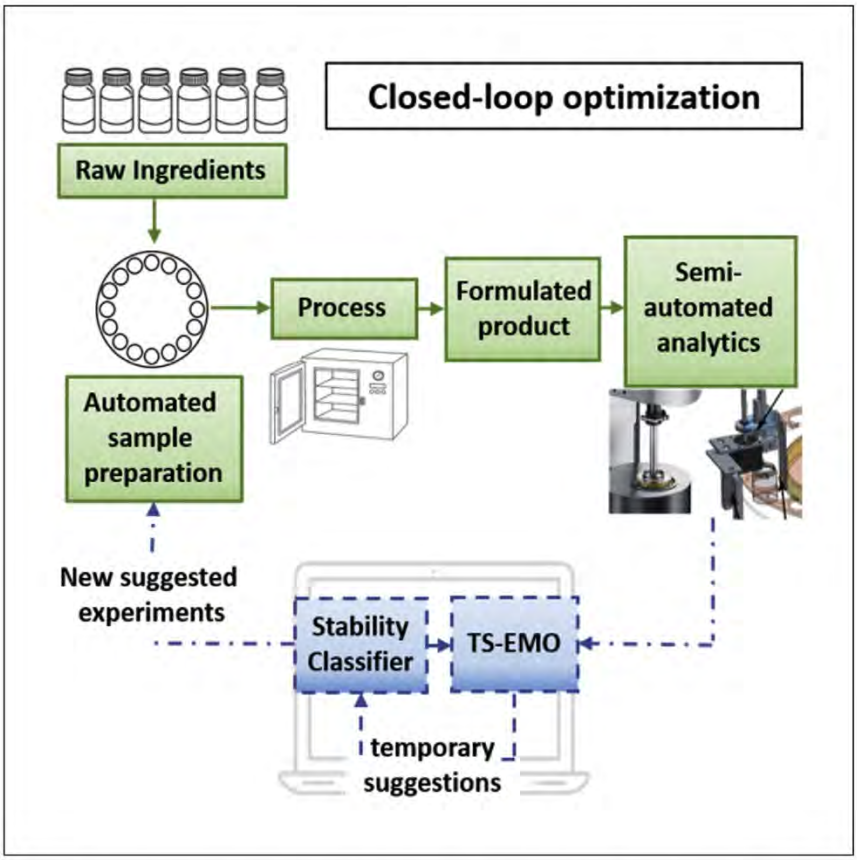
Cell Reports
Physical Science

CellPress
OPEN ACCESS

Article

# Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE




Robotic experiments and machine learning are a promising paradigm for fast development of complex formulated products. Cao et al. report the integration of a built-in robotic platform and Bayesian algorithms for the fast optimization of both continuous and discrete objectives. The proposed methodology may speed time to market.

Liwei Cao, Danilo Russo, Kobi Felton, ..., Huanhuan Gao, Leroy Cronin, Alexei A. Lapkin

aal35@cam.ac.uk

## HIGHLIGHTS

A commercial formulation is optimized using a Bayesian approach

Discrete and continuous targets are optimized without any available physical model

A robotic platform can perform semiautomated sample preparation and characterization

The adopted iterative approach gives satisfactory results within 15 working days

Cao et al., Cell Reports Physical Science 2, 100295
January 20, 2021 © 2020 The Authors.
https://doi.org/10.1016/j.xcrp.2020.100295

Please cite this article in press as: Cao et al., Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE, Cell Reports Physical Science (2020), https://doi.org/10.1016/j.xcrp.2020.100295

Cell Reports
Physical Science

CellPress
OPEN ACCESS

Article

# Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE

Liwei Cao,[1,2] Danilo Russo,[1] Kobi Felton,[1] Daniel Salley,[3] Abhishek Sharma,[3] Graham Keenan,[3] Werner Mauer,[4] Huanhuan Gao,[5] Leroy Cronin,[3] and Alexei A. Lapkin[1,2,6,*]

## SUMMARY

**Formulated products are complex mixtures of ingredients whose time to market can be difficult to speed due to the lack of general predictable physical models for the desired properties. Here, we report the coupling of a machine learning classification algorithm with the Thompson sampling efficient multiobjective optimization (TSEMO) algorithm for the simultaneous optimization of continuous and discrete outputs. The methodology is successfully applied to the design of a formulated liquid product of commercial interest for which no physical models are available. Experiments are carried out in a semiautomated fashion using robotic platforms triggered by the machine learning algorithms. The procedure allows one to find nine suitable recipes meeting the customer-defined criteria within 15 working days, outperforming human intuition in the target performance of the formulations.**

## INTRODUCTION

Liquid formulated products have a large number of applications in chemical, energy, and environmental industries and include a wide range of products, such as pharmaceuticals, food, fuels, cosmetics and personal care products, and polymers.[1–3] They consist of mixtures of ingredients and are processed to meet specific target properties.[4] Despite the fact that some ingredients might be completely miscible and form a homogeneous phase, many formulated products of industrial interest are, in fact, emulsions of immiscible phases, stabilized with additional ingredients, such as ionic surfactants,[2,5,6] and using specific process conditions. As a result, one of the main objectives in liquid formulated product design is to find a proper composition to obtain a stable emulsion, with no phase separation, meeting specific functionalities or customer-defined properties, such as a certain viscosity, color, or fragrance.[7]

Unfortunately, such a product can be very complex from a physicochemical point of view, and often, no direct correlations are available to predict the target properties given the composition of the formulation. As a consequence, the problem of the design of formulated products is of great interest to both academia and industry, and most relevant publications manifested the need for a general framework for their development.[8]

Pioneering work in this area has been carried out by the group of Prof. Rafiqul Gani.[4,9–13] According to these sources, the general approaches to formulated product design consist of three main strategies: (1) a trial-and-error approach, based on the expertise of the operators and completely dependent on experimental data (this approach is highly resource and time demanding but in most cases is still widely

[1]Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK

[2]Cambridge Centre for Advanced Research and Education in Singapore, CARES Ltd. 1 CREATE Way, CREATE Tower #05-05, 138602 Singapore, Singapore

[3]WestCHEM. School of Chemistry, University of Glasgow, Glasgow G12 8QQ, UK

[4]BASF Personal Care and Nutrition GmbH, 40589 Duesseldorf-Holthausen, Germany

[5]BASF Advanced Chemical Co., Ltd., No. 300 Jiangxinsha Road, 200137 Shanghai, China

[6]Lead Contact

*Correspondence: aal35@cam.ac.uk

https://doi.org/10.1016/j.xcrp.2020.100295


This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

adopted); (2) a model-based approach that consists of the generation of a few options to be experimentally tested using available models to predict the target properties (in this case, the resources employed are minimal, but accurate and reliable physical models are needed); and (3) integrated approaches combining model predictions and experimental data. Starting from the general framework proposed by Cussler and Moggridge,[14] Conte et al.[4] reported an integrated multistage approach. First, the main performance criteria and the properties determining them are taken into account, based on *a priori* knowledge, setting constraints and restricting the search space. Later on, available models can be used to design formulations with a specific target function, such as a fixed viscosity and stability, and the tradeoff between different candidates is evaluated. Experimental data might be used at different stages in the framework to verify the validity of the prediction or reject some of the candidates. This approach still relies on the availability of predictive models.

Unfortunately, for complex commercial products with a large number of ingredients, multiple empirical or semi-empirical models might be available, and in some cases, none of them can successfully predict the experimental outcomes. As an example, predicting viscosity of multicomponent emulsions can be a rather challenging task when little information is available about the complex interactions of surfactants, especially for high concentrations of active ingredients. An overview of the different models to predict viscosity of formulated products can be found elsewhere,[3,15,16] highlighting the limitations of each model and the strong assumptions that often are invalid for industrially relevant formulations.

