

A soft-computing framework for automated optimization of multiple product quality criteria with application to micro-fluidic chip production.

ZĂVOIANU, A.-C., LUGHOFER, E., POLLAK, R., EITZINGER, C. and RADAUER, T.

2021



A Soft Computing Framework for Automated Optimization of Multiple Product Quality Criteria with Application to Micro-Fluidic Chip Production

Alexandru-Ciprian Zăvoianu^a, Edwin Lughofer^b, Robert Pollak^b, Christian Eitzinger^c, Thomas Radauer^d

^aSchool of Computing, Robert Gordon University, Aberdeen, Scotland, UK

^bDepartment of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria

^cProfactor GmbH, Steyr-Gleink, Austria

^dStratec Consumables, Anif, Austria

Abstract

We describe a general strategy for optimizing the quality of products of industrial batch processes that comprise multiple production stages. We focus on the particularities of applying this strategy in the field of micro-fluidic chip production. Our approach is based on three interacting components: (i) a new hybrid design of experiments (DoE) strategy that combines expert- and distribution-based space exploration with model-based uncertainty criteria to obtain a representative set of initial samples (i.e., settings of essential machining process parameters), (ii) construction of linear and non-linear predictive mappings from these samples to describe the relation between machining process parameters and resulting quality control (QC) values and (iii) incorporation of these mappings as surrogate fitness estimators into a multi-objective optimization process to discover settings that outperform those routinely used by operators. These optimized settings lead to final products with better quality and/or higher functionality for the clients. The optimization module employs a co-evolutionary strategy we developed that is able to deliver better Pareto non-dominated solutions than the renowned NSGA-II multi-objective solver. We applied our proposed high-level surrogate-based multi-objective strategy both in a single/late-stage optimization scenario and in a more challenging multi-stage scenario, yielding final optimization results that improved parameter settings and thus product quality compared to standard expert-based production process parameterizations.

Keywords: single- and multi-stage process optimization, evolutionary multi-objective optimization, hybrid design of experiments, surrogate modeling, process parameters, production quality criteria, micro-fluidic

Email addresses: c.zavoianu@rgu.ac.uk (Alexandru-Ciprian Zăvoianu), edwin.lughofer@jku.at (Corresponding Author) (Edwin Lughofer), robert.pollak@jku.at (Robert Pollak), christian.eitzinger@profactor.at (Christian Eitzinger), t.radauer@stratec.com (Thomas Radauer)

chips.

1. Introduction

The automated optimization of production processes is an important topic in modern industrial facilities, as it is well suited to complement preventive maintenance systems inside holistic Industry 4.0 approaches (see third part in [34]) that aim to prevent poor product quality, faulty batches or machine failures. The main idea is to take well-timed corrective action when items or parts of the production process display certain behaviors that may (at a later stage) result in machine or process failures, degraded throughputs or a downtrend in product quality [33] [41]. Therefore, one key component of production optimization in preventive maintenance systems is the use of techniques from the field of prognostics [15] [32] and forecasting [4] that are able to predict product/machining quality [48] [18] and to recognize at an early stage non-optimal parameter settings or problems dynamically during the production cycle [35]. Ideally, such detection strategies are combined with a mechanism for (re-)balancing the process [46] and ultimately restoring production quality.

This work focuses on the optimization task, as we build upon our initial findings reported in [67] and demonstrate how a clever combination of soft-computing techniques can be used to improve final product quality substantially. More specifically, we investigated how the “process parameters”, that is, controllable (machine) settings that are typically kept fixed over a certain time period in a production process, can be optimized. In a typical (classical) production process, the process (control) parameters are adjusted manually from time to time by production operators and domain experts according to changing external factors, such as new product variants, changes in raw material quality, and supply chain disruptions. Choosing suitable settings is extremely important in an industrial production environment to ensure high-quality, well-functioning products that maximize customer satisfaction. Furthermore, improved parameter settings can reduce production waste [5], which saves costs in the longer term.

In many cases, the experts themselves do not know the ideal settings for the process parameters, that is, those that result in final products with the ideal quality control (QC) indicators. This is because (i) humans tend to set and adjust the parameter values using vague experience and rules of thumb and (ii) operators are more likely to accept conventionally used settings. In other words, manual settings are typically a result of long-term experience with standard default values, which, though they do not degrade product quality, are often far from being ideal.

After outlining in Section 1.1 what distinguishes our approach from other model-based production process optimization strategies reported in literature, in Section 2, we describe and formalize the multi-objective optimization problem (MOOP) that results from the requirement to simultaneously improve multiple quality control indicators. In Section 3, we give an overview of our soft-computing framework for optimizing the parametrization of single and multi-stage industrial processes.

Each of the following three sections focuses on one key part of our proposed framework:

- Section 4 describes a novel hybrid design of experiments (DoE) which is able to select representative samples (i.e., settings of essential machining parameters) in order to build reliable, accurate predictive mappings between the machining parameters and the resulting QC indicators, which represent the final product quality. Its hybrid nature is realized in an expert- and distribution-based sample-gathering mechanism combined with a model-based certainty-sampling technique. The latter aims to decrease parameter uncertainty and to increase predictive performance.
- Section 5 provides details on a predictive mapping construction module based on linear and non-linear regression coupled with a procedure we designed to avoid overfitting and to select the models/variants with the highest expected predictive generalization performance.
- Section 6 describes a multi-objective optimization module we developed that uses the predictive mappings as surrogates for fast fitness calculation. This module incorporates an approach to reduce the number of objectives, and integrates several (co-)evolutionary strategies and archiving concepts for faster convergence.

To verify and evaluate the framework and demonstrate the interaction of our methods, we present two application scenarios that require different levels of direct interaction with human domain experts:

- Section 7 describes the particularities of the micro-fluidic chip production process in which both of these applications scenarios are grounded.
- Section 8.1 is dedicated to the first scenario, where we focused on optimizing a single stage (the final one) of the production process. Our aim is to show how insights gained from a combination of well-chosen soft-computing methods can complement expert knowledge and help to fine-tune process parameters that influence important QC values.

- Section 8.2 focuses on a more complex scenario in which we aimed to perform a full (i.e., multi-stage) process optimization based largely on machine-control settings used over the course of one year. The goal of this optimization was to automatically infer optimal process control settings for the whole production chain at the earliest stage of production (injection molding in our case study). We submitted the new optimal settings to domain experts for detailed analysis and eventual on-site evaluation.

The results we report throughout Section 8 showed that improved quality values were achieved when the new settings were tested in production. Section 9 describes two potential future research avenues and articulates the main conclusions which indicate that, based on the results achieved on both micro-fluidic chip production scenarios, our proposed optimization framework can act as a “higher-level” tool that helps operators/companies to improve overall product quality (and thus industrial competitiveness) and also to gain more insights into the machining process.

1.1. Comparison with Related Work

A model-based strategy for driving a dynamic process towards higher quality has previously been suggested in [2] where the proposed method was shown to display better results than a trajectory tracking approach in closed-loop simulations of a polymerization system. However, the strategy from [2] relies on 80 historic production simulations to construct data-driven models and also requires consecutive measurements within the dynamic batch process of certain variables (e.g., mixture temperature, volume viscosity). In contrast, the approach we envision needs far less training data and does not require in-production measurements as it provides suggestions for the control settings in advance, well before the main production cycle starts. Classical DoE strategies have been proposed in various process optimization applications [20] [6] in order to identify the significance of parameters to the final product quality. While these applications successfully identified key process parameters, the DoE strategies were employed “as-is” without attempts to customize them to the particular problem or to subsequently refine them. In general, advanced DoE strategies (like our new hybrid approach) that are designed to guide the data acquisition process in an ideal direction in a model-based, predictive accuracy sense while considering multiple criteria remain scarce in the field of process optimization.

In [29], the authors used a numerical single-objective solver based on particle swarm optimization (PSO) to perform a surrogate-based optimization of four parameters that control a flux-cored arc-welding process. The surrogate models described in [29] were trained on a set of samples generated by a classical DoE

strategy – a Taguchi L_{25} orthogonal array. Full-factorial design is another classical DoE strategy that has been employed successfully in combination with surrogate-based multi-objective evolutionary algorithms to find optimal process-parameter combinations whenever the search space is sufficiently small [45] (i.e., five process parameters). While both [29] and [29] report overall successful optimization results, the neural network models on which their surrogates are based show signs of overfitting (likely on account of the low number of available data samples). In order to mitigate this, the procedure we propose for constructing surrogate models is biased towards simplicity (i.e., linear models) and informed by predictive performance stability over random cross-validation splits.

A recent approach for process parameter (input setting) optimization has been suggested in [44], which is based on the construction of non-linear surrogate models based using Bayesian and OLS-based regression and conducting optimization through minimization of the total and generalized variances of the responses (model predictions). This approach employs specific non-linear models using polynomials up to an order of two and restricts the synchronous optimization to at most two quality targets (responses), thus to two objectives (achieved by the reformulation of the problem with an additional constraint on the second objective). Similar considerations (two objectives restriction) go to the approach in [58], which employs RBF neural networks as surrogate models, but no explicit DoE is described there. The approach in [21] performs a plain orthogonal experimental design to generate an initial data set for model training and achieves single-objective (i.e., lens thickness) process optimization through reinforcement learning techniques, which require permanent user/operator feedback during the on-line process (which is not available/applicable in our use case).

Compared to these approaches, our optimization framework (i) integrates an advanced hybrid DoE strategy that combines expert- and distribution-based sample-gathering with model-based certainty sampling, (ii) employs multi-layer perceptron networks (MLPs) for predictive surrogate mapping construction which allow to properly model a higher degree of non-linearity possibly contained in the input-output relation and (iii) advocates use of an advanced multi-objective solver in combination with objective reduction strategies, which makes it applicable to (larger) many-objective problems and — in combination with predictive surrogate mappings — allows it to explore the search space within a reasonable amount of time (no real test at the machine needed). The solver we developed, called DECMO2, is based on a co-evolutionary concept with specific archiving procedures and is able to outperform the classical NSGA-II algorithm (see Results section). The outcome of the process optimization solver is a set of potential parameter solutions which yield Pareto-optimal trade-offs with respect to the given QC indicators. These parameter solutions can be used directly by operators as new machine settings and should improve the final product quality. In fact, we

were able to achieve a significant improvement with one of the Pareto-optimal settings suggested by our approach. Furthermore, our approach can be and has been successfully applied (see Section 8) to a multi-stage production process – a context that has not been considered in any of the aforementioned approaches.

