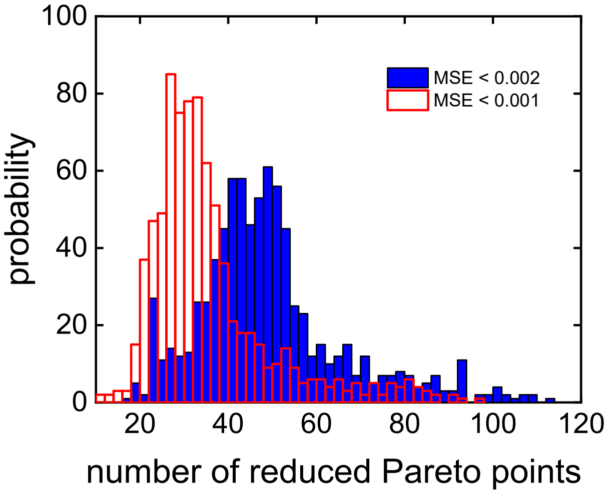


Figure 6. The influence of the MSE limit on the distribution of the number of filtered Pareto front points based on **MMD**$^{\text{test.}}$

Figure 7 allows for comparison of MSEs obtained via ML prediction-based (left) and MOO-based approaches (right) for reverse engineering calculated for all elements of **MMD**$^{\text{test}}$. Minimal MSEs are preferable. For the ML-based prediction (left), the MSE values of almost all *MMD*$^{\text{target}}$ from **MMD**$^{\text{test}}$ are less than 10^{-3}, while most MSEs obtained by the MOO-based approach with an MSE limit of $2{\cdot}10^{-3}$ are by two orders of magnitude lower. Most values are less than 10^{-5}.

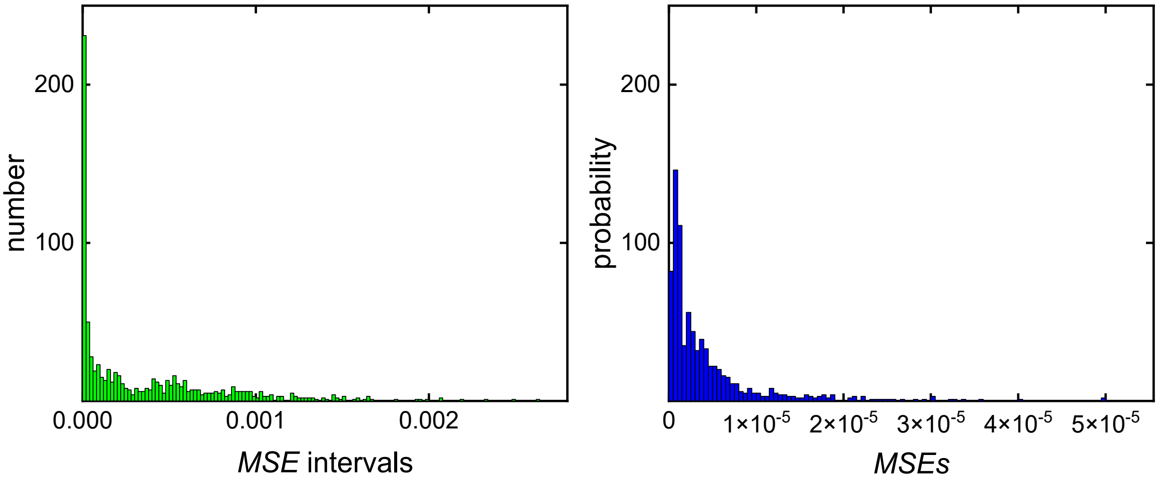


Figure 7. MSEs of the MMDs of the recipes obtained by ML prediction based (left) and MOO (right) approaches on the basis of **MMD**$^{\textbf{test.}}$

The next step is to evaluate the candidate recipes based on the weighted sum of the objective values according to Eq. 1. Figure 8 and Table 3 present the candidate recipes for the different combinations of objective weights indicated. In Figure 8, a color code is assigned to the individual candidate recipes according to the score of the weighted sum of the values of the three objective functions f_{MSE}, f_{cm}, f_{t} calculated according to Eq. 1. For the weights $w_{\mathrm{MSE}} = 0.9$, $w_{\mathrm{cm}} = 0.02$, and $w_{\mathrm{t}} = 0.08$ in Figure 8A the focus is on the MSE. As a minimization problem is solved, a minimal score value (visualized by big pink points) determines the best candidates, given by the recipes with IDs 0, 15, 16 in Table 3. Other weight combinations with focus on the monomer conversion in Figure 8B and with equal weights in Figure 8C demonstrate, which recipe candidates are selected depending on specific requirements.