An alternative methodology for developing formulations is to employ an intelligent design of experiments (DoE)[17] methodology that would be capable of learning key features of a particular formulation during an experimental campaign. Statistical DoE methods, such as two-level full factorials,[18] fractional factorials,[19] Plackett-Burman method,[20] mixed-level fractional factorial design,[21] D-optimal design,[22] and others,[23] were applied in the field of formulated products development. The DoE methods are typically used to determine a statistical surrogate model by generating samples that cover the design space,[24] but these are usually applied to relatively simple models, focusing on main effects or a main effect plus identification of interactions with few factors.[25] Such DoE methods are inefficient when a complex model (in the case of a large number of ingredients and complex interactions) is required to achieve predefined design objectives.[26]

More recently, machine learning algorithms have emerged as a potential solution for challenging tasks, such as the discovery of new chemical reactivity and prediction of reaction outcomes.[23] Among these methods, Bayesian optimization active learning provides an adaptive paradigm to sample the design space more efficiently for identifying the optimum. For example, Zhang et al. presented a Bayesian optimization framework for design of materials, where a latent variable approach was integrated with Gaussian process (GP) modeling in order to support a variety of materials design applications with mixed qualitative and quantitative design variables.[27] This method was efficient and effective in optimizing material constituents of a hybrid organic-inorganic perovskite.

Another common problem in formulated products design is finding tradeoffs between different conflicting objectives that can be a combination of continuous and discrete variables. To solve this problem, the combination of surrogate models and multiobjective optimization has recently presented an exciting opportunity to

maximize the amount of useful information gained per experiment. Fitzpatrick et al. applied scalarization approach, where multiple objectives are combined into a single objective function with different weightings, in order to simultaneously optimize throughput, conversion, and consumption for a model reaction.[28] A scalarization approach is very useful and computationally efficient, but it is frequently difficult to objectively define suitable weights, especially without sufficient *a priori* knowledge, not to mention that even a minor change to the weightings can result in significant changes to the obtained solution.[29] On the other hand, evolutionary algorithms, such as non-dominated-sort genetic algorithms, are designed to converge on the Pareto front using a Pareto dominance ranking system.[30] By coupling GPs with genetic algorithm, several multiobjective optimization schemes were proposed and applied in chemical reaction and process design.

The multiobjective active learner (MOAL)[31] was proposed for expensive to evaluate multiobjective optimization tasks and successfully applied to discovery of emulsion polymerization recipes with 14 input variables and two desired objectives (namely full monomer conversion and a specific mean particle diameter).[32] Similar machine learning methodologies, such as Pareto efficient global optimization (ParEGO)[33] and expected hypervolume improvement (EHI),[34] were also proposed in the literature. In a previous study, some of the authors developed the Thompson sampling efficient multiobjective optimization (TSEMO) algorithm, which randomly samples from the GPs and uses the NSGA-II (Non-dominated Sorting Genetic Algorithm II) algorithm to identify the Pareto front of each random sample.[35] Although a number of recent studies explored the topic optimization of chemical systems in the presence of continuous and discrete input variables,[36–39] there are no examples of multiobjective optimization of both discrete and continuous outputs.

With the recent advances in laboratory automation, a large amount of highly reproducible data can be available to train machine learning models to correlate multiple target outputs to complex high-dimensional inputs.[40,41] Such models can be integrated in algorithms for the DoE that are capable of suggesting suitable conditions to optimize the process with respect to the targets of interest and combined with robotic platforms for the generation and analyses of experimental samples.[42,43]

In this paper, we present such closed-loop optimization system for the multiobjective optimization of a commercial formulated product. The interaction between a classification algorithm and an efficient multiobjective optimization algorithm for continuous variables enables one to simultaneously meet discrete (namely, formulation stability) and continuous (namely, viscosity, turbidity, and price) targets. The proposed methodology enables one to find suitable solutions within a relatively short time period (i.e., 15 working days) using little empirical prior knowledge about the physical system to define the constraints of the input variables. This makes the proposed pipeline particularly suitable for the early stages of the formulated products design.

## RESULTS AND DISCUSSION

### Classification Algorithm for Stability Prediction

As explained in detail in Experimental Procedures, optimization of the formulated product under investigation was carried out with respect to different conflicting objectives, both discrete, i.e., stability, and continuous, i.e., turbidity, viscosity, and price. A classification algorithm was developed and trained in order to classify the suggestions of the coupled TSEMO algorithm based on their stability and avoid

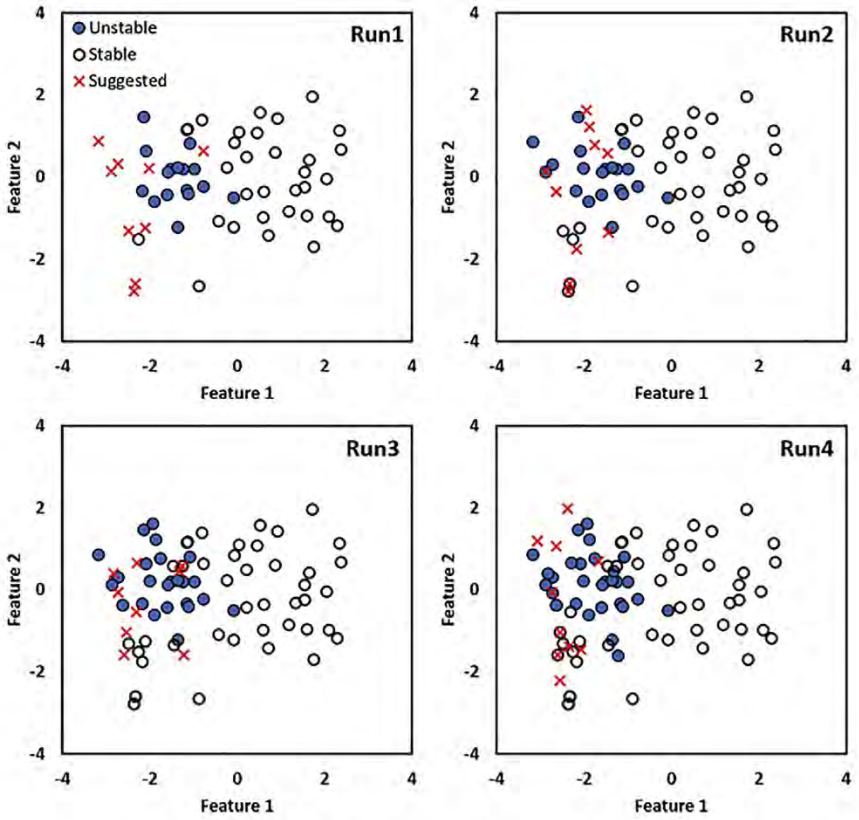



**Figure 1. Active Learning Iterations of the stability classifier in the Bi-dimensional LDA Reduced Space**

Four consecutive iterations of the active learning classifier show the efficiency of the algorithm in suggesting new conditions (red crosses) at the boundary between the stable and unstable domains. The repulsion criterium described in Experimental Procedures results in suggestions that are not too similar to each other for efficient exploration. Given the 5-dimensional space, LDA was used as a dimensionality reduction technique for visualization of the data. A direct comparison with the suggestions using random sampling is provided in Figure S6.

waste of resources and time, generating samples that exhibit the undesired phase separation. The smart sampling classifier was developed as described in Experimental Procedures. An initial dataset of 48 samples was generated using a Latin hypercube design as a space-filling technique.[44] The samples were generated by two complete rounds of the robotic platforms R1 and R2. The initial dataset was used to train the classifier and generate 12 new potential experiments to be performed at each iteration. The new generated samples were added to the dataset and the procedure repeated for four complete cycles. The initial number of 48 experiments was chosen according to the full capacity of the robotic platform and the previously reported 10×d rule of thumb, with d representing the number of input variables under consideration.[45]