When comparing with our initial strategy sketched in [67], the present work features:

- an extended and fully fleshed out hybrid DoE procedure;
- an improved mapping construction routine centered on a simple two-step assessment of estimated predictive performance stability designed to mitigate the risk of cross-validated model selection, based on an optimistically biased estimate of generalization performance;
- the development and integration of multi-stage predictive mappings that enable the automatic optimization of the full multi-stage production process (in [67], only a single-stage process has been considered);
- the inclusion of on-site validation results for the most promising solutions discovered by our approach.

The organization of the remaining paper is as follows: Section 2 defines the optimization problem and discusses the basic challenges therein; Section 3 provides a round picture of our optimization framework by visualizing a work-flow which shows the components and explains their characteristics and basic functionality; Section 4 describes the hybrid design of experiments procedure with all details resulting in a pseudo-code demonstration of its algorithm; Section 5 presents aspects regarding the predictive mapping construction module; Section 6 describes the multi-objective optimization module, including a visualization of the work-flow of our evolutionary-based solver; Section 7 demonstrates the application scenarios in which our approach has been verified and evaluated; Section 8 shows comprehensive results we obtained from these scenarios; Section 9 concludes the paper and discusses potential future works.

2. Problem statement

As the name suggests, a multi-stage industrial production process is characterized primarily by the fact that two or more separate processing steps must be orchestrated to deliver the final product. Usually, each of these processing steps is carried out with the help of a specialized and independently controlled machine. The goal of this research was to provide an efficient approach for discovering the best process-parameter combination for each machine that is part of the multi-stage process in order to improve final product quality.

2.1. Formalization

Let us assume that all the production stages of a given multi-stage industrial process can be controlled using a total of p process parameters $\{x_1, \dots, x_p\}$, $x_1 \in [l_1, u_1], \dots, x_p \in [l_p, u_p]$, with l_p the lower allowed value and u_p the upper allowed value of parameter p , and that one wishes to evaluate the final outcome of this production process by means of q QC indicators. Under the assumption that (some of) these process parameters can directly impact every essential QC indicator under examination (e.g., due to being part of the final customer's specifications), we can define the QC indicator set as $\{f_1(\vec{x}), \dots, f_q(\vec{x})\}$, where $\vec{x} = \{x_1, \dots, x_p\} \in [l_1, u_1] \times \dots \times [l_p, u_p]$ denotes the process parameter combination (i.e., a multi-dimensional point).

When the essential QC indicators of a production process are to be optimized, the goal is ideally to find a parameter combination ($\vec{x}^* = \{x_1^*, \dots, x_p^*\}$) that delivers the best values for all QC indicators simultaneously. Thus, when assuming an unconstrained minimization formulation for optimal QC values, \vec{x}^* has the property that:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{arg\,min}} F(\vec{x}) = \underset{\vec{x}}{\operatorname{arg\,min}} [f_1(\vec{x}), \dots, f_q(\vec{x})] \quad (1)$$

$$\text{where } x_j \in [l_j, u_j] \subset \mathbb{R}, \quad 1 \leq j \leq p.$$

In Equation 1, the ideal (i.e., minimal) values for all QCs, the variation intervals of the process parameters (i.e., $[l_j, u_j] \subset \mathbb{R}$) are defined by domain experts according to production goals and machining limitations. Typically (and in our case as well), $f_i(\vec{x})$ denotes a surrogate mapping for the i th quality criterium, that is, the expected outcome of quality criterium i when the machining parameters are set as in \vec{x} . Alternatively, if the time lag between changing machining parameters and seeing the effect in final product quality is sufficiently short, one may consider a real on-line test with each setting during optimization and use the resulting QC value as f . However, since this is laborious and causes wear to the machines during optimization, this approach is often unrealistic.

Since in many practical cases there are intrinsic conflicts between the various QC indicators, Equation (1) usually defines a nontrivial multi-objective optimization problem (MOOP) for which there is no single solution (i.e., $\nexists \vec{x}^* \in \mathcal{S}$), as it is impossible to achieve all the ideal QC values simultaneously. The solution to such an optimization problem comes in the form of a Pareto-optimal set (*PS*) defined by the property that no member of the *PS* is better than any other with respect to all the objectives considered. More formally,

$$PS = \{\vec{x} \in \mathcal{S} \mid \nexists \vec{y} \in \mathcal{S} (\forall_{k \in \{1, \dots, q\}} (f_k(\vec{y}) < f_k(\vec{x})))\} \quad (2)$$

with S being the solution space. The projection of the PS in objective space is called the Pareto Front (PF), and its purpose is to illustrate the trade-offs between the achievable optimal values of the individual optimization objectives. For most MOOPs, and especially for those based on real-value optimization, the PS is infinite and/or unknown, and practitioners of multi-objective optimization generally settle for discovering a Pareto non-dominated set (PN) that uses a fixed number of members (solution candidates) but is able to provide a (very) good approximation of the PS . In light of these considerations, the goal of this work can be refined to introducing a framework for discovering high-quality PN s which contain solution candidates based on optimized process parameter settings. As these candidates are represented by multi-dimensional vectors whose entries define their expected QC outcomes, the PN s that contain them provide those operating the process with a very accurate picture of the trade-offs between the best achievable QC values for all criteria to be optimized: Given the minimization formulation in Equation (1), lower entries indicate higher-quality outputs. Thus, QC vectors that contain only low entries are indicative of parameter settings that deliver a high quality across all criteria considered.

3. Our Framework

The soft-computing framework we propose is based on three modules, independent of the particularities of the process optimization scenario:

1. a novel hybrid design of experiments (DoE) which is able to select representative samples (= settings of essential machining parameters) in order to build reliable and accurate predictive mappings between the machining parameters and the resulting QC values that represent the final product quality;
2. an effective modeling module that can, given the particularities of the optimization scenario, create accurate predictive mappings for QC indicators (i.e., linear or non-linear regression models) based on the samples selected from the hybrid DoE; these can be used as surrogates $\{f_1(\vec{x}), \dots, f_q(\vec{x})\}$ during the optimization process to avoid time-intensive on-line tests;
3. a robust and efficient optimization module that is able to deal with both multiple objectives and non-linear targets (i.e., QC indicators in our case).

In the case of the optimization module, we chose to rely on state-of-the-art multi-objective evolutionary algorithms (MOEAs) as default solvers, since they have emerged as one of the most successful methods for solving challenging MOOPs [9]. These algorithms (i) have the ability to produce accurate PN s after single runs and (ii) provide robust performance on practical problems from many application domains [8]. The

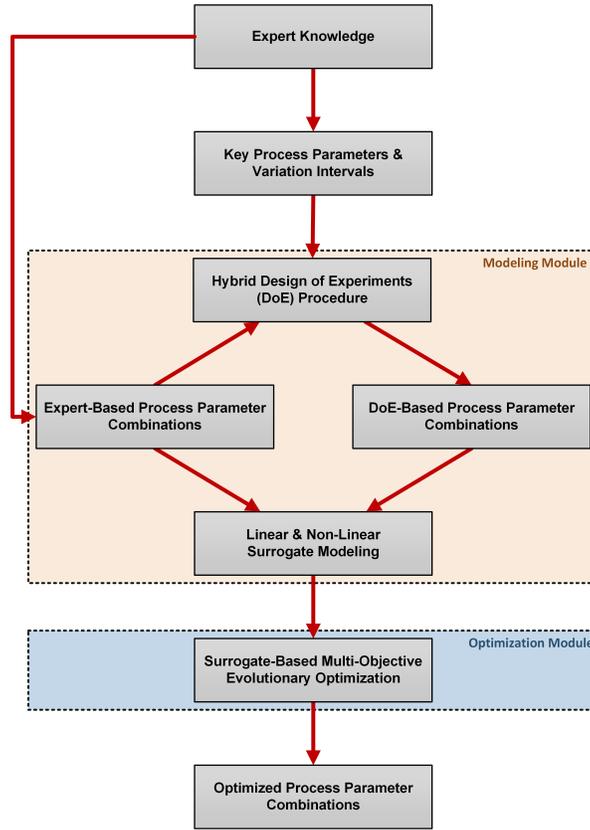


Figure 1: The proposed soft-computing framework for optimizing process parameter values with respect to multiple objectives.

main drawback of MOEAs in particular (and of evolutionary computation solvers in general) is their need to evaluate the quality (i.e., fitness) of a very large number of solution candidates (i.e., individual parameter combinations in our case) during the optimization run. This is particularly problematic in application scenarios that are based on batch processes, because the QC indicators can in many cases be determined only after the completion of a whole production cycle. Since a production cycle can last between a few seconds and several days, there is a significant lag between setting the process control parameters and observing their impact on the final product quality. Thus, an on-line (i.e., machine-in-the-loop) evaluation of solution candidates generated by the MOEAs during the optimization runs is highly unrealistic.

Ideally, this drawback could be overcome by having access to analytical simulation models of the QC indicators. Directly using the actual mathematical formulae that define $f_i(\vec{x}), 1 \leq i \leq q$ from Equation 1 inside the MOEA solver would then provide a rather straightforward solution to the entire process optimization.

tion task. However, for most application scenarios (micro-fluidic chip production included) such formulae are unknown or difficult to model explicitly. Therefore, a suitable strategy [25] [43] [19] relies on (i) constructing data-driven regression models (predictive mappings) between the process parameters (inputs) and the corresponding QC indicator values (outputs), and (ii) using these regression models as surrogate quality estimators $\{f_1(\vec{x}), \dots, f_q(\vec{x})\}$ that can steer the evaluation-intensive MOEAs towards promising regions of the search space when used as fitness-function evaluators. In our optimization framework, the modeling module is responsible for creating accurate predictive mappings based on the particularities of the optimization scenario under consideration.

Obtaining data samples with good coverage of the input feature space can lead to a significant uncertainty reduction in the resulting regression models (by increasing parameter significance and reducing extrapolation cases) and is essential when aiming for high-quality surrogates that exhibit good generalization on new samples [49]. Therefore, in Section 4 we demonstrate how to combine expert knowledge and model/data-based DoE effectively in order to generate valuable data samples for training surrogate models.

In Figure 1, we present the work-flow that characterizes our framework for optimizing the parametrization of single- and multi-stage industrial processes. The core components of the framework are discussed in the subsequent three sections. From a methodological point of view, the difference between single- and multi-stage processes lies in the (achievable) prediction horizons of the surrogate models and in the possibility and means of performing sample selection for surrogate construction. These distinctions are evidenced by a concrete use case in the Results section.

