ROBUST MULTI-OBJECTIVE OPTIMIZATION IN PRODUCT DEVELOPMENT WITH RESPECT TO USER-SCENARIO AND MANUFACTURING UNCERTAINTIES

Wolniak, Philipp;
Cramer, Jakob;
Lachmayer, Roland

Leibniz University Hannover

ABSTRACT

In product development, user-scenarios are a way of tailoring requirements to defined customer groups. Furthermore, a product design often involves multiple conflicting objectives that are analyzed within an iterative process. The models typically used for the analysis often do not accurately reflect the real-world representation. This can be alleviated by finding robust product designs. While usually uncertainties due to manufacturing tolerances are investigated, we additionally consider uncertainties in the user-scenario. Therefore, we present a robustness evaluation in a multi-objective numerical optimization in product development. For this, we consider manufacturing tolerances using an adjusted Latin Hypercube Sampling as well as deviations in the user-scenario by means of a Gaussian distribution. In the case study, we present the robust development of a customer specific coffee machine, where we show the robustness evaluation and the impact of the proposed adjustments. The advantage of the presented process is a product design tailored to the customer's requirements under specified uncertainties. In addition, this enables a time benefit in the product development due to the automated analysis used in the optimization.

Keywords: Robust design, Multi-objective optimization, User-scenario, Uncertainty, Computational design methods

Contact:
Wolniak, Philipp
Leibniz University Hannover
Institute of Product Development
Germany
wolniak@ipeg.uni-hannover.de

Cite this article: Wolniak, P., Cramer, J., Lachmayer, R. (2021) ‘Robust Multi-Objective Optimization in Product Development with Respect to User-Scenario and Manufacturing Uncertainties’, in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.520

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

1 INTRODUCTION

Product developments are driven by the fulfillment of a set of requirements. Typically, market analyses and the anticipation of the customer's needs are taken as bases for the definition of these requirements. As the market and the customer's requirements may be inhomogeneous, the result are often broad requirements, either geometrically and/or functionally oversizing the product or excluding certain customer groups. To find a case-specific optimal solution a possibility is to employ user-scenarios as a basis (Filippidou, 1998). With this, a new variant of a given design concept can be tailored to the specific requirements of a customer. However, this also imposes a more dynamic and flexible necessary reaction to new customer needs (Meier *et al*., 2010; Schreiber *et al*., 2018), leading to a high spectrum of necessary evaluations of user-scenarios and design variants.
In product development these evaluations are typically performed in an iterative process. Thereby, a set of characteristics of the potential solution is analyzed and the properties are compared to the requirements. This process is repeated until a suitable match for the requirements is found (Wynn and Eckert, 2017). As even small geometric or functional changes can have a strong influence on these properties, this design task can lead to a high number of necessary model changes and analysis routines. Due to this high amount of iterations an automation of this task represents a useful approach. Especially the use of numerical optimization offers the possibility of an efficient solution space exploration (Renzi *et al*., 2017; Riesenfeld *et al*., 2015; Lachmayer *et al*., 2017).
In engineering design processes, several, mostly conflicting objectives might be pursued simultaneously like high quality products at low costs in product development. In numerical optimization this is considered by the use of multi-objective optimization, where several objective functions are taken into account, leading to the decision of the best compromise for the given design task. However, in a model-based realization of engineered systems, model representations are typically incomplete and inaccurate. The physical world is not represented completely and accurately, making it almost impossible to predict the future state exactly (Riesenfeld *et al*., 2015). Nevertheless, by identifying robust solutions it is possible to work around these limitations. These solutions are relatively insensitive to inaccuracies or uncertainties embodied in the analysis models or the physical representations; see (Deb and Gupta, 2005).
Dealing with these uncertainties can be performed on the one hand by methods of decision making and probabilistic reasoning aiming at assessing the uncertainties prior to the development, ensuring a product capable of withstanding these uncertainties (Chalupnik *et al*., 2009; Jensen and Nielsen, 2007). This represents an explicit form of uncertainty modeling by using probabilistic values for the evaluation of the most likely scenario. On the other hand and contrary to this explicit approach, in robust numerical optimization an implicit definition of uncertainties is used. Here, a given design concept and its corresponding models are used. Rather than implementing probabilistic values, the inputs of the analysis models are varied according to the uncertainties, showing the resulting deviation of the objective and therefore the influence of the uncertainties.
The use of this form of implicit robust numerical optimization in product development regarding manufacturing uncertainties like geometrical tolerances is a topic broadly applied and described in literature (Heling *et al*., 2018; Lee and Park, 2001). However, the defined user-scenario represents another source of uncertainties, influencing the outcome of the product.
To alleviate this, we present an applied example of the robust assessment of solutions within a multi-objective optimization, where we combine the assessment of manufacturing uncertainties, as well as user-scenario uncertainties. For this task, we firstly use the well-known method of Latin Hypercube Sampling (LHS) and accelerate the evaluation process by adding some minor extensions. Secondly, we consider the influence of uncertainties regarding the user-scenarios by adding a Gaussian distribution to simulate a user deviation. As a case study we use the multi-objective optimization of a coffee-machine as presented in (Wolniak *et al*., 2020b). The basis for the optimization is a user-scenario given by a customer with timely defined demands of coffee. The optimization problem is to find robust solution alternatives of a machine with the best trade-off between serving-efficiency and water-use.