As described in detail in Experimental Procedures, a naive Bayes classifier was chosen due to its simplicity, efficiency, and accuracy in classification problems. The performance was evaluated as the average for prediction accuracies of the 5-fold cross-validation between the classes of stable and unstable formulations.[46]

A bi-dimensional representation of the initial dataset is shown in Figure 1 (run 1), together with the first 12 suggested experiments. Linear discriminant analysis

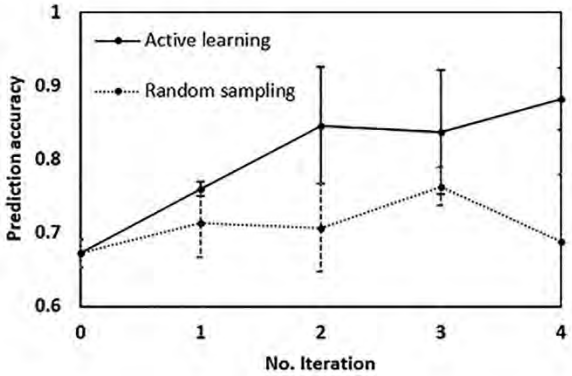



**Figure 2. Prediction Accuracy at Different Iterations Using Active Learning and Random Sampling**
The prediction accuracy of the Bayesian classifier trained using active learning is superior to the one observed using random sampling. The so-trained classifier is embedded in the general algorithmic framework as described in Experimental Procedures. Prediction accuracy has been calculated as the mean value (i.e., the average correct rates of the five times 5-fold cross-validation and corresponding standard deviation results from the SVM classifier).

(LDA) was found to be a suitable dimensionality reduction algorithm for better visualization of the dataset. The results of the iterative training of the classifier are shown in Figure 1. As one can see, the suggested experiments at each iteration are nicely distributed at the border between the two clusters, which represents the area with the highest uncertainties. Moreover, the repulsive criterion adopted for the batch sequential design provided a better exploration of the space. For the sake of comparison, the same procedure was repeated, adopting a random sampling strategy. For the sake of comparison, full representation of the suggested experiments using a random sampling is reported in Table S2.

The prediction accuracy was evaluated using a support vector machine (SVM) classifier. In fact, as the naive Bayes model was used in the active learning algorithm method, we might have collected data solely tailored to the model built by the naive Bayes classifier. Figure 2. shows the prediction accuracy of the active learning algorithm and the random sampling, proving the superiority of the former, at each iteration.

## Optimization Results

The same 96 data points collected to train the classification algorithm described in the previous section were used to initiate the TSEMO algorithm. Once initiated, 16 iterations of the optimization procedure were carried out, generating a total of 128 samples within 15 working days. As described in Experimental Procedures, the algorithm suggests conditions in order to find the best predictions of the actual Pareto front, minimizing uncertainties and finding a compromise between the minimization of the conflicting objective functions. The target properties for the specific product under considerations were (1) stability and low turbidity, (2) honey-like viscosity (target 3 Pa×s at a shear rate of 10 $s^{-1}$ and 25°C), and (3) low price of the adopted ingredients. Interestingly, the percentage of suggested unstable formulations presenting phase separation significantly decreased over the iterative optimization loop, and the algorithm stopped suggesting unstable conditions from the $12^{th}$ iteration, as shown in Figure 3. This can be ascribed to the fact that unstable samples often present a higher value of turbidity, which is one objective chosen for the optimization procedure. However, the integration of the stability classifier helped to

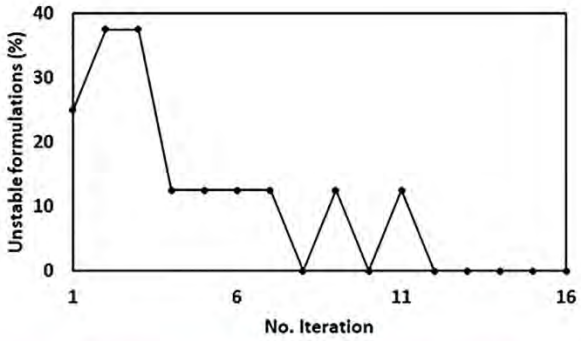



**Figure 3. Percentage of Suggested Unstable Formulations at Each Iteration**

The percentage of suggested unstable formulation was recorded and reported as a function of the iteration number. The number significantly decreases with time, suggesting that the algorithm is searching the space, looking for stable formulations characterized by a low turbidity. No unstable formulations were suggested after the 12$^{th}$ iteration.

avoid running experiments giving unstable products, saving time and reducing the waste of resources.

The experimentally collected data were automatically analyzed to provide a full list of the non-dominated experimental solutions, which represent the experimental Pareto front of the dataset. Non-dominated solutions were defined as the ones where an improvement in one objective would lead to a worsening in at least one other objective. The full list of 32 non-dominated solutions was identified, 11 of which were in the training dataset and 19 of which were in the suggested experiments (Table S3). In Table 1, we reported only non-dominated solutions meeting the viscosity criterion. The full dataset is reported in Tables S1, S2, and S4.

As one can see, although a good number of clear formulations were already present in the training set, the algorithm was able to explore the input space more efficiently, finding alternative solutions with a significant reduction in the price. The obtained solution was compared with the best solution provided in a dataset guided by experts' intuition. In a previous independent experimental campaign, formulation scientists from BASF used factorial design[47,48] for the initial screening of the design space and based on the results designed the final batch of experiments for testing and optimization (280 experiments in total). In this case, the closest solution to the target was found using the following recipe: S1 = 4.00 g/L, S2 = 5.00 g/L, S3 = 6.00 g/L, P1 = 2.00 g/L, T1 = 2.00 g/L. In this case, a homogeneous formulation with a viscosity of 9,270 mPa×s was obtained, with a turbidity value slightly higher than 200 nephelometric turbidity units (NTU) and a cost of 2.19 $/L, proving that

**Table 1. Non-dominated Solutions Passing the Viscosity Criterium**

| | S1 (g/L) | S2 (g/L) | S3 (g/L) | P1 (g/L) | T1 (g/L) | Turbidity (NTU) | Viscosity (mPa×s) | Price ($/L) |
|---|---|---|---|---|---|---|---|---|
| Training data | 3.97 | 3.16 | 7.87 | 0.84 | 1.60 | 11.2 | 2,678 | 2.00 |
| | 4.61 | 2.62 | 7.76 | 1.24 | 1.04 | 4.4 | 3,574 | 2.06 |
| | 8.40 | 3.18 | 3.43 | 0.48 | 1.80 | 17.2 | 2,948 | 2.43 |
| | 2.79 | 2.93 | 9.28 | 1.72 | 0.88 | 7.4 | 2,633 | 1.92 |
| | 6.16 | 4.29 | 4.55 | 0.52 | 1.77 | 56.1 | 2,889 | 2.20 |
| | 10.94 | 3.94 | 0.12 | 1.80 | 1.36 | 28.0 | 2,992 | 2.80 |
| Suggested data | 4.02 | 3.09 | 0.99 | 0.99 | 1.39 | 15.1 | 2,824 | 2.00 |
| | 3.01 | 0.64 | 11.35 | 0.53 | 1.75 | 49.6 | 2,789 | 1.87 |
| | 3.06 | 1.99 | 9.95 | 0.31 | 1.42 | 23.7 | 3,301 | 1.82 |