4. Hybrid Design of Experiments (DoE)

4.1. Integration of expert knowledge

Expert knowledge is integrated at the initial data collection / generation stage to obtain a representative initial set X of process-parameter combinations. Such a set is needed for the model-based sample selection discussed in the subsequent section (which requires an input data/regression matrix) to produce an initial estimate of the expected parameter uncertainty in the models. First, based on expert input, a so-called Cause-Effect (CE) diagram can be established to identify the process parameters that are most likely to influence QC indicators. These are the relevant process parameters to be optimized. Figure 2 presents an example CE diagram of the bonding liner stage of micro-fluidic chip production. Expert knowledge is not only used for parameter identification, but also to filter invalid parameter combinations and to restrict the

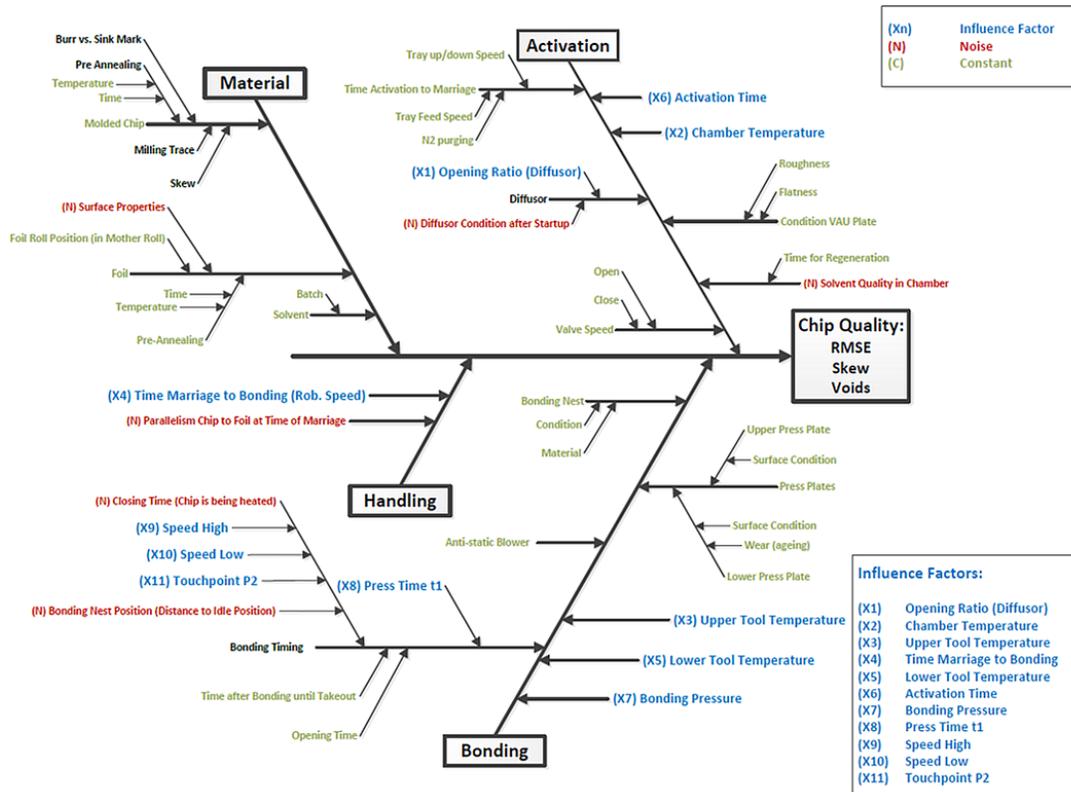


Figure 2: A cause-effect diagram for the micro-fluidic chip production process at the bonding liner. The relevant process parameters (marked x_1 to x_{11}) are highlighted in blue and explicitly mentioned in the lower right corner.

domain of each parameter, that is, the feasible search intervals $[l_j, u_j]$ for the j process values considered by the unconstrained optimization problem in Equation (1) and used by the hybrid DoE algorithm to generate the corner points.

After identification of the relevant process parameters, the first few parameter settings (i.e., data samples) to be used are recommended by the production process operators. Computer-aided design-selection strategies, such as the Taguchi L_{12} method or full factorial [42], may support the experts in choosing appropriate combinations given parameter domain and combination-related limitations. In many cases, a few data samples corresponding to expert-based settings are also available in advance; these are settings with which the experts are familiar due to long-term use or which the experts see as beneficial based on past experience with various product charges or types etc.

4.2. Distribution- and Model-Based Sampling for Uncertainty Reduction

The second data-collection step builds on a domain-independent DoE-based strategy. Using the initial samples X proposed by the domain experts as input, it aims to generate N new samples that are well distributed in the parameter space and simultaneously provide the greatest reduction in prediction uncertainty (\rightarrow hybrid DoE).

For measuring this reduction, we propose using various optimal-design criteria, such as A-optimality, D-optimality and E-optimality [16] — which all rely on the Fisher information matrix [17] — while assuming linear dependencies (mappings) between process parameters and QC values because of the (usually) very low number of expert-based samples. These model-based design criteria are combined with pure space-filling techniques based on Latin hypercube (LH) sampling [38] and min-max optimization [26], which guarantees that the samples to be tested are well-distributed in the input parameter space.

In particular, our strategy (shown in Algorithm 1) starts by generating a pool of $M + 2^p$ samples that are drawn from the parameter space, the dimensionality of which is denoted by p . M of these samples are generated via LH sampling combined with min-max optimization, and the remaining 2^p are the corner points of the parameter space, as we do aim to reduce the likelihood of extrapolation wherever possible. If p is too large, a subset of the most important (i.e., influencing) process parameters that can be identified by using expert knowledge can be used (e.g., the 11 bonding liner parameters shown in the CE diagram in Figure 2). Alternatively, a statistical analysis, such as an ANOVA test [22], of the initial parameter settings gathered from the experts can be performed.

The DoE strategy we propose combines various sample-selection strategies. In each odd-numbered iteration of sample selection, each of the remaining $M + 2^p - |S|$ samples generated (where S and $|S|$ denote the set and number of samples selected so far) is evaluated based on how much it improves one of the following criteria:

- A-optimality (variant 1): the ability to minimize the trace of the inverse of the Fisher information matrix [17];
- D-optimality (variant 2): the ability to maximize the determinant of the Fisher information matrix;
- E-optimality (variant 3): the ability to maximize the minimum eigenvalue of the Fisher information matrix.

Algorithm 1 Hybrid DoE for Sample Selection

Input: X = set of initial samples, p = number of process parameters, M = no. of LHS samples to explore, $N =$

$|S|$, $optCrit$ = optimality criterion

Output: S = suggested Hybrid DoE samples

```
1: function HybridDoE( $X, M, N, optCrit$ )
2:    $S \leftarrow \emptyset, G \leftarrow \text{GenerateLHSSamples}(size = M)$ 
3:    $G \leftarrow G \cup \text{GenerateCornerSamples}(size = 2^p)$ 
4:   for  $j = 1$  to  $N$  do
5:     for  $i = 1$  to  $M + 2^p$  such that  $\vec{x}_i \leftarrow G_i \wedge \vec{x}_i \notin S$  do
6:        $A, D, E, Dst \leftarrow \emptyset$ 
7:        $X_{ext} \leftarrow [X; S; \vec{x}_i]$ 
8:       if odd( $j$ ) then ▷ iteration no. is odd
9:          $A_i \leftarrow \langle key = \text{trace}([X_{ext}^T X_{ext}]), val = \vec{x}_i \rangle$ 
10:         $D_i \leftarrow \langle key = \text{det}([X_{ext}^T X_{ext}]), val = \vec{x}_i \rangle$ 
11:         $E_i \leftarrow \langle key = \text{minEig}([X_{ext}^T X_{ext}]), val = \vec{x}_i \rangle$ 
12:       else ▷ iteration no. is even
13:         $L_{\vec{x}_i} \leftarrow \text{EuclideanDistances}(\vec{x}_i, [X; S])$ 
14:         $Dst_i \leftarrow \langle key = \text{selectMin}(L_{\vec{x}_i}), val = \vec{x}_i \rangle$ 
15:       end if
16:     end for
17:     if odd( $j$ ) then ▷ iteration no. is odd
18:       if  $optCrit = \text{"A-optimality"}$  then
19:          $\vec{x}_{sel} \leftarrow \text{GetValForMinKey}(set = A)$ 
20:       end if
21:       if  $optCrit = \text{"D-optimality"}$  then
22:          $\vec{x}_{sel} \leftarrow \text{GetValForMaxKey}(set = D)$ 
23:       end if
24:       if  $optCrit = \text{"E-optimality"}$  then
25:          $\vec{x}_{sel} \leftarrow \text{GetValForMaxKey}(set = E)$ 
26:       end if
27:     else ▷ iteration no. is even
28:        $\vec{x}_{sel} \leftarrow \text{GetValForMaxKey}(set = Dst)$ 
29:     end if
30:      $S \leftarrow S \cup \vec{x}_{sel}$ 
31:   end for
32:   return  $S$ 
33: end function
```

The sample that is able to deliver the best improvement with regard to the pre-selected optimality criterion is added to S during each odd-numbered iteration (lines 18 to 26 of Algorithm 1).

In each even-numbered iteration, we compute the minimal distances between each of the remaining $M + 2^p - |S|$ generated samples and the samples already selected (including the initial expert-based ones). In this case, the sample with the largest minimal distance is selected, as its novelty content is highest (line 28 of Algorithm 1).

Note that, as we apply linear model-based DoE, in each iteration, the Fisher information matrix is equivalent to the Hessian matrix $X_{ext}^T X_{ext}$, with X_{ext} containing the initial expert-based samples (stored in X) plus all the samples selected so far over the DoE iterations (stored in S) and the new sample to be checked for improvement of the optimality criteria (line 7 in Algorithm 1).

In summary, a maximum of N most important samples are selected from an initial pool of $M + 2^p$ meaningful samples, where N is a user-defined parameter. In general, N is chosen according to a realistic number of tests that can be conducted or that the operators are ready to conduct at the machine within a reasonable amount of time at reasonable cost and effort to determine the ultimate impact on product quality. However, these N samples can also be ordered from most to least important based on their estimated novelty content (i.e., their ability to reduce model/parameter uncertainty), and a subset of samples (for testing) can be obtained by thresholding (e.g., in accordance with a stability measure, as described in Section 8.1). We want to emphasize that our DoE technique describes a fully unsupervised strategy that relies only on the input data matrix and is independent of associated output measurements (i.e., QC values). This is a great advantage whenever sample outputs are costly or very time-intensive to obtain (as in our application scenario). Using a high value for N , the DoE can produce many samples in advance to offer insights into overall uncertainty, and from the sorted list of samples, only the most important ones need to be chosen for final testing and gathering of corresponding QC values (i.e., sample outputs).

5. Predictive Mapping Module

As with most data-driven modeling tasks, a standard pre-processing step should aim to reduce input dimensionality (and thus also potential model complexity) by removing some process parameters that are expected to have a negligible influence on the targeted QC indicators. The filtering can be done based on expert input or a combination of expert knowledge and statistical methods. For example, based on expert knowledge, the process-control parameters can be divided into broad groups (e.g., low, medium,

high) according to their perceived influence on the final QC values. A multi-way ANOVA analysis [22] of the available data can subsequently confirm the impacts of various parameters or groups of parameters. Parameters that do not appear to have any influence can be ignored in future modeling steps.