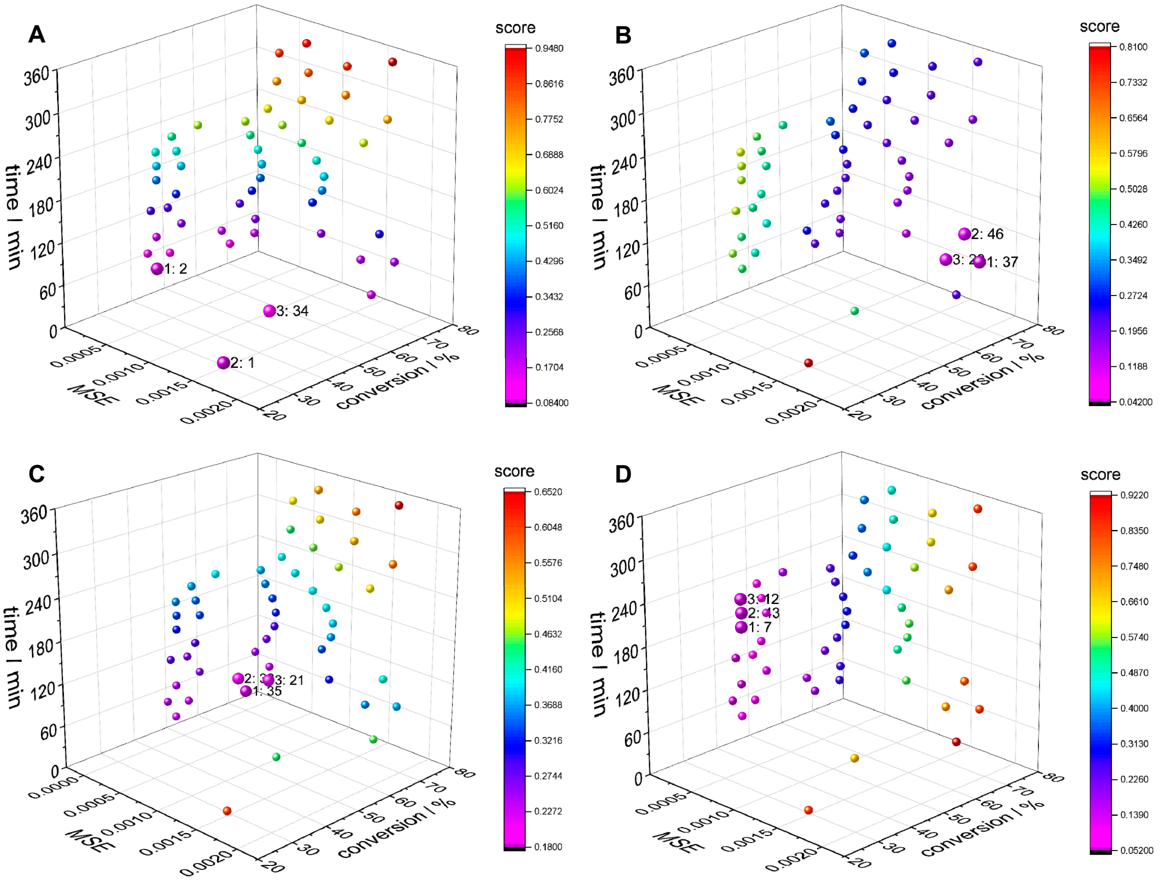


Figure 8. Influence of the combination of objective weights on the selection of the optimal recipes for a sample *MMD*^{target}: time focused (w_{MSE} = 0.05, w_{cm} = 0.05, w_t = 0.9) (A), monomer conversion focused (w_{MSE} = 0.01, w_{cm} = 0.8, w_t = 0.19) (B), equal weights (w_{MSE} = 1/3, w_{cm} = 1/3, w_t = 1/3) (C), MSE focused (w_{MSE} = 0.9, w_{cm} = 0.09, w_t = 0.01) (D). The best three solutions are shown with their IDs for each case.

Table 3. The best recipes for each combination of objective weights (the focal weight is given in bold).