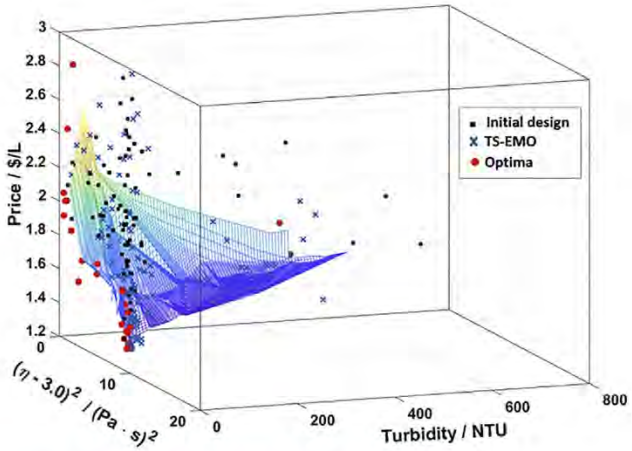



**Figure 4. Dataset and Experimental and Predicted Pareto Front**

The non-dominated experimental solutions (red dots) are reported, together with all the experiments carried out during the iterative optimization (blue crosses) and the initial design. The represented surface is the calculated Pareto front according to the predictions of the surrogate GP models used in the algorithm. The three objectives considered are price, turbidity, and squared distance between the actual viscosity and the target. The figure shows a good agreement between the prediction of the models and the experimentally found best solutions.

an appropriate space-filling technique coupled with an algorithmic search can significantly outperform human intuition in a relatively short amount of time with very little prior knowledge about the physical system.

In order to evaluate the predictive capability of the trained surrogate models, the predicted Pareto front was plotted together with the experimental optima. In Figure 4, we report the predicted non-dominated solutions surface, the data used for the initial training of TSEMO, all suggested experiments, and the non-dominated experimental optima. As shown, the predicted Pareto front gives a good approximation of the actual best solutions, laying in the neighborhood of the calculated surface. *A posteriori* analyses of the shape of the Pareto front can also give some physical information to human operators to provide a better insight about the system. From the data reported in Figure 4, it is clear that a large number of solutions close to the Pareto front have a low value of turbidity. Also, as a general trend, as viscosity approaches the target value, the price and turbidity of the system tend to increase. This would suggest that, on average, the most expensive ingredients (S2, T1, and P1) would be the ones responsible for an increase in viscosity and turbidity of the samples, which was found to be the case for most of the samples in the dataset. Of course, these are only general semiquantitative indications, which do not reveal any information about more complex interactions that might occur between the different ingredients at different concentrations. However, it is worth noting that the presented methodology can also offer some guidelines to experts for further improvements and considerations about the actual physical role of the input variables on the desired properties of the product. Different angles of view of Figure 4 are provided in Figure S7.

In this regard, the values of the hyperparameters of the trained GPs can also provide information about the relevance of the input variables for each objective function.[49] For the adopted surrogate models, a lower value of the hyperparameter of an input variable indicates a greater contribution to the objective. The values of the hyperparameters are reported in Table 2.

**Table 2. Hyperparameters of GP Models**

| | GP1 (Viscosity) | GP2 (Turbidity) |
|---|---|---|
| S1 | $8.57 \times 10^{-2}$ | $5.9 \times 10^{-2}$ |
| S2 | $3.16 \times 10^{1}$ | $1.62 \times 10^{-1}$ |
| P1 | $9.42 \times 10^{0}$ | $7.85 \times 10^{-1}$ |
| T1 | $4.00 \times 10^{-2}$ | $7.47 \times 10^{-2}$ |

The analysis of the hyperparameters suggests again a stronger influence of T1 on the viscosity and turbidity; more complex interactions between S1 and T1 seem to be responsible for variations in the viscosity value, whereas S2 seems to also have a relevant effect on the turbidity of the samples. This kind of qualitative information may lay the foundation for integrated approaches for the simultaneous black-box optimization and physical knowledge generation by using robotic platform, in combination with other recently published promising methodologies.[50–52]

In conclusion, time-saving and highly reproducible robotic experiments were coupled with machine learning algorithms for the efficient optimization of the recipe of a complex formulated product of industrial interest. The optimization procedure outperformed human experts' intuition and suggested more convenient and low-priced solutions within 15 working days. The coupling of a naive Bayes classifier with the TSEMO algorithm allowed us to take into account in the optimization procedure binary discrete outputs while also avoiding wasting time and material resources. Despite the fact that the optimization was carried out in the absence of predictive physical models, *a posteriori* analysis of the Pareto front and analysis of hyperparameters of the surrogate statistical model gave some important qualitative information about the physics of the system. Future research will focus on the connection between data-driven statistical models and the generation of physical models, which would give an insight into the physics of the system and lead to an optimal design of similar types of products. Moreover, future efforts will address the problem of optimization under unknown constraints and its integration in formulated product design to minimize the need for prior knowledge about the system.

## EXPERIMENTAL PROCEDURES

### Resource Availability

#### *Lead Contact*

The lead contact is A.L. (aal35@cam.ac.uk)

#### *Materials Availability*

This study did not generate new unique reagents. The described surfactants, polymer, and thickener were provided by BASF.

#### *Data and Code Availability*

The file repository used for this work can be found at the following GitHub page: https://github.com/sustainable-processes/centripeta. All data are provided in Tables S1, S2, and S4.

### Case Study and Materials

The case study under consideration is a commercial formulation consisting of three different surfactants (S1, Texapon SB3; S2, Dehyton AB30; and S3, Plantacare 818), a polymer (P1, Dehyquart CC7), and a thickener (T1, Arlyon TT). The pH was adjusted using citric acid (ACS reagent, $\geq$99.5%) from Sigma-Aldrich, used as received. Turbidity standards (1, 2, 5, 10, 100, 500, and 1,000 NTU) were purchased from Sigma-Aldrich.