The default strategy of the proposed modeling module is to train linear regression models between process parameter settings (inputs) and QC values (targets). This is an intuitive approach, especially considering the limited number of data samples available due to the high effort required for collecting the resulting QC values (see also previous section and Results section for a concrete use case). Whenever data is scarce, linear models are likely to deliver the best (generalization) performance [23]. Whenever a reasonable number of samples is available, more advanced non-linear regression modeling paradigms, such as shallow and/or deep artificial neural networks (ANNs) [24], support vector machines [10] or genetic programming [47], are also to be tested, especially for QC indicators that could not be modeled with sufficient accuracy.

One major concern that must be addressed by any data-driven learning strategy is overfitting, especially in the case of non-linear models. Therefore, our approach is to always apply an n -fold cross-validation procedure for both (i) tuning the model hyper-parameters and (ii) estimating the predictive performance of the final models (i.e., average R^2 over the n splits). The relatively low overall sample count (expert-based and hybrid DoE) that can be tested in most multi-stage industrial production processes imposes limits on the number of cross-validation folds (e.g., $3 \leq n \leq 5$). Furthermore, as illustrated in [55], having limited samples compounds the known issue that the cross-validated performance of the best hyper-parameter configuration is an optimistically biased estimate of the performance of the final model [28] [54]. As we attempt to mitigate this phenomenon, we propose a simple two-step assessment of estimated predictive performance stability: (i) repeat the n -fold cross-validation procedure t_{cv} times using new randomly generated splits and record each time the expected performance of the best-performing model (e.g., the model with the highest average R^2 over the n folds), and (ii) compute the average ($\mu_{t_{cv}}$) and jackknife bias-corrected relative standard deviation ($rsd_{t_{cv}}$) [40] of the previously recorded best-performances over the t_{cv} cross-validation repeats. When both linear and non-linear regression are investigated for a given QC indicator, the corresponding values of $\mu_{t_{cv}}$ and $rsd_{t_{cv}}$ will then offer valuable insights into whether the non-linear modeling strategy is likely to achieve superior modeling performance compared to its linear counterpart. While a statistical hypothesis test, such as Mann-Whitney-Wilcoxon [36], can be employed to confirm observed differences observed in average performance between the linear and the best-performing non-linear model across the t_{cv} cross-validation repeats, this can easily be seen as redundant: Given the limited sample count, if non-linear modeling does not exhibit a clearly higher $\mu_{t_{cv}}$ (coupled with a lower or similar $rsd_{t_{cv}}$), the simpler, linear regression is

preferable as it is less likely to overfit the data [23].

6. Multi-objective Optimization Module

Problems that have more than three objectives (QC indicators or groups of QC indicators to be minimized in our case) that must be optimized simultaneously are called many-objective optimization problems. Specialized algorithms that deal with such types of problem have recently been proposed, for instance, NSGA-III [13] and MOEA/D adaptations [53]. Nevertheless, given the serious inherent difficulty of trying to solve (surrogate-based) many-objective optimization problems, a natural first step is to try to reduce the number of objectives to be optimized. This would in turn facilitate usage of more robust MOEAs that incorporate well-proven efficient multi-objective optimization strategies. When opting to reduce the number of objectives, extra care must be taken to ensure that potential solutions of the reduced MOOP formulation remain relevant to the original problem. Moreover, by reducing objectives, some of the characteristics of the original problem are lost and potential solutions that would be based on them could also become unattainable.

6.1. Reducing the number of objectives

There are numerous ways of reducing the number of objectives in a many-objective optimization problem, but the majority can be broadly classified as either explicit or implicit.

Explicit objective-reduction strategies rely on the assessment of the decision maker (DM), that is, the process operator in our case – regarding the relative importance of each objective. Thus, one straightforward approach is to directly remove objectives with a (very) low priority from the problem formulation. Alternatively, these low-priority objectives could be replaced with a newly defined synthetic objective that aggregates them. For example, such a synthetic objective can easily be constructed via any of the classical objective-reduction techniques [39] that require no articulation of preference on behalf of the DM (e.g., Tschebyscheff min-max, global-criterion, averaging) or a priori articulation of preference (e.g., weighted sum, lexicographic ordering, goal programming).

Implicit objective reduction is primarily data-driven and can be achieved by clustering objectives that are cross-correlated. Depending on the strength of intra-cluster cross-correlation, the entire cluster of objectives could be reduced to:

- one of its members that acts as a “cluster representative” during the optimization process (very strong intra-cluster cross-correlation);

- a new synthetic objective that aggregates all the members of the cluster (mild intra-cluster cross-correlation).

Obviously, explicit objective reduction can be far more subjective in nature (e.g., when using a-priori articulation of preference), and this can be either an advantage or a disadvantage in the subsequent many/multi-objective optimization, as the final results will be biased more by the initial choices made by the DM. Using an implicit objective-reduction strategy is expected to result in more unbiased optimization results that can provide the DM with an accurate understanding of the existing trade-offs. Depending on the complexity and specifics of the objective reduction task, implicit and explicit strategies can be combined in order to achieve a highly reduced but useful MOOP formulation – as shown, for example, in [51], where type-1 and type-2 fuzzy logic systems were used to define MOOPs suitable for optimizing real-world task-allocation processes.

6.2. Multi-objective evolutionary algorithms

Due to its good performance and relatively simple design, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [14] has become the default go-to metaheuristic multi-objective solver over the 15 years since its introduction. NSGA-II incorporates a highly elitist evolutionary model underpinned by a selection-for-survival operator that features a primary Pareto-based filtering criterion (i.e., non-dominated sorting) and a secondary tie-breaking objective-space crowding measure. The non-dominated sorting operator, although simple when compared to newer MOEA strategies (like decomposition [61]), is highly robust and enables NSGA-II to discover high-quality *PNs* in many application domains [37] [59]. The relative success of NSGA-II among other capable modern MOEAs (like PAES [30] and PESA-II [11]) has also helped to popularize two genetic operators for real-valued optimization: simulated binary crossover (SBX) and polynomial mutation (PM) [12].

In our framework, we also employ a MOEA approach termed DECMO2 [65], which we designed specifically to exhibit fast average convergence and well-spaced *PNs* across a broad class of MOOPs by capitalizing on early insights [1] [64] into the strength of coevolutionary solvers in multi-objective optimization contexts. In order to achieve this, the DECMO2 co-evolutionary model combines (and dynamically pivots between) three well-established multi-objective search-space exploration paradigms, each of which is implemented via a dedicated subset of individuals (i.e., a sub-population):

- *P* is one of the two equally sized sub-populations actively evolved in DECMO2, and uses a SPEA2 [63] evolutionary model that is centered around the environmental selection operator. This operator also

implements a two-tier (Pareto dominance and objective-space density) selection-for-survival strategy that is very similar to that used in NSGA-II. The main genetic operators used to evolve population P are SBX and PM.

- Sub-population Q is also actively evolved, but it implements the DEMO [50] / GDE3 [31] evolutionary model, which aims to capitalize on the very good performance exhibited by differential evolution [52] operators (e.g., DE/rand/1/bin in our case) on continuous optimization problems.
- The third multi-objective optimization paradigm comes in the form of an archive A of well-spaced elite solutions that are maintained according to a decomposition-based strategy similar to that proposed in [27] and popularized by MOEA/D-DE [62]. Although A can be regarded as a passive sub-population, during each odd-numbered iteration and during stages of the run in which the other search paradigms under-perform, a limited number of solution candidates are evolved directly from A using differential evolution.

During the run, DECMO2 actively rewards the best-performing strategy – which is assessed using the relative ratio of individuals evolved in the current iteration that have been accepted in the elite archive A – by allowing the sub-population that implements it to generate an extra number of $|B| = \frac{1}{10} \cdot |A|$ bonus individuals. The default sizes of sub-populations P and Q are also set in relation to the size of the main archive: $|P| = |Q| = \frac{|A| - |B|}{2}$. A schematic overview of the search strategy proposed by the DECMO2 coevolutionary solver is presented in Figure 3.

The implementation of DECMO2 and NSGA-II used to carry out the numerical experiments described in Section 8 can be accessed at: https://github.com/czavoianu/DECMO_Algs_JMetal3.

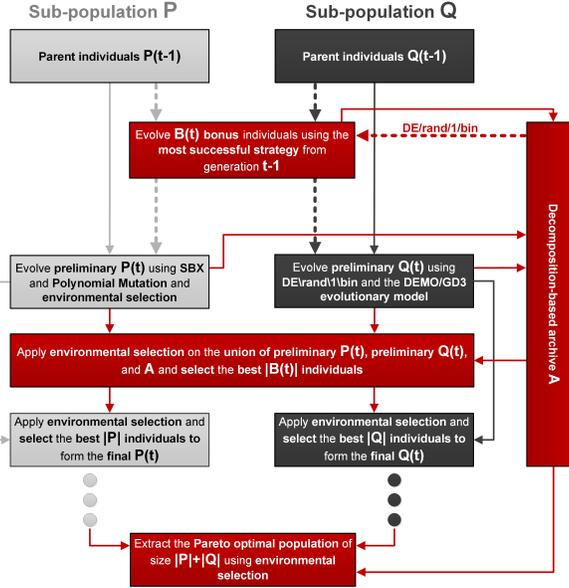


Figure 3: The DECMO2 evolutionary model for iteration (generation) t .

7. Application Scenarios — Micro-fluidic Chip Production

The lower half of Figure 4 presents a high-level overview of the micro-fluidic chip production process we used as a case study for our optimization framework. The process contains three serially-linked stages that are performed by specialized machines: an injection-molding machine (IMM), an industrial oven (IO) and a bonding liner (BL). While the first and last stages of the process can and should be parameterized in order to improve QC indicators, the industrial oven is typically used with standard fixed settings. In the classical scenario (on which we aimed to improve), all p parameters that can be varied to control the IMM and the BL are set using expert-based knowledge.

The outcome of this multi-stage process is a batch of micro-fluidic chips that are used for sample preparation in DNA (deoxyribonucleic acid) sequencing. The quality of the final products is supervised at the end of the production cycle (i.e., after the bonding stage) using up to $q = 26$ QC indicators that belong to three main groups: RMSE, flatness and flat panel inspection (FPI) flaws.