2 STATE OF THE ART

Multi-objective optimization consists of a design variable space, representing different system compositions, as well as an objective function space where these different compositions are rated by their objectives. The values of the design variables are varied by the optimization algorithm to obtain

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

the values of the objective functions. Due to the often conflicting objectives in a multi-objective optimization, it is generally not possible to generate one optimal solution, which is why the concept of Pareto optimality is introduced. Pareto optimal solutions are non-dominated to each other and no other solution in the objective function space dominates these solutions. Therefore, the set of Pareto optimal points represents the set of the best compromises and is called Pareto front (Deb and Gupta, 2005).
Furthermore, robustness in optimization processes is based on the search for a solution, which at the same time shows an insensitivity against uncertainties. These uncertainties result from changes in real-world applications or inaccuracies in analysis models. Therefore, from (He *et al.*, 2020) and (Gunawan and Azarm, 2005) the general robust and multi-objective optimization problem, is given by:

$$\text{minimize } x \qquad F(x) = (f_1(x', p'), \ldots, f_m(x', p'))^T \tag{1}$$

$$\text{with} \qquad x' = (x_1 + \delta_1, \ldots, x_n + \delta_n)^T \;;\; p' = (p_1 + \gamma_1, \ldots, p_k + \gamma_k)^T \qquad (2);(3)$$

$$\text{subject to} \qquad x \in \Omega.$$

The vector F includes the m number of objective functions, x is the vector of the design variables, which belong to the design variable space Ω and p is the vector of design parameters. In contrast to the design variables, the design parameters are static values throughout the optimization. The disturbance vector γ is added to the design parameters and the disturbance vector δ is added to the design variables, respectively representing the uncertainties. Every entry of the disturbance vector is defined as a region, e.g. $-\delta_i^{max} \le \delta_i \le \delta_i^{max}$ with $i = 1, \ldots, n$. Accordingly, there is an area around each design point in the design variable space, which is defined as the amount of all possible variations of x taking into account the uncertainties. The same conditions also apply to the design parameters. This uncertainty area is also reflected in the objective function space with its size depending on how robust the point is. To illustrate this, an example is given in Figure 1, where point B is said to be more robust than point A. The reason for this is, that the uncertainties in the design variable space in Figure 1a create a smaller area in the objective function space in Figure 1b for point B than point A, although the deviation in the design variable space is identical. Furthermore, it can be seen that the generated area in the objective function space for point C includes a gap. On the one hand, these gaps can be of natural origin, if no combination of the design variables result in this area of the objective function space. On the other hand, gaps can occur if combinations of design variables create solutions that are not valid due to constraints included in the optimization run.

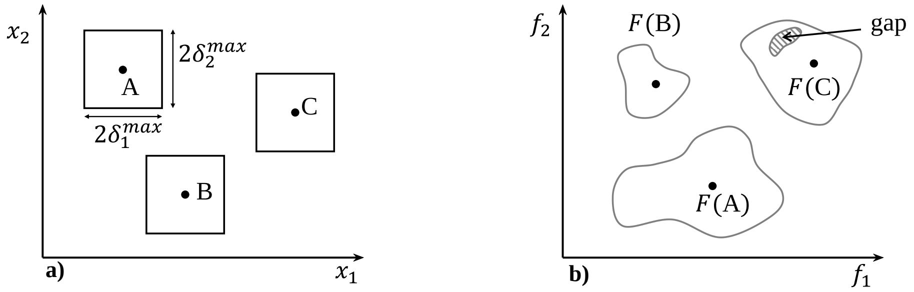

Figure 1. a) Deviation in design variable space.; b) Deviation in objective function space.