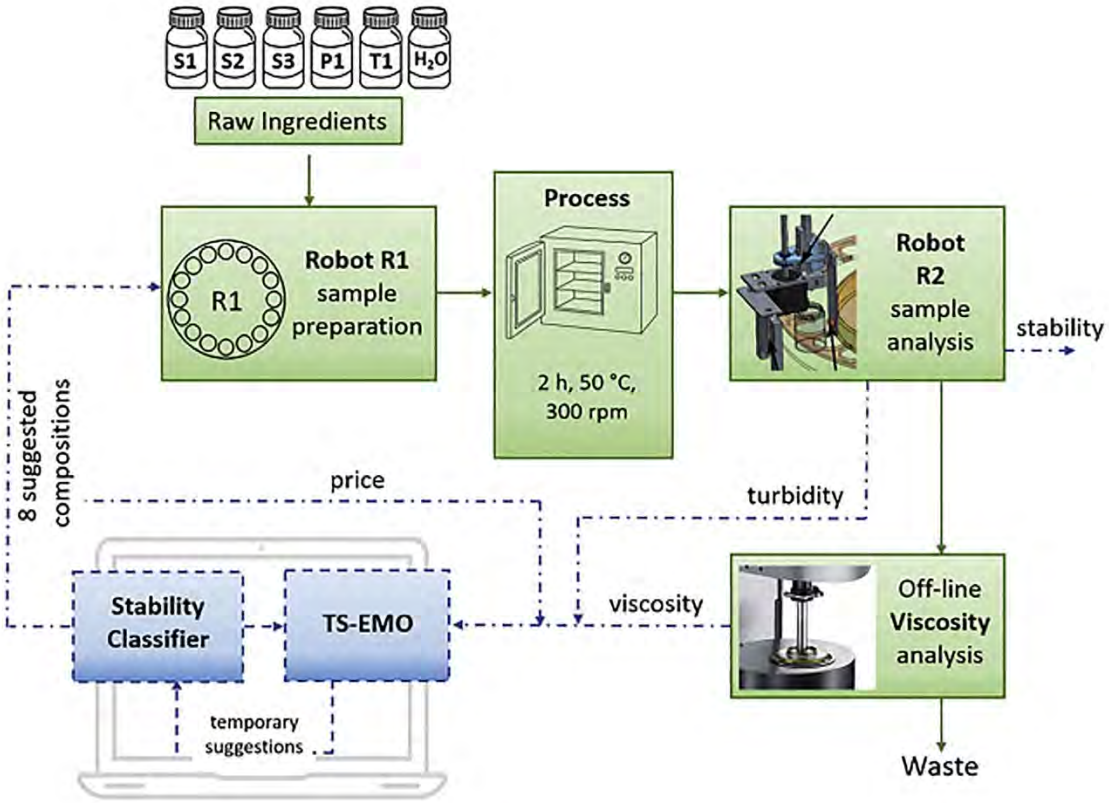



**Figure 5. Scheme of the Adopted Closed-Loop Optimization Workflow**

Material flow (continuous lines) and information flow (dashed lines) are reported. Ingredients are mixed following the suggested recipes in robot R1, processed, and analyzed with a combination of in-line automated operations and offline manual analyses. The results are then collected and processed by the algorithm to suggest a new set of experiments to run for the next iteration.

## Optimization Procedure

A general scheme for the optimization procedure is given in Figure 5. In Figure 5, continuous lines represent the materials flow, whereas dashed lines represent the information flow. The formulation was simultaneously optimized with respect to viscosity, turbidity, stability, and price. At each iteration, a batch of eight different suggested samples is prepared using the robot R1. 15% of the batches were randomly run in triplicate to ensure repeatability. The so-prepared samples are then processed to generate the final product. The samples are successively transferred to the robot R2, which can automatically perform pH, turbidity, and stability tests. The samples are finally analyzed offline to measure viscosity. Details on the robotic platforms R1 and R2 and the experiments are provided in the next section, Supplemental Experimental Procedures, and Figures S1–S5. The turbidity and viscosity values are used to train surrogate GP models for prediction of the target outputs. The price is analytically calculated using the unitary price of each ingredient and their relative amounts, as follows:

$$\frac{\sum_i u_i \cdot v_i}{\sum_i v_i}$$

where $u_i$ is the unitary price ($/L) of the ingredients and $v_i$ (L) the volumes of each adopted ingredient $i$ in the sample.

Based on the predictions, the Bayesian TSEMO algorithm generates eight new conditions to be tested in order to find a compromise between exploitation (finding the

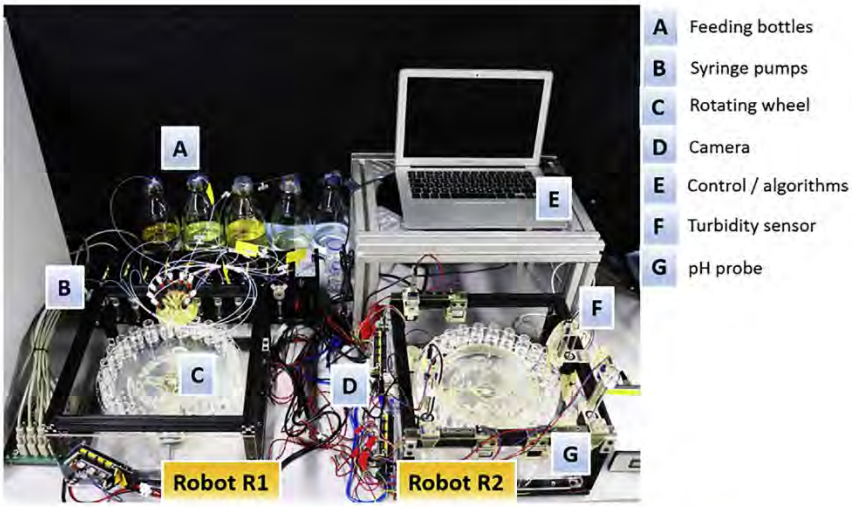



**Figure 6. Image of the Experimental Setup based on the Two Formulation Robots**

The picture shows the actual experimental setup as used for the experiments. Automated syringe pumps (B) are connected to feeding bottles (A) to dispense ingredients to different vials located on the rotating wheel (C) of robot R1. Samples are then moved to the offline incubator for processing and placed in robot R2, where image collection (D), turbidity (F), and pH (G) analyses can be run. The platforms are controlled by the PC (E), where data are stored and fed to the algorithm for the generation of the next iteration.

best conditions to minimize the objectives) and exploration (reducing the uncertainties) of the input chemical space. The generated temporary suggestions are then tested *in silico* using a classification algorithm to predict which samples would be stable. The conditions that give unstable formulations according to the classification algorithm are discarded, and the TSEMO algorithm is reused to generate other suggestions, until an entire batch of eight stable conditions is available. The new suggested conditions are finally added to the dataset and used to trigger a new iteration. Details about the TSEMO algorithm and classification algorithm are provided in the following sections.

## Robotic Experiments

Samples preparation and analyses were partially automated by using two robotic platforms adapted from Salley et al.[53] An image of the platforms R1 and R2 is given in Figure 6. Schematic representation of the different parts and detailed descriptions can be found in Supplemental Experimental Procedures.

Briefly, both platforms consist of a laser-cut rotating wheel that can allocate up to 24 sample vials per batch. In R1, a 3D-printed element, equipped with a variable number of needles, is connected to automated syringe pumps (Tricontinent, Gardner Denver, C-Series), dispensing the five different ingredients. The three surfactants are pre-diluted in water to achieve a concentration of active matter of 20 g/L in the feeding bottles. pH was pre-adjusted to the desired value of 5.5 using citric acid. The requirement for the pH value was fixed for the specific application of the product under consideration. The polymer (P1) and thickener (T1) were used as received. The system was automated and triggered by a Python script to generate eight samples at each iteration of the optimization procedure. Samples were always prepared according to the following constraints for the concentrations, defined using some semi-empirical prior knowledge about the system: $S1 + S2 + S3 = 15\ g_{active\ matter} \times L^{-1}$; $P1 \leq 2$ g/L; and $T1 \leq 2$ g/L. The generated samples were then transferred to an incubator (Corning LSE 71L Shaking Incubator), where they were mixed for 2 h at 50°C and 300 rpm. The obtained formulations were

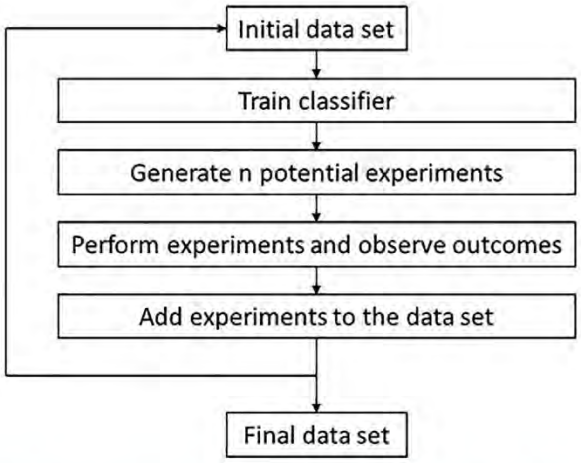



**Figure 7. A Workflow for Active Learning of the Developed Stability Classifier**
The initial dataset of 48 experiments was collected using a Latin hypercube sampling technique. This was used to train the classifier and generate new potential experiments to improve the prediction accuracy of the model. The new results are then added to the initial dataset and the iterative procedure repeated until the algorithm gives satisfactory predictions.

cooled down to room temperature before being transferred to R2. In this second robotic platform, we perform three kinds of analyses in an automated fashion. A pH probe confirms that no pH changes occur during the process (VWR pH electrode, Semi-micro, Pellon junction 662-1767). No significant deviation from the target pH of 5.5 was recorded at any time. A built-in turbidity sensor is used to measure the turbidity value in NTU. Calibrations with turbidity standards were carried out every 3 days. Finally, an automated camera was used to take pictures of the samples and discriminate between stable homogeneous samples and unstable formulations presenting phase separation. In this work, phase separation was detected offline by a human operator; however, automated image analysis can be integrated in the future. The samples were tested offline to measure viscosity at a shear rate of 10 $s^{-1}$ and 25°C using a rotational viscometer (ARES Rheometric Scientific, strain controlled, Couette configuration).