These indicators are based on closed-loop surface inspection using machine learning classifiers [56]. While the quality-control process is highly automatized for most QC indicators, an analysis by human experts is still required in order to estimate some qualitative aspects. More importantly, the a posteriori nature of

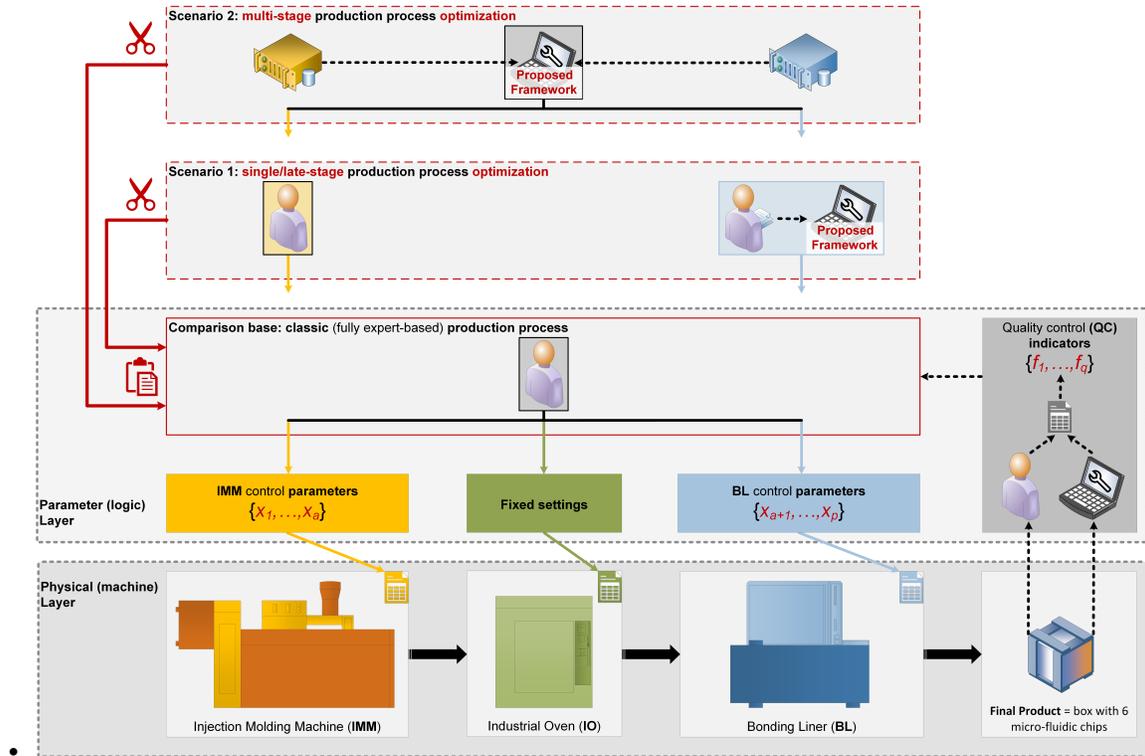


Figure 4: Overview of the classical, expert-centered industrial process for producing micro-fluidic chips and of the two main optimization scenarios we investigated. The thick arrows show the direction of the physical production flow, and the thin arrows indicate the information flow from (solid) and to (dashed) the process controllers.

the entire inspection process means that faulty batches can be disregarded only after having been produced. While this approach improves customer satisfaction, it does not reduce the costs and waste associated with suboptimal parameter settings producing suboptimal (or bad) chips. Hence, the motivation for optimizing the parameters that control the production process is high.

The upper half of Figure 4 illustrates the two optimization scenarios with which we tested our computational intelligence framework:

- Scenario 1 investigated whether the QC indicators can be improved by optimizing only the last stage of the production process (i.e., the BL parameters) while keeping the IMM control parameters fixed to expert-based settings. Thus, this single/late-stage optimization can be seen as a “fine-tuning” experiment that tried to deliver the best final quality from a batch of intermediate products output by

the injection-molding stage. A key operational constraint was therefore to obtain good results with a relatively limited number of tries, and this was achieved by constructing predictive mappings (for QC indicators) that are based on a combination of expert-knowledge and data-driven DoE methods.

- Scenario 2 investigated whether it is possible to optimize the QC indicators by relying mostly on data-driven methods and past parameter settings for both the IMM and the BL. The purpose of this global multi-stage optimization was two-fold: firstly, we aimed to determine whether predictive mappings based solely on latent expert knowledge are as effective as the hybrid ones from Scenario 1, and secondly, we investigated whether additionally optimizing the IMM parameters brings further improvement in QC values.

8. Numerical Experiments and Results

In the next two sections, we describe the details of the numerical experiments we performed and of the results we obtained for the single/late-stage and multi-stage multi-objective optimization of the micro-fluidic chip production process under consideration.

8.1. Scenario 1: Single/late-stage optimization

In the initial stage of the data-collection procedure, domain experts settled on 12 process parameter combinations for the bonding liner (BL) that were expected to deliver good-quality chips based on 26 QC indicators. Each of these process parameter combination contained values for the 11 control parameters identified in the CE diagram shown in Figure 2. Using these 12 combinations (i.e., expert-based data samples), we subsequently performed a series of numerical experiments to measure the evolution of model uncertainty when using different settings for our hybrid DoE strategy listed in Algorithm 1, that is, choices of the optimality criterion (A, D or E) and the number of samples to be selected.

Figure 5 plots on the y-axis the value of $cond(X)$ (the condition of the regression covariance matrix $X^T X$) for various DoE criteria when incrementally selecting up to $N = 100$ samples (thus, X as regression matrix contains all samples selected so far). Here, $cond(X)$ is defined as:

$$cond(X) = \frac{\max(\vec{eig}(X^T X))}{\min(\vec{eig}(X^T X))} \quad (3)$$

with $\vec{eig}(X^T X)$ the vector containing all eigenvalues of $X^T X$. It elicits the proportion between the maximal and the minimal eigenvalue of the regression covariance matrix and is a well-known proper measure of parameter and model uncertainty in the context of regression problems [22].

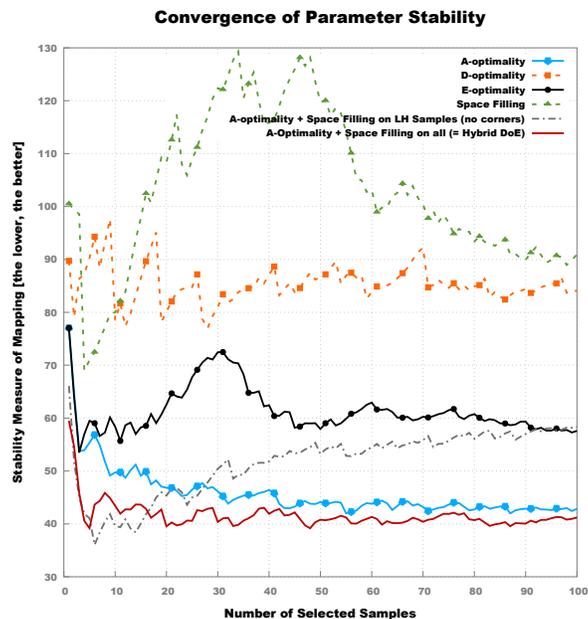


Figure 5: Parameter stability for various DoE-based sample-selection criteria when selecting samples incrementally (step-wise) up to a total of 100.

This numerical experiment showed that A-optimality considerably outperforms other sample-selection criteria, both in the state-of-the-art stand-alone configuration [16] and in our hybrid combination with a space-filling approach (i.e., hybrid DoE), which uses a sample pool based on LH and corner points. This combination of LH and corner points was actually shown to outperform the classical A-optimality criterion, as the model uncertainty could be reduced faster, especially in the crucial first 10-20 samples. For comparison, Figure 5 also presents the performance of hybridizing A-optimality with a space-filling strategy that considers only Latin hypercube samples and no corner points (dash-dotted gray line without point markers). While this simpler strategy performs similarly after 7-15 samples to the hybrid DoE, the LH-only variant seems unable to deliver stability with regard to expected model certainty when incrementally more samples are being considered.

Given these observations and on-site availability, 11 additional (bonding liner) process-parameter combinations were (i) generated using the hybrid DoE strategy and (ii) thoroughly tested at the end of the production cycle in order to determine the 26 corresponding QC values for each parameter combination. Including the 12 initial expert-based parameter combinations, the data set used as input to the predictive

mapping module thus comprised a total of 23 samples.

We attempted to construct high-quality predictive mappings for each of the 26 QC indicators using both linear regression and (non-linear) shallow neural networks, that is, multi-layer perceptrons (MLPs) with a single hidden layer. We adopted the standard approach and combined a best-parameter grid search with a final model-selection process based on $n=5$ -fold cross-validation [7] [3] in order to avoid overfitting the MLPs. Thus, in the grid search aimed at discovering the best training parameters for the MLPs, we varied the number of hidden units between 2 and 28, and we varied the two control parameters of the back-propagation [57] algorithm:

- the learning rate (between 0.05 and 0.5 with a step size of 0.05) and
- the momentum (between 0 and 0.9 with a step size of 0.1).

As a result, for each QC target, 2700 candidates (i.e., MLP-based regression models) were created during the best-training-parameter grid search. The final choice of the “best-performing MLP” was made using a model-selection strategy designed to balance predictive accuracy (i.e., imposing a 5-fold cross-validation R^2 performance [7] [3] higher than the average performance of the best 2% of all MLP models trained) and model simplicity (i.e., a lower number of hidden units) as proposed in [66].

For each of the 26 QC indicators, Table 1 presents the comparative cross-validation-based modeling performance of the linear and non-linear (i.e., best-performing MLP) predictive mappings considering an arbitrary 5-fold cross-validation partition. Even though the tabulated data suggests that opting for non-linear modeling can bring clear benefits, as MLPs performed (considerably) better for 19 out of 26 QC indicators, we believe that the scarcity of training data (23 samples in total) justifies the in-depth stability analysis of the tested surrogate modeling paradigms described at the end of Section 5.

Figure 6 shows the average 5-fold cross-validation R^2 values (i.e., $\mu_{5-folds}$) obtained by the linear and best-performing MLP regression models for three QC indicators (selected by experts) and $t_{cv}=25$ repeats of the 5-fold cross validation procedure, where each repeat considered a random split of the original 23 samples. Finally, for each strategy and each QC indicator, we compute the average model performance over the 25 cross-validation repeats (i.e., $R_{LIN}^2_{\mu_{t_{cv}}}$ and $R_{MLP}^2_{\mu_{t_{cv}}}$) and the jackknife bias-corrected relative standard deviation (i.e., $R_{LIN}^2_{rsd_{t_{cv}}}$ and $R_{MLP}^2_{rsd_{t_{cv}}}$). Note that the three indicators were chosen such that each of the three QC groups of interest is represented: f_1 was selected from the RMSE group, f_{12} from the flatness group and f_{18} from the FPI group. The plots in Figure 6 and the corresponding linear and MLP $R^2_{\mu_{t_{cv}}}$ and $R^2_{rsd_{t_{cv}}}$ values offer very valuable insights, as they indicate that:

Table 1: Comparison of the performance of the linear and best-performing MLP predictive mappings. Superior results are indicated in bold font and QC indicators are divided into three groups based on an initial expert-based classification.