IDs of the best recipe		1	34	35		46	20	33
	w_i	time focus (A)			w_i	conversion focus (B)		
$c_{m,0}$ / mol·L^{-1}		2.0	2.21	2.21		2.21	2.43	2.21
$c_{ini,0}$ / mmol·L^{-1}		20	20	20		8.4	16.1	10.5
time / min	**0.9**	20	40	60	0.19	120	80	100
MSE /·10^{-3}	0.05	1.63	1.37	0.35	0.01	1.72	1.58	1.16
conversion / %	0.05	25.9	46.6	61.8	**0.8**	71.3	69.6	68.7

IDs of the best recipe		35	32	21		43	12	0
	w_i	equal weights (C)			w_i	MSE focus (D)		

$c_{m,0}$ / mol·L^{-1}		2.21	2.21	2.21		2.0	2.0	2.0
$c_{ini,0}$ / mmol·L^{-1}		20.0	10.5	13.0		12.4	10.0	8.5
time / min	**1/3**	60	80	80	0.01	180	200	60
MSE / ·10^{-5}	**1/3**	35	31	54	**0.9**	0.086	0.032	0.64
conversion / %	**1/3**	61.8	60.2	64.4	0.09	48.3	48.2	45.1

3.2. Clustering Supported Pareto Optimization

The clustering-supported optimization is performed for the same MMD^{target} as in the previous section. It is assumed that all points contained in the target cluster $\mathbf{R}^{\text{target}}$ have satisfactory MSEs. For this reason, only the two objectives f_{cm} and f_{t} have been chosen to be optimized. Figure 9 shows $\mathbf{R}^{\text{target}}$ in the coordinate space, which corresponds to the objective functions. The Pareto front points $\mathbf{R}^{\text{par}} \subset \mathbf{R}^{\text{target}}$ are indicated by colored markers. Table 4 presents all Pareto front points with the corresponding recipes and objective weights.

To find the weighted solution for a time focused result with $w_{\text{cm}} = 0.2$ and $w_{\text{t}} = 0.8$ the candidate with ID 1 reaching a conversion of 47 % at a reaction time of 40 min is best. If conversion is considered to be more important than reaction time as in the case of $w_{\text{cm}} = 0.8$ and $w_{\text{t}} = 0.2$, the candidate with ID 4 is the best solution leading to a conversion of 77 % with a reaction time of 100 min.

MMD clustering focuses only on a single objective function f_{MSE} when identifying the most similar cluster for the target MMD. The accuracy is determined on the basis of the MSE, which represents the deviation of the selected solution from MMD^{target} and depends on the cluster size, and consequently, on the number of clusters. For a small number of clusters, every cluster contains a broader spectrum of MMDs as compared to a large number of clusters and leads to an increase in the MSEs of MMD^{target} and the MMDs contained in this cluster. However, the cluster covers a larger value range of the objective function, allowing for a broader range of predicted recipes. Figure 10 depicts how the MMD^{target} can be reached for clustering the search space into 20, 40, and 60 clusters, respectively. Figure 10A–C shows the points of the target clusters in the objective function space. The shape of the clusters is similar. As the number of clusters grows, the number of candidates per cluster decreases, because there are fewer points fitting the required accuracy with respect to the MMD. Therefore, the size of the whole cluster shrinks. The red points in Figure 10 mark the Pareto front points, from which the optimal candidates are selected.

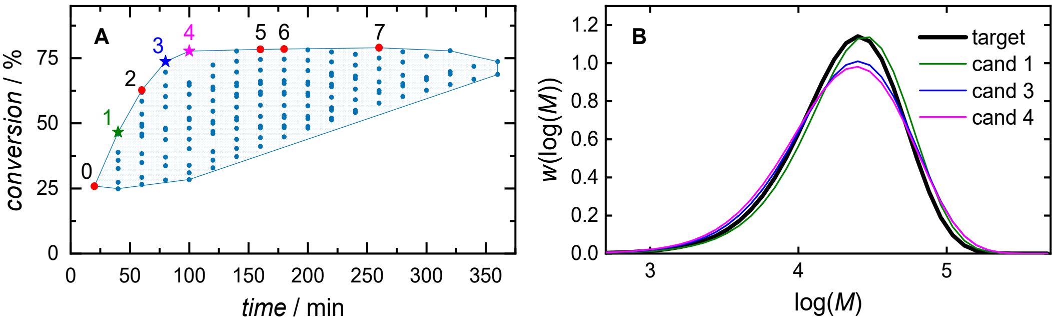


Figure 9. Clustering supported Pareto optimization. A: Illustration of the target cluster $\mathbf{R}^{\text{target}}$. The Pareto front points $\mathbf{R}^{\text{par}} \subset \mathbf{R}^{\text{target}}$ are marked (colored markers) and are listed in Table 4. A time focused result ($w_{\text{cm}} = 0.2$, $w_{\text{t}} = 0.8$), an equal weighted result ($w_{\text{cm}} = 0.5$, $w_{\text{t}} = 0.5$) and a conversion focused result ($w_{\text{cm}} = 0.8$, $w_{\text{t}} = 0.2$) are highlighted (1, 2, 3). The corresponding MMDs in comparison with MMD^{target} are shown in B.