## Classification Algorithm

Classification is a type of model assigning labels to regions of the parameters space given only a few known labeled training data.[54] The algorithm used in this study was developed based on the multiple active learning methodologies, which were developed and successfully applied to images classification,[55] music annotation,[56] and text categorization.[57] The method was applied to our case study to distinguish between stable and nonstable formulations faster than randomly accumulating and searching experimental evidence. A naive Bayes classifier and an uncertainty-based sampling strategy were adopted. A flowchart for this framework is shown in Figure 7.

A small set of initial data is needed to first train the model and generate possible experiments for the next step. The trained classifier then can predict the outcome of these experiments and selects the most uncertain experiment (i.e., the one with the lowest confidence regarding the predictions). The selected experiment is then performed on the real system, and the result is added to the dataset, which can be used to train a new classifier. The process is repeated again until a given termination criterion is met. The so-collected final dataset should be more informative than one built using a non-active acquisition method. In this work, a batch sequential design was used, suggesting 12 different experiments at each iteration.

According to Bayes theorem, the probability of given $\boldsymbol{x}$ being class $c$ is given by Equation 1,[58] with the assumption that all attributes are independent given the value of the class variables (Equation 2):

$$p(y = c \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid y = c)p(c)}{p(\boldsymbol{x})} \quad \text{(Equation 1)}$$

$$p(y = c|\boldsymbol{x}) \propto p(\boldsymbol{x})p(\boldsymbol{x} \mid y = c). \quad \text{(Equation 2)}$$

Due to the fact that features S1, S2, and S3 sum to a constant, Equation 2 was modified to consider the subsets of independent features, following a joint distribution. The posterior distribution will then be given by Equation 3:

$$p(y = c \mid \boldsymbol{x}) \propto p(\boldsymbol{x}) \prod_j p(\boldsymbol{x}_j \mid y = c), \quad \text{(Equation 3)}$$

where $j$ denotes each subset.

In this work, there are three subsets of features: (1) features S1, S2, and S3 sum to a constant value and follow the Dirichlet distribution, which is denoted as $\boldsymbol{x}_0$:

$$p(\boldsymbol{x}_0'|y = c) = Dir(\boldsymbol{x}_0' \mid \boldsymbol{\alpha}_c), \quad \text{(Equation 4)}$$

where $\boldsymbol{x}_0' = \boldsymbol{x}_0/t$ and $\boldsymbol{\alpha}_c \in \mathbf{R}^3$ are the parameters of the Dirichlet distribution; (2 and 3) features P1 and T1 follow a normal distribution, which is denoted as $\boldsymbol{x}_i$:

$$p(\boldsymbol{x}_i|y = c) = N(x_i \mid \mu_{i,c}, \sigma_{i,c}) \quad \text{(Equation 5)}$$

where $\mu_{1,c}$ represents the mean of feature $x_i$ prediction $y = c$, $\sigma_{1,c}$ represents the standard deviation of feature $x_i$ for prediction $y = c$, $i = 1$ for P1, and $i = 2$ for T1.

Therefore, the posterior probability is given by

$$p(y = c|\boldsymbol{x}) \propto p(\boldsymbol{x})p(\boldsymbol{x}_0' \mid y = c)p(\boldsymbol{x}_1|y = c)p(\boldsymbol{x}_2 \mid y = c) \quad \text{(Equation 6)}$$

Adopting a constant prior $p(\boldsymbol{x})$, Equation 6 becomes

$$p(y = c|\boldsymbol{x}) \propto p(\boldsymbol{x}_0' \mid y = c)p(\boldsymbol{x}_1|y = c)p(\boldsymbol{x}_2 \mid y = c) \quad \text{(Equation 7)}$$

The parameters were estimated by using maximum likelihood estimation:

$$\max_{\boldsymbol{\theta}} log p(y = c \mid \boldsymbol{x}), \quad \text{(Equation 8)}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}_c, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c)$. $\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c$ have analytical solutions by setting the derivative of log likelihood equals 0; $\boldsymbol{\alpha}_c$ is found by using the mean precision algorithm.[59]

Given the nature of the classification algorithm, each batch at a given iteration, would likely be made of similar experiments. Therefore, the algorithm assigns a score to represent the importance of each sample in terms of local uncertainty and global exploration. From the score, we assign a probability to each of the sample according to a pair of predefined probabilities: local uncertainty and global exploration.

The local uncertainty is measured by Shannon entropy[60]:

$$S_{i,local} = \sum_{k \in c} \widehat{p_{i,k}} \sum_{\log} \widehat{p_{i,k}}, \quad \text{(Equation 9)}$$

where $k$ represents class k, $\widehat{p_{i,k}}$ is the predicted probability, and $c$ is the number of the classes. The score for global exploration is given by

$$S_{i,\text{global}} = \min_j \|\mathbf{x}_i - \mathbf{x}_j\|, \quad \text{(Equation 10)}$$

where $j$ represents sample $j$. Using $S_{i,local}$ and $S_{i,global}$, we sort the samples separately according to each of the scoring criteria. Then, we assign the probability for each of the data points using a discretized exponential distribution, given by Equations 11, 12, 13, 14, 15, and 16. The parameter ($\lambda$) was set with the objective defined in such a way that top 5% of the experiments should be assigned 95% of the probability. In other words, we have a 95% chance of sampling a point within the 5% best points ranked by uncertainty value:

$$I_{local} = (I_{1,local},\ I_{2,local}, \ldots, I_{n,local}) \quad \text{(Equation 11)}$$

$$I_{global} = (I_{1,global},\ I_{2,global}, \ldots, I_{n,global}) \quad \text{(Equation 12)}$$

$$p_{i,local} = \exp(\textit{index of } i \textit{ for } i \textit{ in } I_{local} \mid \lambda) \Big/ \sum_{j=1}^{n} \exp(\textit{index of } j \textit{ for } j \textit{ in } I_{local} \mid \lambda) \quad \text{(Equation 13)}$$

$$p_{i,global} = \exp(\textit{index of } i \textit{ for } i \textit{ in } I_{global} \mid \lambda) / \sum_{j=1}^{n} \exp(\textit{index of } j \textit{ for } j \textit{ in } I_{global} \mid \lambda) \quad \text{(Equation 14)}$$

$$\boldsymbol{p}_{local} = \left(p_{1,local}, p_{2,local}, \ldots, p_{n,local}\right) \quad \text{(Equation 15)}$$

$$\boldsymbol{p}_{global} = \left(p_{1,global}, p_{2,global}, \ldots, p_{n,global}\right), \quad \text{(Equation 16)}$$

where $\boldsymbol{p}_{local}$ and $\boldsymbol{p}_{global}$ are arrays of assigned probability according to $S_{local}$ and $S_{global}$ $i$ is used to denote the index of the sample. Therefore, the overall probability for sampling data is given by

$$P'_{overall} = \boldsymbol{p}_{local} \odot \boldsymbol{p}_{global}, \quad \text{(Equation 17)}$$

where $\odot$ is element-wise multiplication.