QC Indicator	5-fold cross-validation R^2	
	Lin. reg. $\mu_{5\text{-folds}} \pm \sigma_{5\text{-folds}}$	Best-perf. MLP $\mu_{5\text{-folds}} \pm \sigma_{5\text{-folds}}$
f_1	0.581 \pm 0.317	0.795 \pm 0.092
f_2	0.517 \pm 0.114	0.759 \pm 0.177
f_3	0.863 \pm 0.080	0.879 \pm 0.061
f_4	0.882 \pm 0.060	0.913 \pm 0.080
f_5	0.786 \pm 0.118	0.742 \pm 0.336
f_6	0.922 \pm 0.065	0.874 \pm 0.093
f_7	0.887 \pm 0.061	0.829 \pm 0.100
f_8	0.920 \pm 0.038	0.896 \pm 0.089
f_9	0.645 \pm 0.062	0.744 \pm 0.184
f_{10}	0.926 \pm 0.043	0.938 \pm 0.031
f_{11}	0.895 \pm 0.066	0.841 \pm 0.097
f_{12}	0.947 \pm 0.046	0.923 \pm 0.111
f_{13}	0.870 \pm 0.079	0.827 \pm 0.094
f_{14}	0.888 \pm 0.059	0.901 \pm 0.091
f_{15}	0.580 \pm 0.284	0.793 \pm 0.124
f_{16}	0.727 \pm 0.247	0.850 \pm 0.056
f_{17}	0.732 \pm 0.115	0.854 \pm 0.086
f_{18}	0.847 \pm 0.152	0.905 \pm 0.057
f_{19}	0.656 \pm 0.227	0.845 \pm 0.121
f_{20}	0.781 \pm 0.108	0.831 \pm 0.058
f_{21}	0.371 \pm 0.633	0.863 \pm 0.100
f_{22}	0.783 \pm 0.157	0.875 \pm 0.080
f_{23}	0.620 \pm 0.368	0.817 \pm 0.070
f_{24}	0.801 \pm 0.210	0.880 \pm 0.054
f_{25}	0.618 \pm 0.415	0.849 \pm 0.137
f_{26}	0.814 \pm 0.145	0.878 \pm 0.120

- f_1 is generally difficult to model by both linear and non-linear strategies, as $R^2_{LIN-\mu_{cv}} = 0.446$ (with $R^2_{LIN-rsd_{cv}} = 26.76\%$) and $R^2_{MLP-\mu_{cv}} = 0.405$ (with $R^2_{MLP-rsd_{cv}} = 42.74\%$). Furthermore, the high relative standard deviations indicate that for this QC indicator, and especially for MLPs, 5-fold cross-validation-based R^2 estimates are highly biased by the random composition of the cross-validation partitions (i.e., they are very unstable).
- f_{12} is easier to model, as $R^2_{LIN-\mu_{cv}} = 0.925$ (with $R^2_{LIN-rsd_{cv}} = 5.05\%$) and $R^2_{MLP-\mu_{cv}} = 0.881$ (with $R^2_{MLP-\sigma^2_{cv}} = 5.26\%$). The lower relative standard deviations indicate that the estimated linear and MLP prediction accuracies are not only higher (than those of f_1), but also more robust with regard to the random composition of the cross-validation partitions.
- modeling f_{18} is similar to modeling f_{12} , as $R^2_{LIN-\mu_{cv}} = 0.874$ (with $R^2_{LIN-rsd_{cv}} = 5.80\%$) and $R^2_{MLP-\mu_{cv}} = 0.855$ (with $R^2_{MLP-rsd_{cv}} = 7.39\%$). Again, the lower variances indicate that the cross-validation-based estimated prediction accuracies are somewhat stable (especially for linear models).
- overall, usage of MLP non-linear models clearly does not seem to bring a consistent advantage for any of the investigated QC indicators.

In summary, for Scenario 1, linear models are the more reliable choice when surrogates are to be built, as they perform similarly to — or even better than — MLPs but are less complex. Furthermore, linear models are well known to operate better than non-linear ones on extrapolation cases [23].

Using the 23 expert and hybrid DoE data samples, we also constructed the cross-correlation matrix shown in Figure 7 for the 26 QC indicators of interest to our industrial partner. The cross-correlation results show that the original expert-based separation of the indicators into three main groups is highly relevant, as:

- QC indicators f_1 and f_2 are strongly intercorrelated and, from an optimization perspective, can be united in an RMSE cluster;
- QC indicators f_3 to f_{14} can form the flatness cluster;
- QC indicators f_{15} to f_{26} can form the FPI cluster.

In addition to very strong intra-cluster cross-correlations, the results from Figure 7 also indicate that there is (i) a small positive correlation between the indicators in the RMSE and flatness clusters, (ii) a small negative correlation between the QC indicators in the RMSE cluster and those in the FPI cluster, and (iii) no apparent or even very small negative correlation between flatness and FPI QC indicators.

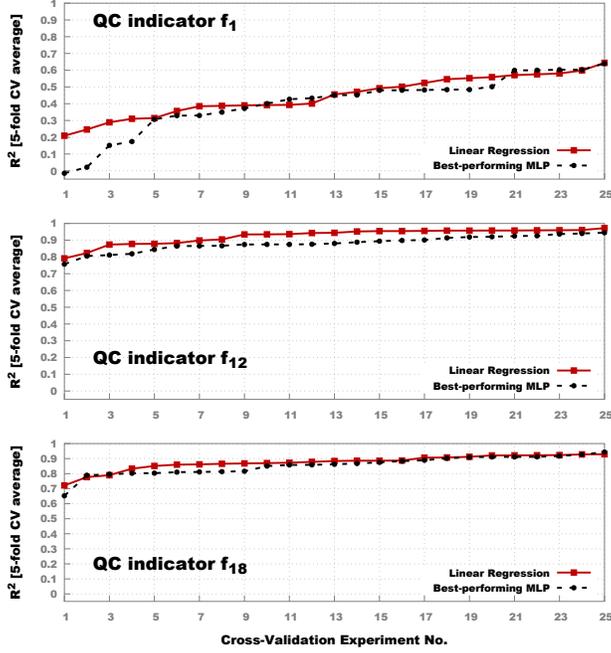


Figure 6: Performance comparison of linear and best-performing MLP models for three QC indicators over 25 different cross-validation repetitions. The 25 modeling experiments are sorted in ascending order of average R^2 over the 5 cross-validation folds.

In view of the aforementioned surrogate modeling and cross-correlation results as well as the consideration outlined in Section 6.1, we defined a surrogate-based MOOP based on linear predictive mappings that contains one representative from each QC cluster (namely, f_1, f_{12} and f_{18}). By simultaneously aiming to minimize these three QCs, we searched for bonding liner parameter settings that are expected to deliver Pareto-optimal solutions (i.e., QC values) for all 26 RMSE, flatness and FPI indicators. When defining this surrogate-based MOOP, we opted for a formulation where optimizing an objective means minimizing its absolute value towards 0. In Figure 8, which compares the final optimization results obtained by NSGA-II and DECMO2 (both parameterized with settings recommended in the literature), we report the performance in relation to f_1 by considering 4 broad quality groups (indicated by different point markers). The two Pareto fronts indicate that:

- it is not possible to optimize f_{12} and f_{18} while at the same time achieving the lowest possible value for f_1 since the individuals (i.e., BL parameter combinations) closest to the origin of the Pareto space

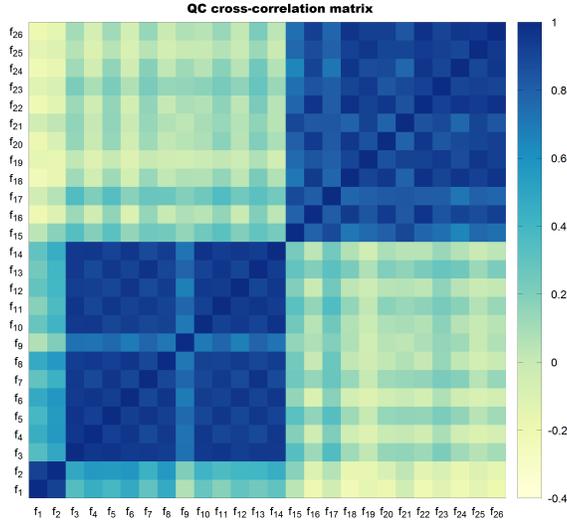


Figure 7: Cross-correlation between the 26 QC indicators. Note the three very well-defined clusters.

between f_{12} and f_{18} tend to achieve the highest values for f_1 and vice versa (the domain experts had expected this situation, as they are aware of the somewhat “hard-to-optimize” trade-off between flatness and RMSE that highlights the need for a Pareto-based solution);

- integration of a decomposition-based space exploration and PN archiving paradigm enables DECMO2 to obtain a better spread across the entire PF than NSGA-II.

The initial multi-objective optimization results have been very well received by domain experts (i.e., process operators) and our industrial partner. Therefore, some of the most promising parameter settings discovered using our methodology were finally tested at the production site. In Table 2, we present the final real QCs achieved by on-line tests at the machine (i) with the standard expert-based parameter combination for the bonding liner (first row) and (ii) with two promising parameter combinations obtained via the proposed optimization framework and selected from the Pareto fronts. Thus, compared to the expert solution, the parameter combination that yielded the QC values shown in the last row of Table 2 improved the absolute average flatness over the 12 corresponding QC indicators significantly (by around 50%) from 19.16 to 10.83 while perfectly optimizing FPI indicators (to 0) and obtaining equally good RMSE QCs. Note that the ideal value for flatness and FPI is 0, while for the RMSE indicators it is 0.007.