doi:10.20944/preprints202402.0199.v1

Table 4. Pareto front points $\mathbf{R}^{\mathrm{par}}$ and their recipes and objective functions. Every highlighted point (bold letters) is the best result for the corresponding example objective weights.

objective weights	cand. ID	$c_{\mathrm{m},0}$ / mol·L^{-1}	$c_{\mathrm{ini},0}$ / mmol·L^{-1}	time / min	MSE / 10^{-3}	conversion / %
	0	2.00	20.0	20	1.63	25.9
time focus: ($w_{\mathrm{cm}} = 0.2$, $w_{\mathrm{t}} = 0.8$)	**1**	**2.21**	**20.0**	**40**	**1.37**	**46.6**
	2	2.43	20.0	60	1.94	62.7
equal weights: ($w_{\mathrm{cm}} = 0.5$, $w_{\mathrm{t}} = 0.5$)	**3**	**2.43**	**20.0**	**80**	**1.81**	**73.7**
conversion focus: ($w_{\mathrm{cm}} = 0.8$, $w_{\mathrm{t}} = 0.2$)	**4**	**2.43**	**16.1**	**100**	**2.61**	**77.7**
	5	2.43	6.86	160	2.79	78.3
	6	2.43	5.54	180	2.83	78.5
	7	2.43	2.91	260	2.98	79.0

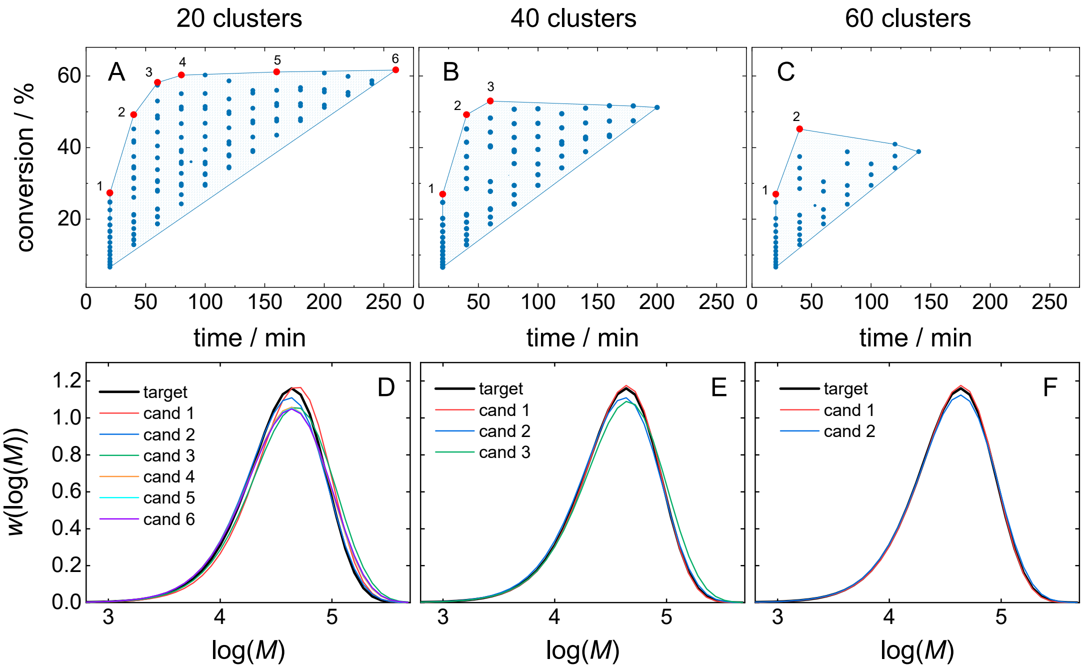


Figure 10. Cluster size dependence. A-C: Clusters for *MMD*$^{\mathrm{target}}$ with a total number on 20 (A), 40 (B) or 60 (C) clusters. D-F: MMDs for the Pareto front points.