The samples are sorted according to the value of $P'_{overall}$, and the overall probability for sampling is given by

$$p_{i,overall} = \exp(\textit{index of } i \textit{ for } i \textit{ in } I_{overall} \mid \lambda) / \sum_{j=1}^{n} \exp(\textit{index of } j \textit{ for } j \textit{ in } I_{overall} \mid \lambda) \quad \text{(Equation 18)}$$

$$I_{overall} = (I_{1,overall},\ I_{2,overall}, \ldots, I_{n,overall}) \quad \text{(Equation 19)}$$

$$\boldsymbol{p}_{overall} = \left(p_{1,overall}, p_{2,overall}, \ldots, p_{n,overall}\right) \quad \text{(Equation 20)}$$

## TSEMO Algorithm

TSEMO optimization was chosen as the DoE algorithm. A detailed presentation of the algorithm can be found in Bradford et al.[35] Briefly, the iterative algorithm consists of the following steps: (1) train GPs for each of the outputs to be optimized based on an initial dataset, (2) sample functions from the obtained GPs using Thompson spectral sampling, (3) find the Pareto front of the sampled functions, (4) find the points that are predicted to give the largest improvement of the hypervolume, and (5) test experimentally the selected data points and add them to the training set.

The optimization procedure can be stopped when the maximum number of evaluations is reached or when the operator is satisfied with the obtained results. This can be automated by terminating the algorithm when the objective functions are lower than a given epsilon.

For this specific case study, the number of suggested experiments at each iteration was set to 8. The input variables were chosen as the concentrations of the ingredients, and the three targets to minimize were chosen as the turbidity value in NTU, the squared distance between the measured viscosity and the target viscosity of 3 Pa×s, and the cost of the adopted ingredients ($/L). The latter was calculated as the sum as the unitary prices of the ingredients multiplied by the adopted amounts in each sample. As this target was an explicit function of the input variables, the code was modified to not train a GP for this specific target and using the directly calculated values instead. The targets were chosen according to the indications of the product supplier, and an ideal product is a stable homogeneous clear formulation with a viscosity as close to 3.00 Pa×s as possible at the lowest possible cost.

## SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.xcrp.2020.100295.

## ACKNOWLEDGMENTS

L. Cao is grateful to BASF for co-funding her PhD. This project was co-funded by the UKRI project "Combining Chemical Robotics and Statistical Methods to Discover Complex Functional Products" (EP/R009902/1) and the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program as a part of the Cambridge Centre for Advanced Research and Education in Singapore (CARES).

## AUTHOR CONTRIBUTIONS

L. Cao conceived the research methodology, built the experimental setup, developed the Bayesian classifier, analyzed samples, ran experiments, and contributed to writing the manuscript; D.R. conceived the research methodology, built the experimental setup, integrated the TSEMO algorithm in the framework, analyzed samples, ran experiments, and contributed to writing the manuscript; K.F. contributed to the use of the software to control the robotic platform; D.S., A.S., and G.K. designed the hardware and software of the robotic platform, guided platform construction, and contributed to writing of the Supplemental Information; W.M. and H.G. provided the chemicals and the case study, guided the optimization through human intuition, and defined the product properties and industrial outlook; L. Cronin and A.A.L. conceived the concept and the specific project, oversaw all aspects of the project, and secured funding for their realization.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: August 21, 2020
Revised: October 25, 2020
Accepted: December 1, 2020
Published: January 6, 2021

Please cite this article in press as: Cao et al., Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE, Cell Reports Physical Science (2020), https://doi.org/10.1016/j.xcrp.2020.100295