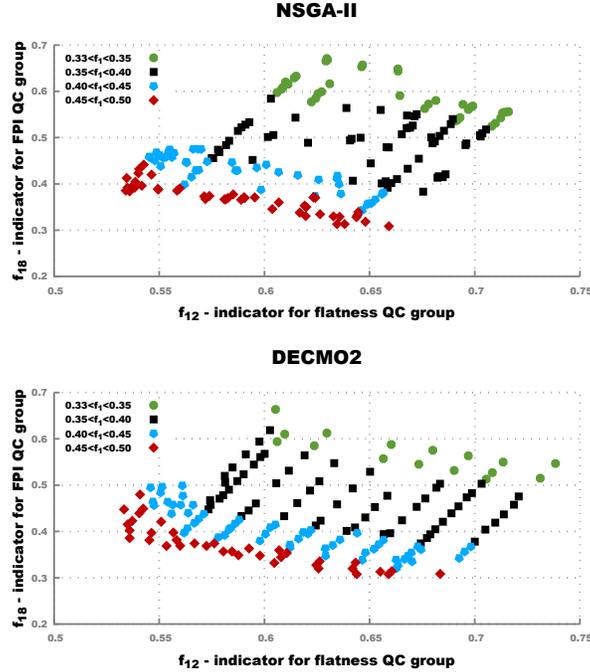


Figure 8: Comparison of the f_{12} vs. f_{18} 2D Pareto fronts obtained using NSGA-II and DECMO2 (best result out of 5 runs for each algorithm). Each point denotes an individual represented by a vector of 11 parameter values (= 1 setting which can be directly used at the BL machine by operators); thus, the operator can select a promising setting, e.g. that closest to the origin of the axes $([0.5, 0.2])$, or that with the lowest f_{18}/f_{12} value if one of these indicators seems to be more important than the other, or that with the best trade-off between the three QCs if all are equally important.

8.2. Scenario 2: Multi-stage optimization

As in the single/late-stage optimization scenario, an initial set of typically used parameter settings was suggested by the domain experts. The main difference was that the multi-stage samples contained process-wide settings (see Figure 4) that parameterize both the injection molding machine (IMM) and the bonding liner (BL). Domain experts also performed the real-life experiments and collected the QC values (i) at the end of the injection molding phase and (ii) at the end of the bonding liner phase (i.e., the end of the entire production process). The latter were essential for us in order to establish multi-stage predictive mappings that also integrate IMM parameters (as described in Section 5). The expert samples were carefully chosen according to important long-term DoE and machining plans used in the past that focused on parameters that control “dosage”, “tool temperature” and “injection speed and cooling time”. In total,

Table 2: Comparison of the QC values obtained by the standard expert-based parameter combination for the bonding liner (first row) and two parameter combinations obtained using the proposed soft-computing framework (last two rows). Absolute values closer to 0 are better; the best solution obtained is highlighted in gray background.

RMSE QCs		Flatness QCs (opt value: 0)												FPI QCs (opt value: 0)												
f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	
0.010	0.008	2	-13	11	-61	-25	3	3	-12	12	-60	-24	4	1	0	0	0	0	0	0	0	0	0	0	0	1
0.011	0.008	4	-68	-6	8	10	17	5	-67	-5	9	11	18	1	0	2	1	0	1	0	0	0	0	1	0	
0.010	0.008	-4	-18	-18	-6	-7	-15	-3	-17	-17	-5	-6	-14	0	0	0	0	0	0	0	0	0	0	0	0	0

only 20 samples (i.e., different parameter combinations) were finally obtained. Since these samples contained 14 IMM parameters (after redundancy reduction among 36 original parameters) and 14 BL parameters (3 more bonding parameters were considered in addition to the core 11 used in Scenario 1), their coverage of the parameter space was deemed very limited and highly likely to result in overfitted predictive mappings.

In order to address this problem, we scanned the production data base over a time frame of about one year to identify variations in the parameter settings used at the IMM during regular production and their resulting QC values after the BL stage. This approach was also motivated by the wish to reduce the sample gathering and evaluation efforts for experts/operators, especially since testing the parameter settings takes much longer in the delay-prone multi-stage scenario that also features an oven sub-process between the IMM and BL stages (as illustrated in Figure 4).

The historic data samples (containing both molding and bonding control parameters) used in the multi-stage optimization scenario were more heterogeneous because of the different evaluation methods required by the QC indicators. Thus, indicators that are assessed exclusively by a highly automatized process are collected relatively often (e.g., FPI indicators), while indicators that require an analysis by human experts are collected less often (e.g., RMSE indicators). Furthermore, production batches for which all indicator groups were computed were scarce. Ultimately, the input for the modeling module in the case of the multi-stage process came in the form of three independent data sets, one for each main indicator group:

Table 3: Performance comparison of linear and best-performing MLP predictive mappings for the multi-stage scenario in which QC predictions are based on a combination of 14 IMM parameters and 14 BL parameters. Superior results are in bold font.

QC Indicator	5-fold cross-validation R^2	
	Lin. reg. $\mu \pm \sigma$	MLP $\mu \pm \sigma$
$f_{RMSE-28}$	0.5024 \pm 0.1469	0.5312 \pm 0.0973
$f_{flatness-28}$	0.5108 \pm 0.1316	0.7236 \pm 0.0564
f_{FPI-28}	0.4687 \pm 0.2007	0.6359 \pm 0.0458

- the data set containing historic RMSE indicator values with 265 samples;
- the data set containing historic flatness indicator values with 317 samples;
- the data set containing historic FPI indicator values with 5453 samples;

Because of the usage of different data sets and the scarcity of common samples, a complete cross-correlation matrix between all the QC clusters could not be computed. However, the intra-cluster cross-correlation that could be computed for each data set independently remained very high and enabled use of the same implicit objective reduction strategy described in Section 8.1.

In Table 3, we present a comparison of the linear and best-performing MLP modeling results for three QC indicators selected by human experts – one for each main QC cluster: RMSE, flatness, FPI. In these tests, the goal was to create predictive mappings for each of the three indicators based on a parameter combination that is able to control the entire micro-fluidic chip production process (i.e., 14 IMM and 14 BL parameters). The overall testing methodology and best-performing MLP model selection were the same used in Section 8.1, but because of the increased sizes of the data sets and of the search space (28 parameters instead of 11 in the single/late-stage case), the best-parameter grid searches had to be adjusted in order to finish within a time frame compatible with specific industrial integration requirements. Thus, we varied the number of hidden units between $\{2, 5, 10, 25, 51, 75\}$, the learning rate between $\{0.05, 0.20, 0.35, 0.6\}$, and the momentum between $\{0.0, 0.1, 0.2, 0.4, 0.8\}$.

The results from Table 3 indicate that MLP modeling outperforms linear regression for all three QC indicators (i.e., cluster representatives) we considered, even though all 5-fold cross-validation R^2 values were

relatively poor. However, while the improvements for targets $f_{flatness-28}$ and f_{FPI-28} were noteworthy and stable across $t_{cv}=25$ repeats of the cross-validation procedure, the improvement for target $f_{RMSE-28}$ was minor and unstable (thus not marked in bold font). The surrogate MOOP that we defined for this scenario therefore used a linear predictive mapping for $f_{RMSE-28}$ and MLP-based non-linear predictive mappings for the other two objectives.

Applying the DECMO2 solver to the surrogate MOOP, we were able to discover multiple process parameter combinations that were expected to yield near-perfect results (i.e., all three QC indicator values to be minimized ≤ 0.0033). In particular, this suggested that also optimizing the IMM parameters (and not just those that control the BL, as done in the previous subsection for the single-stage process) could further improve the expected quality of the micro-fluidic chips. This was deemed plausible by the expert and finally confirmed. In fact, the expert expected much better results than in Scenario 1, as it was known that the IMM process influences final product quality. Furthermore, some of the most promising multi-stage parameter combinations suggested by the MOOP-based optimization were selected for further in-depth inspection and on-site evaluation, as they appeared realistically valuable to our industrial partner. First tests showed again improved QC values that were about 20-30% better in terms of RMSE and flatness than those obtained by optimizing only the bonding parameters.

9. Conclusion

Here, we have proposed a general optimization framework that can be used to improve (the machining parameters of) industrial production processes that are assessed using multiple quality control indicators. Its components include a new hybrid design of experiments with integration of expert knowledge through predictive mapping construction employing stability-based model selection strategies and a robust multi-objective optimization solver we developed that integrates several (co-)evolutionary strategies and archiving concepts. The application domain we examined was the production of micro-fluidic chips via a multi-stage batch process characterized by very time- and cost-intensive evaluations of tested parameter settings. Thus, our framework is centered on creating linear and non-linear predictive mappings that can be used as surrogate fitness estimators by state-of-the-art multi-objective evolutionary algorithms.

In order to demonstrate the robustness of our approach, we considered two optimization scenarios that pose different sets of challenges:

1. The single/late-stage optimization scenario was very restrictive in the number of different parameter

combinations that can be tested during optimization and showed that one key element in obtaining useful surrogate models / predictive mappings (and, consequently, very good optimization results) is the proposed two-stage data collection phase which combines expert knowledge with a novel hybrid design of experiments strategy.

2. The multi-stage optimization scenario showed that, whenever the search space is too large to perform a meaningful hybrid DoE-enhanced data collection stage, the proposed mapping-creation strategy can also be used to exploit latent expert-knowledge (stored as historic parameter combinations) by constructing useful linear and non-linear regression models from more heterogeneous data sources.

It is noteworthy that in both scenarios, the final optimization results (i.e., process-wide or machine-specific parameter combinations) led to improved product quality compared to standard expert-based parameterizations. This reduces costs for the company and improves final customer satisfaction.

It would also be important to remark that, as highlighted in Section 8, the success of the multi-objective optimization task is highly dependent on the quality of the linear and non-linear surrogate regression models which, in turn, are contingent on the availability and quality of process data, which may be time-consuming to record and collect, depending on the latency of the production process (and thus of the final product quality) with respect to changes in the process parameters. While the proposed hybrid DoE procedure is designed to steer the crucial data acquisition step in a direction to get the most significant and important parameter settings for modeling, it is usually beneficial that the initial data samples used by the hybrid DoE integrates expert input.

In the future, we firstly plan to further refine the data collection stage using surrogate-based MOEA runs to augment the current hybrid DoE strategy by suggesting a handful of parameter combinations that can simultaneously fill multiple blank spaces in the objective-wise histograms. The idea behind this is to leverage the outputs of the initial DoE samples inside a secondary step which aims to generate a few well-positioned samples that reduce modeling uncertainty across multiple objectives. Secondly, we plan to investigate the potential of applying more advanced machine learning techniques — for instance, transfer learning [60] between source and target production machines — to further reduce the time needed to train predictive mappings when a new production site is considered. This is expected to lead to an even faster on-site optimization of QC indicators as it would further reduce the need to carry out a full hybrid DoE (i.e, elicitation and integration of expert knowledge coupled with nearly no historic samples).

Acknowledgements

The authors acknowledge the support of the Austrian Research Funding Association (FFG) within the scope of the “IKT of The Future” program, project “Generating process feedback from heterogeneous data sources in quality control (mvControl)” (contract no. 849962). The first and second authors acknowledge the support of the ”LCM — K2 Center for Symbiotic Mechatronics” within the framework of the Austrian COMET-K2 program.