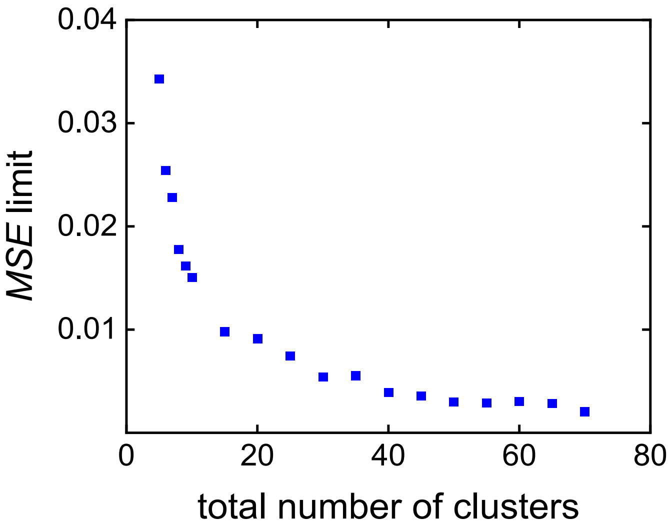
Figure 10D–F shows the MMDs for all Pareto front points. With increasing number of clusters, the MSE of *MMD*$^{\mathrm{target}}$ and Pareto front MMDs decreases. The objective function value range also decreases as illustrated in Table 5. For the example, given f_{MSE} of *MMD*$^{\mathrm{target}}$ improves by a factor of 100 when going from a total number of 20 clusters to a total number of 60 clusters. This improvement is achieved by a loss in conversion, which is reduced from 58 % to 27 %. In this example, reaction time is also lowered with increasing number of clusters. The last two rows in Table 5 provide the scores calculated by Eq. 1 for the best recipes obtained with clustering of the search space into 20, 40, and 60 clusters for two cases with a different number of objectives. Note, that only the scores obtained with the same combination of weights (given in a single row of Table 10) are comparable. For equal weights of two objectives (w_{cm} = 0.5, w_t = 0.5) the minimal score of 0.39 was obtained with clustering into 40 clusters. Although the optimization has been carried out within one cluster using only two objectives, the results can be related to the direct Pareto optimization, including the MSE objective. It is shown that for equal weight of three objectives (w_{MSE} = 1/3, w_{cm} = 1/3, w_t = 1/3) the minimal score of 0.30 was also obtained with the same clustering into 40 clusters.

The comparison of the direct and the clustering-supported Pareto approach is based on the MSE limit. For the direct approach, the MSE limit can be chosen by the operator. For the clustering-supported approach, the individual MSE limit for a given MMD^{target}, which is evidently depending on the number of clusters, is defined as the maximal MSE value for all MMDs contained in $\mathbf{R}^{target}$.

Both approaches were compared on the basis of $\mathbf{MMD}^{test}$. Then, for the clustering supported approach, the MSE limit is defined as the maximal value of all individual MSE limits over $\mathbf{MMD}^{test}$. Figure 11 shows the dependency of the MSE limit on the total number of clusters. For more than 30 clusters, the MSE limit is almost constant at a level of 0.003. The MSE limit increases rapidly for less than 30 clusters.

Table 5. Pareto optimal results with balanced weights for different number of clusters.

Number of clusters		20	40	60
cand. ID/		3	2	1
property	w_i			
$c_{m,0}$ / mol·L^{-1}		3.50	3.07	2.86
$c_{ini,0}$ / mmol·L^{-1}		13.0	20.0	20.0
time / min	0.5	60.0	40.0	20.0
MSE /·10^{-3}		3.04	0.32	0.03
conversion / %	0.5	58.2	49.2	27.0
score (w_t = 0.5, w_{cm} = 0.5)		0.50	0.39	0.50
score (w_{MSE} = 1/3, w_{cm} = 1/3, w_t = 1/3)		0.67	0.30	0.33



doi:10.20944/preprints202402.0199.v1

Figure 11. *MSE* limit calculated for the clustering of the search space.

Figure 12A illustrates the average number of Pareto front points depending on the MSE limit. Increasing the MSE limit results in a reduction in the number of clusters (see Figure 11) associated with an enlargement in cluster size, and thereby leading to a higher count of Pareto front points. The clustering-supported approach is more selective considering the average number of Pareto front points. The standard deviation is used to describe the value ranges of the objective function and of the MSE function. Wider ranges are advantageous, because they provide more space for optimization. Averaging the value ranges of each objective function over **MMD**$^{\text{test}}$ yields the average ranges illustrated in Figure 12B–D. The value ranges of the reaction time and conversion are wider (Figure 12C,D) for the clustering supported approach, while the MSE value range is more restricted (Figure 12B). The findings indicate that the clustering approach allows for a better optimization of monomer conversion and reaction time while the direct approach is better suited for optimization of the MSEs.