## REFERENCES


1. Costa, R., Moggridge, G.D., and Saraiva, P.M. (2006). Chemical product engineering: an emerging paradigm within chemical engineering. AIChE J. *52*, 1976–1986.
2. Yu, Y., Zhao, J., and Bayly, A.E. (2008). Development of surfactants and builders in detergent formulations. Chin. J. Chem. Eng. *16*, 517–527.
3. Goodarzi, F., and Zendehboudi, S. (2019). A comprehensive review on emulsions and emulsion stability in chemical and energy industries. Can. J. Chem. Eng. *97*, 281–309.
4. Conte, E., Gani, R., and Ng, K.M. (2011). Design of formulated products: a systematic methodology. AIChE J. *57*, 2431–2449.
5. Lukic, M., Pantelic, I., and Savic, S. (2016). An overview of novel surfactants for formulation of cosmetics with certain emphasis on acidic active substances. Tenside Surfactants Deterg. *53*, 7–19.
6. Sakamoto, K., Lochhead, R.Y., Maibach, H.I., and Yamashita, Y. (2017). Cosmetic Science and Technology: Theoretical Principles and Applications (Elsevier).
7. Schubert, H., and Engel, R. (2004). Product and formulation engineering of emulsions. Chem. Eng. Res. Des. *82*, 1137–1143.
8. Uhlemann, J., Costa, R., and Charpentier, J.C. (2020). Product design and engineering—past, present, future trends in teaching, research and practices: academic and industry points of view. Curr. Opin. Chem. Eng. *27*, 10–21.
9. Gani, R., and Ng, K.M. (2015). Product design: molecules, devices, functional products, and formulated products. Comput. Chem. Eng. *81*, 70–79.
10. Mattei, M., Kontogeorgis, G.M., and Gani, R. (2014). A comprehensive framework for surfactant selection and design for emulsion based chemical product design. Fluid Phase Equilib. *362*, 288–299.
11. Mattei, M., Kontogeorgis, G.M., and Gani, R. (2012). A systematic methodology for design of emulsion based chemical products. Computer-Aided Chem. Eng. *31*, 220–224.
12. Constantinou, L., Bagherpour, K., Gani, R., Klein, J.A., and Wu, D.T. (1996). Computer aided product design: Problem formulations, methodology and applications. Comput. Chem. Eng. *20*, 685–702.
13. Ng, K.M., Gani, R., and Dam-Johansen, K. (2007). Chemical Product Design: Towards a Perspective through Case Studies, First Edition (Elsevier).
14. Cusller, E.L., and Moggridge, G.D. (2011). Chemical Product Design, Second Edition (Cambridge University Press).
15. Pal, R. (2008). Viscosity models for multiple emulsions. Food Hydrocoll. *22*, 428–438.
16. Derkach, S.R. (2009). Rheology of emulsions. Adv. Colloid Interface Sci. *151*, 1–23.
17. Weissman, S.A., and Anderson, N.G. (2015). Design of experiments (DoE) and process optimization. A review of recent publications. Org. Process Res. Dev. *19*, 1605–1633.
18. Petelin, P., Homar, M., Bajc, A., Kerč, J., and Simona, B. (2012). Use of factorial design for evaluation of factors affecting the chemical stability of sirolimus (rapamycin) in solid dosage form. Acta Chim. Slov. *59*, 156–162.
19. Saripella, K.K., Loka, N.C., Mallipeddi, R., Rane, A.M., and Neau, S.H. (2016). A quality by experimental design approach to assess the effect of formulation and process variables on the extrusion and spheronization of drug-loaded pellets containing Polyplasdone® XL-10. AAPS PharmSciTech *17*, 368–379.
20. Fahmy, R., Kona, R., Dandu, R., Xie, W., Claycamp, G., and Hoag, S.W. (2012). Quality by design I: Application of failure mode effect analysis (FMEA) and Plackett-Burman design of experiments in the identification of "main factors" in the formulation and process design space for roller-compacted ciprofloxacin hydrochloride immediate-release tablets. AAPS PharmSciTech *13*, 1243–1254.
21. Rahman, Z., Xu, X., Katragadda, U., Krishnaiah, Y.S.R., Yu, L., and Khan, M.A. (2014). Quality by design approach for understanding the critical quality attributes of cyclosporine ophthalmic emulsion. Mol. Pharm. *11*, 787–799.
22. Lee, A.R., Kwon, S.Y., Choi, D.H., and Park, E.S. (2017). Quality by Design (QbD) approach to optimize the formulation of a bilayer combination tablet (Telmiduo®) manufactured via high shear wet granulation. Int. J. Pharm. *534*, 144–158.
23. Clayton, A.D., Manson, J.A., Taylor, C.J., Chamberlain, T.W., Taylor, B.A., Clemens, G., and Bourne, R.A. (2019). Algorithms for the self-optimisation of chemical reactions. React. Chem. Eng. *4*, 1545–1554.
24. Alam, S., Aslam, M., Khan, A., Imam, S.S., Aqil, M., Sultana, Y., and Ali, A. (2016). Nanostructured lipid carriers of pioglitazone for transdermal application: from experimental design to bioactivity detail. Drug Deliv. *23*, 601–609.
25. Grangeia, H.B., Silva, C., Simões, S.P., and Reis, M.S. (2020). Quality by design in pharmaceutical manufacturing: A systematic review of current status, challenges and future perspectives. Eur. J. Pharm. Biopharm. *147*, 19–37.
26. Eriksson, L. (2008). Design of Experiments: Principles and Applications (MKS Umetrics AB).
27. Zhang, Y., Apley, D.W., and Chen, W. (2020). Bayesian optimization for materials design with mixed quantitative and qualitative variables. Sci. Rep. *10*, 4924.
28. Fitzpatrick, D.E., Battilocchio, C., and Ley, S.V. (2016). A novel internet-based reaction monitoring, control and autonomous self-optimization platform for chemical synthesis. Org. Process Res. Dev. *20*, 386–394.
29. Marler, R.T., and Arora, J.S. (2010). The weighted sum method for multi-objective optimization: New insights. Struct. Multidiscipl. Optim. *41*, 853–862.
30. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. *6*, 182–197.
31. Peremezhney, N., Hines, E., Lapkin, A., and Connaughton, C. (2014). Combining Gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions. Eng. Optim. *46*, 1593–1607.
32. Houben, C., Peremezhney, N., Zubov, A., Kosek, J., and Lapkin, A.A. (2015). Closed-loop multitarget optimization for discovery of new emulsion polymerization recipes. Org. Process Res. Dev. *19*, 1049–1053.
33. Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. *10*, 50–66.
34. Emmerich, M., Yang, K., Deutz, A., Wang, H., and Fonseca, C.M. (2016). A multicriteria generalization of Bayesian global optimization. Springer Optim. Its Appl. *107*, 229–242.
35. Bradford, E., Schweidtmann, A.M., and Lapkin, A. (2018). Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. J. Glob. Optim. *71*, 407–438.
36. Mokarram, V., and Banan, M.R. (2018). A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables. Struct. Multidiscipl. Optim. *57*, 509–533.
37. Reizman, B.J., Wang, Y.M., Buchwald, S.L., and Jensen, K.F. (2016). Suzuki-Miyaura cross-coupling optimization enabled by automated feedback. React. Chem. Eng. *1*, 658–666.
38. Hsieh, H.W., Coley, C.W., Baumgartner, L.M., Jensen, K.F., and Robinson, R.I. (2018). Photoredox iridium-nickel dual-catalyzed decarboxylative arylation cross-coupling: from batch to continuous flow via self-optimizing segmented flow reactor. Org. Process Res. Dev. *22*, 542–550.
39. Amar, Y., Schweidtmann, A.M., Deutsch, P., Cao, L., and Lapkin, A. (2019). Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis. Chem. Sci. (Camb.) *10*, 6697–6706.
40. Venkatasubramanian, V. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? AIChE J. *65*, 466–478.
41. Rio-Chanona, E.A., Wagner, J.L., Ali, H., Fiorelli, F., Zhang, D., and Hellgardt, K. (2019). Deep learning-based surrogate modeling and optimization for microalgal biofuel production and photobioreactor design. AIChE J. *65*, 915–923.
42. Granda, J.M., Donina, L., Dragone, V., Long, D.L., and Cronin, L. (2018). Controlling an organic synthesis robot with machine learning to search for new reactivity. Nature *559*, 377–381.
43. Steiner, S., Wolf, J., Glatzel, S., Andreou, A., Granda, J.M., Keenan, G., Hinkley, T., Aragon-Camarasa, G., Kitson, P.J., Angelone, D., and Cronin, L. (2019). Organic synthesis in a

Please cite this article in press as: Cao et al., Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE, Cell Reports Physical Science (2020), https://doi.org/10.1016/j.xcrp.2020.100295



modular robotic system driven by a chemical programming language. Science *363*, eaav2211.

44. Ranjan, P., and Spencer, N. (2014). Space-filling Latin hypercube designs based on randomization restrictions in factorial experiments. Stat. Probab. Lett. *94*, 239–247.

45. Loeppky, J.L., Sacks, J., and Welch, W.J. (2009). Choosing the sample size of a computer experiment: A practical guide. Technometrics *51*, 366–376.

46. Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. Stat. Comput. *21*, 137–146.

47. Bose, R.C. (1947). Mathematical theory of the symmetrical factorial design on JSTOR. Indian J. Stat. *8*, 107–166.

48. Gohel, M.C., and Amin, A.F. (1998). Formulation optimization of controlled release diclofenac sodium microspheres using factorial design. J. Control. Release *51*, 115–122.

49. Schweidtmann, A.M., Clayton, A.D., Holmes, N., Bradford, E., Bourne, R.A., and Lapkin, A.A. (2018). Machine learning meets continuous flow chemistry: automated optimization towards the Pareto front of multiple objectives. Chem. Eng. J. *352*, 277–282.

50. Neumann, P., Cao, L., Russo, D., Vassiliadis, V.S., and Lapkin, A.A. (2020). A new formulation for symbolic regression to identify physico-chemical laws from experimental data. Chem. Eng. J. *387*, 123412.

51. Schmidt, M., and Lipson, H. (2009). Distilling free-form natural laws from experimental data. Science *324*, 81–85.

52. Udrescu, S.M., and Tegmark, M. (2020). AI Feynman: a physics-inspired method for symbolic regression. Sci. Adv. *6*, eaay2631.

53. Salley, D.S., Keenan, G.A., Long, D.L., Bell, N.L., and Cronin, L. (2020). A modular programmable inorganic cluster discovery robot for the discovery and synthesis of polyoxometalates. ACS Cent. Sci. *6*, 1587–1593.

54. Lewis, D.D., and Gale, W.A. (1994). A sequential algorithm for training text classifiers. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994 (Association for Computing Machinery), pp. 3–12.

55. Persello, C., and Bruzzone, L. (2014). Active and semisupervised learning for the classification of remote sensing images. IEEE Trans. Geosci. Remote Sens. *52*, 6937–6956.

56. Chen, G., Wang, T.-J., Gong, L.-Y., and Herrera, P. (2010). Multi-class Support Vector Machine Active Learning for Music Annotation (Inderscience Enterprises).

57. Goudjil, M., Koudil, M., Bedda, M., and Ghoggali, N. (2018). A novel active learning method using SVM for text classification. Int. J. Autom. Comput. *15*, 290–298.

58. Zhang, H. (2004). The optimality of naive Bayes. Proceedings of the 17th International Artificial Intelligence Research Society Conference (AAAI Press), pp. 562–567. https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf.

59. Ronning, G. (1989). Maximum likelihood estimation of Dirichlet distributions. J. Stat. Comput. Simul. *32*, 215–221.

60. Shannon, C.E. (1948). A mathematical theory of communication. Bell Syst. Tech. J. *27*, 379–423.