References

- [1] L.M. Antonio and C.A. Coello Coello. Use of cooperative coevolution for solving large scale multiobjective optimization problems. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2758–2765. IEEE, 2013.
- [2] S. Aumi, B. Corbett, and P. Mhaskary. Model predictive quality control of batch processes. In *2012 American Control Conference*, pages 5646–5651, Fairmont Queen Elizabeth, Montréal, Canada, 2012.
- [3] C. Bergmeir, R.J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70 – 83, 2018.
- [4] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time Series Analysis, Forecasting and Control*. Prentice Hall, Engelwood Cliffs, New Jersey, 1994.
- [5] M. Chandrasekaran, M Muralidhar, M.C. Krishna, and U.S. Dixit. Application of soft computing techniques in machining performance prediction and optimization: A literature review. *International Journal of Advanced Manufacturing Technology*, 46:445–464, 2010.
- [6] K. Chockalingam, N. Jawahar, K.N. Ramanathan, and P.S. Banerjee. Optimization of stereolithography process parameters for part strength using design of experiments. *The International Journal of Advanced Manufacturing Technology*, 29(1):79–88, 2006.
- [7] K.T. Chui, D.C. Lok Fung, M.D. Lytras, and T.M. Lam. Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Computers in Human Behavior*, 107:105584, 2020.

- [8] C..A Coello Coello and G.B. Lamont. Applications of multi-objective evolutionary algorithms. World Scientific, 2004.
- [9] C.A. Coello Coello, G.B. Lamont, and D.A. Van Veldhuisen. Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series. Springer, 2007.
- [10] R. Collobert and S. Bengio. SVM Torch: Support vector machines for large-scale regression problems. Journal of Machine Learning Research, 1:143–160, 2001.
- [11] D.W. Corne, N.R. Jerram, J.D. Knowles, and M.J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, GECCO’01, page 283–290, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [12] K. Deb. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, 2001.
- [13] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference—point—based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Trans. Evolutionary Computation, 18(4):577–601, 2014.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.
- [15] S. Ekwaro-Osire, A.C. Gonçsalves, and F.M. Alemanyehu. Probabilistic Prognostics and Health Management of Energy Systems. Springer, New York, 2017.
- [16] G. Franceschini and S. Macchietto. Model-based design of experiments for parameter precision: State of the art. Chemical Engineering Science, 63(19):4846–4872, 2008.
- [17] B.R. Frieden and R.A. Gatenby. Exploratory Data Analysis Using Fisher Information. Springer Verlag, New York, 2007.
- [18] V. Garcia, J.S. Sanchez, L.A. Rodríguez-Picón, L.C. Mendez-Gonzalez, and H. de J. Ochoa-Dominguez. Using regression models for predicting the product quality in a tubing extrusion process. Journal of Intelligent Manufacturing, pages <https://doi.org/10.1007/s10845-018-1418-7>, 2018.

- [19] R.P.A. Gil, Z.C. Johanyák, and T. Kovács. Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation. *Int. J. Artif. Intell.*, 16(1):20–40, 2018.
- [20] S.Y. Gu, J. Ren, and G.J. Vancso. Process optimization and empirical modeling for electrospun polyacrylonitrile (pan) nanofiber precursor of carbon nanofibers. *European polymer journal*, 41(11):2559–2568, 2005.
- [21] F. Guo, X. Zhou, J. Liu, Y. Zhang, D. Li, and H. Zhou. A reinforcement learning decision model for online process parameters optimization from offline data in injection molding. *Applied Soft Computing*, 85:105828, 2019.
- [22] F.E. Harrel. *Regression Modeling Strategies*. Springer, New York, USA, 2001.
- [23] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction - Second Edition*. Springer, New York Berlin Heidelberg, 2009.
- [24] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall Inc., 2 edition, 1999.
- [25] Y.-S. Hong, H. Lee, and M.-J. Tahk. Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks. *Engineering Optimization*, 35(1):91–102, 2003.
- [26] R.L. Iman, J.C. Helton, and J.E. Campbell. An approach to sensitivity analysis of computer models, part 1. introduction, input variable selection and preliminary variable assessment. *Journal of Quality Technology*, 13(3):174–183, 1981.
- [27] A. Jaszkievicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, 2002.
- [28] D.D Jensen and P.R. Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309–338, 2000.
- [29] D. Katherasan, J.V Elias, P. Sathiya, and A.N. Haq. Simulation and parameter optimization of flux cored arc welding using artificial neural network and particle swarm optimization algorithm. *Journal of Intelligent Manufacturing*, 25(1):67–76, 2014.

- [30] J. Knowles and D. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), volume 1, pages 98–105. IEEE, 1999.
- [31] S. Kukkonen and J. Lampinen. GDE3: The third evolution step of generalized differential evolution. In IEEE Congress on Evolutionary Computation (CEC 2005), pages 443–450. IEEE Press, 2005.
- [32] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018.
- [33] J. Levitt. Complete Guide to Preventive and Predictive Maintenance. Industrial Press Inc., New York, 2011.
- [34] E. Lughofer and M. Sayed-Mouchaweh. Predictive Maintenance in Dynamic Systems — Advanced Methods, Decision Support Tools and Real-World Applications. Springer, New York, 2019.
- [35] E. Lughofer, A.-C. Zavoianu, R. Pollak, M. Pratama, P. Meyer-Heye, H. Zörrer, C. Eitzinger, J. Haim, and T. Radauer. Self-adaptive evolving forecast models with incremental PLS space updating for on-line prediction of micro-fluidic chip quality. *Engineering Applications of Artificial Intelligence*, 68:131–151, 2018.
- [36] H.B. Mann and D.R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1):50–60, 1947.
- [37] F. Marcelloni and M. Vecchio. Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization. *Information Sciences*, 180(10):1924 – 1941, 2010. Special Issue on Intelligent Distributed Information Systems.
- [38] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics (JSTOR Abstract)*, 21(2):239–245, 1979.
- [39] K. Miettinen. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, 1999.
- [40] R.G. Miller. The jackknife — a review. *Biometrika*, 61(1):1–15, 1974.

- [41] R.K. Mobley. *An Introduction to Predictive Maintenance — Second Edition*. Elsevier Science, Woburn, Massachusetts, U.S.A., 2002.
- [42] D.C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, New York & Chichester & Brisbane & Toronto & Singapore, 1991.
- [43] Y.S. Ong, P.B. Nair, A.J. Keane, and K.W. Wong. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. In *Knowledge Incorporation in Evolutionary Computation*, pages 307–331. Springer, 2005.
- [44] L. Ouyang, C. Park, Y. Ma, Y. Ma, and M. Wang. Bayesian hierarchical modelling for process optimisation. *International Journal of Production Research*, on-line and in press(<https://doi.org/10.1080/00207543.2020.1769873>):1–21, 2020.
- [45] H. Pashazadeh, Y. Gheisari, and M. Hamed. Statistical modeling and optimization of resistance spot welding process parameters using neural networks and multi-objective genetic algorithm. *Journal of Intelligent Manufacturing*, 27(3):549–559, 2016.
- [46] E. Permin, F. Bertelsmeier, M. Blum, J. Bützler, S. Haag, S. Kuz, D. Özdemir, S. Stemmler, U. Thombansen, R. Schmitt, C. Brecher, C. Schlick, D. Abel, R. Popraw, P. Loosen, W. Schulz, and G. Schuh. Self-optimizing production systems. *Procedia CIRP*, 41:417–422, 2016.
- [47] R. Poli, W.B Langdon, N.F. McPhee, and J.R. Koza. *A field guide to genetic programming*. Lulu. com, 2008.
- [48] R. Rallo, J. Ferre-Gine, A. Arena, and F. Girault. Neural virtual sensor for the inferential prediction of product quality from process variables. *Computers and Chemical Engineering*, 26(12):1735–1754, 2004.
- [49] R. Russel Rhinehart. *Nonlinear Regression Modeling for Engineering Applications — Modeling, Model Validation, and Enabling Design of Experiments*. John Wiley & Sons, Chichester, United Kingdom, 2016.
- [50] T. Robič and B. Filipič. DEMO: Differential evolution for multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 520–533. Springer, Springer Berlin / Heidelberg, 2005.

- [51] A. Starkey, H. Hagrass, S. Shakya, and G. Owusu. A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization. *Information Sciences*, 329:390–411, 2016.
- [52] R.M. Storn and K. V. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.
- [53] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang. MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & Operations Research*, 40(6):1648–1660, 2013.
- [54] I. Tsamardinos, E. Greasidou, and G. Borboudakis. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Machine learning*, 107(12):1895–1922, 2018.
- [55] I. Tsamardinos, A. Rakhshani, and V. Lagani. Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. *International Journal on Artificial Intelligence Tools*, 24(05):1540023, 2015.
- [56] E. Weigl, W. Heidl, E. Lughofer, T. Radauer, and C. Eitzinger. On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications*, 27(1):103–127, 2016.
- [57] P.J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [58] C. Yang and J. Ding. Constrained dynamic multi-objective evolutionary optimization for operational indices of beneficiation process. *Journal of Intelligent Manufacturing*, 30:2701–2713, 2019.
- [59] Y. Yusoff, M.S. Ngadiman, and A.M. Zain. Overview of nsga-ii for optimizing machining process parameters. *Procedia Engineering*, 15:3978–3983, 2011.
- [60] W. Zellinger, B.A. Moser, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz. Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, 483:174–191, 2019.
- [61] Q. Zhang and H. Li. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

- [62] Q. Zhang, W. Liu, and H Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. Technical report, School of CS & EE, University of Essex, February 2009.
- [63] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.
- [64] A.-C. Zăvoianu, E. Lughofer, W. Amrhein, and E.P. Klement. Efficient multi-objective optimization using 2-population cooperative coevolution. In *Computer Aided Systems Theory - EUROCAST 2013, Lecture Notes in Computer Science*, pages 251–258. Springer Berlin Heidelberg, 2013.
- [65] A.-C. Zăvoianu, E. Lughofer, G. Bramerdorfer, W. Amrhein, and E.P. Klement. DECMO2: a robust hybrid and adaptive multi-objective evolutionary algorithm. *Soft Computing*, 19(12):3551–3569, 2014.
- [66] A.-C. Zăvoianu, E. Lughofer, G. Bramerdorfer, W. Amrhein, and E.P. Klement. An effective ensemble-based method for creating on-the-fly surrogate fitness functions for multi-objective evolutionary algorithms. In *Proceedings of the 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2013)*, pages 237–244. IEEE Computer Society Conference Publishing Services (CPS), 2014.
- [67] A.-C. Zăvoianu, E. Lughofer, R. Pollak, P. Meyer-Heye, C. Eitzinger, and T. Radauer. Multi-objective knowledge-based strategy for process parameter optimization in micro-fluidic chip production. In *Proceedings of the IEEE SSCI 2017 Conference*, pages 1927–1934. IEEE, Honolulu, USA, 2017.