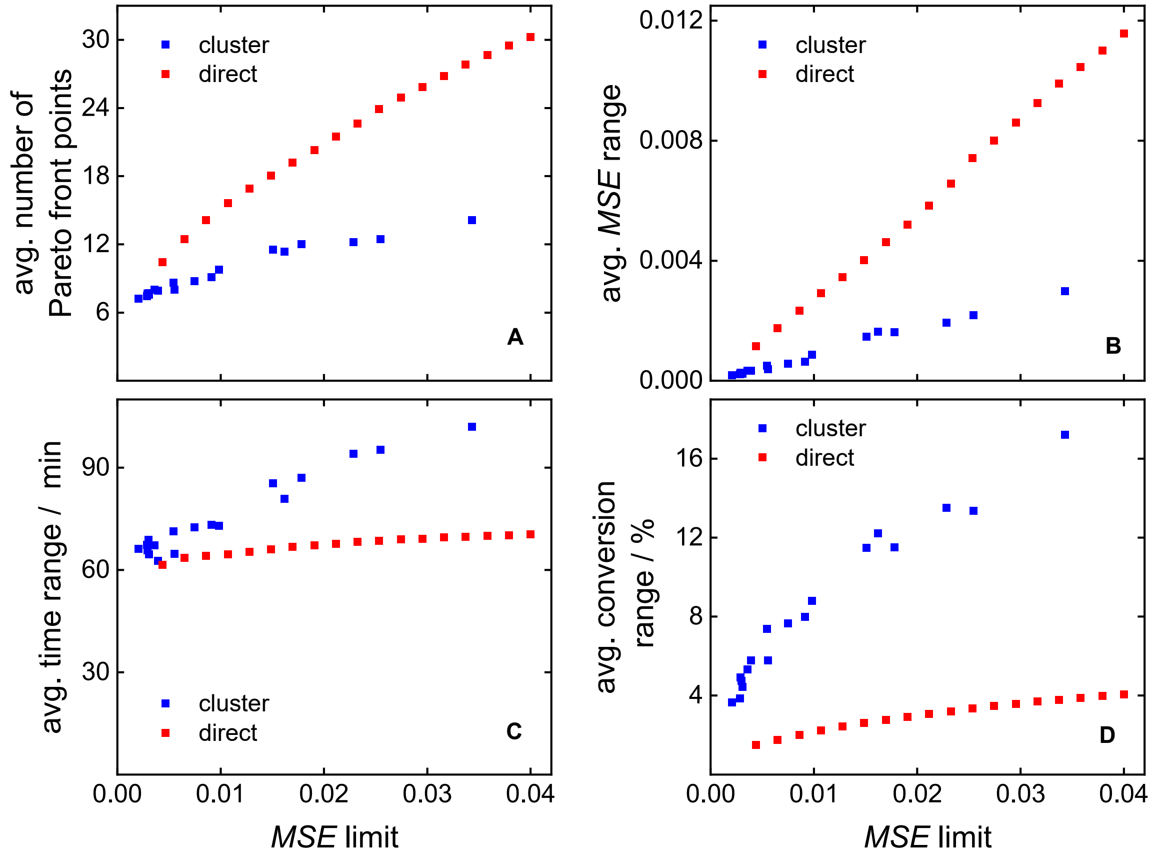


Figure 12. Comparison of the direct and the clustering-supported approach considering (A) the average number of Pareto front points, (B) the average MSE range, (C) the average polymerization time range, and (D) the average conversion range.

Table 6 presents the execution time for both approaches for the test set **MMD**$^{\text{test}}$ consisting of 810 target MMDs. For the clustering-supported approach, the optimization time depends slightly on the number of clusters. However, it is by a factor of ten smaller than for the direct approach.

Table 6. Comparison of optimization time of direct and clustering supported approaches.

approach	direct	clustering supported approach

		20 clusters	40 clusters	60 clusters	80 clusters	100 clusters
execution time, s	679	56	25	18	14	12

The rapidness of the MOO algorithm is especially important, for the next step, when in future genetic algorithms together with ML models will be applied to extend the search space for the polymerization parameters, because Pareto optimization should be executed many times for each evolution stage of genetic algorithms. In this situation, the clustering supported Pareto optimization is preferable.

4. Conclusions and Outlook

The study bridges the gap between multi-objective optimization (MOO) and its application for reverse engineering of polymerization processes. The proposed MOO models allow for simulation-supported determination of an optimal recipe for a targeted molar mass distribution, taking multiple objectives, e.g., minimal reaction time, maximal conversion of monomer, and similarity of the found MMD compared to the target MMD into consideration. The proposed approach can be accelerated by an additional clustering of the simulated MMDs. Moreover, it is possible to obtain a number of suitable candidates considering different weights of the selected objectives. A set of alternative optimal solutions is obtained for each specific combination of weights. First insights are provided on the formulation of a polymerization reverse engineering task as a MOO problem. In future, the understanding gained in combination with previously proposed ML models [11] for polymerizations will be applied to facilitate and to speed-up finding of optimal solutions with a limited number of kMC simulations. ML models can be used to set-up the search space and to evaluate the candidate polymerization procedures rather than using kMC simulations. The search for candidate solutions will be performed with the help of genetic algorithms. Moreover, the consideration of more complex microstructural details, e.g., such as branching in high-temperature polymerization of acrylate [32], may require to increase the number of objectives in MOO.

Data Availability Statement: The data presented in this study are openly available in Mendeley Data at doi:10.17632/hdrwmxgx27.1 with reference: https://data.mendeley.com/preview/hdrwmxgx27?a=d024c5aa-a100-473f-b641-bc0d3f11ebb5.

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

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ballard, N.; Asua, J.M. Radical polymerization of acrylic monomers: An overview. *Prog. Polym. Sci.* **2018**, *79*, 40–60, doi:10.1016/j.progpolymsci.2017.11.002.
2. Meszena, Z.G.; Johnson, A.F. Modelling and simulation of polymerisation processes. *Comput. Chem. Eng.* **1999**, *23*, 375–378, doi:10.1016/S0098-1354(99)80092-1.
3. Saldívar-Guerra, E. Numerical Techniques for the Solution of the Molecular Weight Distribution in Polymerization Mechanisms, State of the Art. *Macromol. React. Eng.* **2020**, *14*, 2000010, doi:10.1002/mren.202000010.
4. Gómez-Reguera, J.A.; Vivaldo-Lima, E.; Gabriel, V.A.; Dubé, M.A. Modeling of the Free Radical Copolymerization Kinetics of n-Butyl Acrylate, Methyl Methacrylate and 2-Ethylhexyl Acrylate Using PREDICI®. *Processes* **2019**, *7*, 395, doi:10.3390/pr7070395.

5. Trigilio, A.D.; Marien, Y.W.; Van Steenberge, P.H.M.; D'hooge, D.R. Gillespie-Driven kinetic Monte Carlo Algorithms to Model Events for Bulk or Solution (Bio)Chemical Systems Containing Elemental and Distributed Species. *Ind. Eng. Chem. Res.* **2020**, *59*, 18357–18386, doi:10.1021/acs.iecr.0c03888.
6. Peikert, P.; Pflug, K.M.; Busch, M. Modeling of High-Pressure Ethene Homo- and Copolymerization. *Chem. Ing. Tech.* **2019**, *91*, 673–677, doi:10.1002/cite.201800206.
7. Trigilio, A.D.; Marien, Y.W.; Van Steenberge, P.H.M.; D'hooge, D.R. Toward an Automated Convergence Tool for Kinetic Monte Carlo Simulation of Conversion, Distributions, and Their Averages in Non-dispersed Phase Linear Chain-Growth Polymerization. *Ind. Eng. Chem. Res.* **2023**, *62*, 2583–2593, doi:10.1021/acs.iecr.2c03979.
8. Brandão, A.L.T.; Soares, J.B.P.; Pinto, J.C.; Alberton, A.L. When Polymer Reaction Engineers Play Dice: Applications of Monte Carlo Models in PRE. *Macromol. React. Eng.* **2015**, *9*, 141–185, doi:10.1002/mren.201500020.
9. Mohammadi, Y.; Saeb, M.R.; Penlidis, A.; Jabbari, E.; J Stadler, F.; Zinck, P.; Matyjaszewski, K. Intelligent Machine Learning: Tailor-Making Macromolecules. *Polymers* **2019**, *11*, 579, doi:10.3390/polym11040579.
10. Mohammadi, Y.; Penlidis, A. Polymerization Data Mining: A Perspective. *Adv. Theory Simul.* **2019**, *2*, 1800144, doi:10.1002/adts.201800144.
11. Fiosina, J.; Sievers, P.; Drache, M.; Beuermann, S. Polymer reaction engineering meets explainable machine learning. *Comput. Chem. Eng.* **2023**, *177*, 108356, doi:10.1016/j.compchemeng.2023.108356.
12. Spyromitros-Xioufis, E.; Tsoumakas, G.; Groves, W.; Vlahavas, I. Multi-target regression via input space expansion: treating targets as inputs. *Mach. Learn.* **2016**, *104*, 55–98, doi:10.1007/s10994-016-5546-z.
13. Ngatchou, P.; Zarei, A.; El-Sharkawi, A. Pareto Multi Objective Optimization. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, 2005.* 13th International Conference on, Intelligent Systems Application to Power Systems, Arlington, Virginia, USA, 06–10 Nov. 2005; IEEE / Institute of Electrical and Electronics Engineers Incorporated, 2005; pp 84–91, ISBN 1-59975-174-7.
14. Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Eng.* **2018**, *5*, 1502242, doi:10.1080/23311916.2018.1502242.
15. Doumpos, M.; Zopounidis, C. Multi-objective optimization models in finance and investments. *J. Glob. Optim.* **2020**, *76*, 243–244, doi:10.1007/s10898-019-00873-z.
16. Gupta, S.; Haq, A.; Ali, I.; Sarkar, B. Significance of multi-objective optimization in logistics problem for multi-product supply chain network under the intuitionistic fuzzy environment. *Complex Intell. Syst.* **2021**, *7*, 2119–2139, doi:10.1007/s40747-021-00326-9.
17. Bhaskar, V.; Gupta, S.K.; Ray, A.K. APPLICATIONS OF MULTIOBJECTIVE OPTIMIZATION IN CHEMICAL ENGINEERING. *Rev. Chem. Eng.* **2000**, *16*, 1–54, doi:10.1515/REVCE.2000.16.1.1.
18. Murugan, C.; Subbaian, S. Multi-Objective Optimization for Enhanced Ethanol Production during Whey Fermentation. In *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS),* Chennai, India, 08–09 Dec. 2022; IEEE, 2022; pp 1–7, ISBN 978-1-6654-6275-4.
19. Fiandaca, G.; Fraga, E.S.; Brandani, S. A multi-objective genetic algorithm for the design of pressure swing adsorption. *Eng. Optim.* **2009**, *41*, 833–854, doi:10.1080/03052150903074189.
20. Ganesan, T.; Elamvazuthi, I.; Ku Shaari, K.Z.; Vasant, P. Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. *Appl. Energy* **2013**, *103*, 368–374, doi:10.1016/j.apenergy.2012.09.059.
21. Charoenpanich, T.; Anantawaraskul, S.; Soares, J.B.P. Using Artificial Intelligence Techniques to Design Ethylene/1-Olefin Copolymers. *Macromol. Theory Simul.* **2020**, *29*, 2000048, doi:10.1002/mats.202000048.
22. Dragoi, E.N.; Curteanu, S. The use of differential evolution algorithm for solving chemical engineering problems. *Rev. Chem. Eng.* **2016**, *32*, 149–180, doi:10.1515/revce-2015-0042.
23. Fernandes, F.A.N.; Lona, L.M. Neural network applications in polymerization processes. *Braz. J. Chem. Eng.* **2005**, *22*, 401–418, doi:10.1590/S0104-66322005000300009.
24. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007, doi:10.1016/j.ress.2005.11.018.
25. Marler, R.T.; Arora, J.S. The weighted sum method for multi-objective optimization: new insights. *Struct. Multidisc. Optim.* **2010**, *41*, 853–862, doi:10.1007/s00158-009-0460-7.
26. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *Appl. Stat.* **1979**, *28*, 100–108, doi:10.2307/2346830.

27. Nielsen, F.; Nock, R.; Amari, S. On Clustering Histograms with k-Means by Using Mixed α-Divergences. *Entropy* **2014**, *16*, 3273–3301, doi:10.3390/e16063273.
28. Henderson, K.; Gallagher, B.; Eliassi-Rad, T. EP-MEANS: An Efficient Nonparametric Clustering of Empirical Probability Distributions. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. SAC 2015: Symposium on Applied Computing, Salamanca Spain, 13–17 Apr. 2015; Wainwright, R.L., Corchado, J.M., Bechini, A., Hong, J., Eds.; ACM: New York, NY, USA, 2015; pp 893–900, ISBN 9781450331968.
29. Banerjee, A.; Merugu, S.; Dhillon, I.S.; Ghosh, J. Clustering with Bregman Divergences. *J. Mach. Learn. Res.* **2005**, *6*, 1705–1749.
30. Drache, M.; Drache, G. Simulating Controlled Radical Polymerizations with mcPolymer—A Monte Carlo Approach. *Polymers* **2012**, *4*, 1416–1442, doi:10.3390/polym4031416.
31. Feuerpfeil, A.; Drache, M.; Jantke, L.-A.; Melchin, T.; Rodríguez-Fernández, J.; Beuermann, S. Modeling Semi-Batch Vinyl Acetate Polymerization Processes. *Ind. Eng. Chem. Res.* **2021**, *60*, 18256–18267, doi:10.1021/acs.iecr.1c03114.
32. Mätzig, J.; Drache, M.; Drache, G.; Beuermann, S. Kinetic Monte Carlo Simulations as a Tool for Unraveling the Impact of Solvent and Temperature on Polymer Topology for Self-Initiated Butyl Acrylate Radical Polymerizations at High Temperatures. *Macromol. Theory Simul.* **2023**, *32*, 2300007, doi:10.1002/mats.202300007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.