Mainly in robust optimization, it is necessary not only to evaluate single points from the design variable space, but also to test points within the uncertainty area. Therefore, sampling methods from the statistical design of experiments can be used to create these samples in a defined area. Next to the straight-forward approach of creating samples randomly, there are other possibilities like LHS (Mckay *et al.*, 2000). This method divides the design variable space into an equal-sized grid with n grids in every dimension, where n also represents the number of samples. The samples in LHS are then distributed across the grid, so that each row and each column contain exactly one sample. This technique provides a better covering of the design variable space, which is why less samples are needed for a robustness estimation. In the work of (Alinejad and Botto, 2019) LHS is used to find the non-feasible areas in the objective function space, caused by parametric models in an optimization application. Other examples for LHS in robust optimization processes can be seen in (Venanzi and Materazzi, 2013) and (Chang *et al.*, 1999).

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

A different approach to create robust and optimized results is based on creating mean values of the objective function by evaluating the initial point and samples from the neighborhood, see (Branke, 1998; Deb and Gupta, 2005). To reduce the amount of used samples, other possibilities like reevaluating only the best samples and including points from the history instead of sampling are presented in (Branke, 1998).

Other works use meta-models to approximate the objective function space using only a few points and samples. With this method, the often computationally-intensive evaluations of the objective functions can be minimized. In (Paenke *et al.*, 2006) different meta-modeling techniques are introduced for creating mean values of the fitness evaluation. For more examples of meta-modeling in robust optimization refer to (Chen *et al.*, 1996), (Park and Lee, 2005) and (Jin *et al.*, 2003).

The work of (Gunawan and Azarm, 2005) presents another approach, using sensitivity regions in the design variable space resulting from the defined robust area in the objective function space. If the so-called worst-case sensitivity (the smallest distance to the non-robust area) in the design variable space is bigger than the predefined uncertainty area, a point is defined as robust. This worst-case sensitivity in this scenario can be calculated using any single-objective and deterministic optimization algorithm, but it implies an objective function space without gaps. The approach proposed in (He *et al.*, 2020) separates the processes of multi-objective optimization and robustness verification in two steps. For this the design variable space gets subdivided into hyperboxes and the objective function space gets divided by vectors, where every point from the objective function space is assigned to the closest vector. Then the robust areas for every vector can be determined by sampling and the robust Pareto front results from the best and robust points from each vector.

For the applied use case of robust optimization in this paper, a solution space with gaps must be assumed, due to the structure of the problem. Therefore, robust algorithms, that assume an area in the objective function space without gaps cannot be used in this case. Examples for such algorithms are the presented meta-models and sensitivity regions. This is why an adapted LHS is used for the presented application.

3 METHOD DESCRIPTION

The proposed method of robustness evaluation is essentially based on three steps (Figure 2). In step one, a multi-objective optimization is performed with the aim of creating a set of points in the objective function space, including a set of Pareto optimal points. By using the most promising points from this initial optimization the robustness evaluation does not have to be performed on all points (Branke, 1998). During the next step, the selected points are checked for their robustness properties with respect to manufacturing tolerances. Following, again only the best points are passed on to step three, which further reduces the number of evaluations during this last step. In this step the transferred points are checked for robustness properties regarding user deviations. As a result, the remaining points inside the objective function space after the robustness evaluation are both robust against manufacturing tolerances and user deviations.

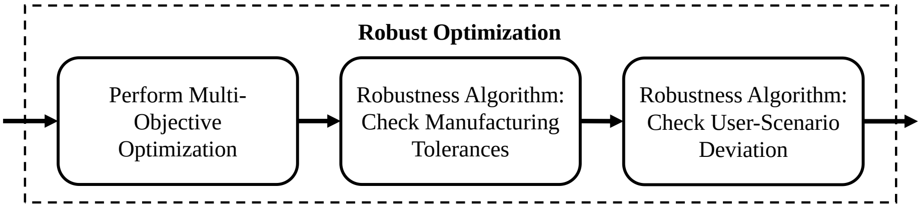
Robust Optimization
Perform Multi-Objective Optimization
Robustness Algorithm: Check Manufacturing Tolerances
Robustness Algorithm: Check User-Scenario Deviation

Figure 2. Robustness evaluation process.

As the three parts of this method are separate and the robustness check can thus be combined with any multi-objective optimization, this enables a high degree of application flexibility.

3.1 Robustness Definition

In order to determine robust points, certain specifications have to be defined. Firstly, the uncertainty areas for every input variable $\delta_1^{max},\dots,\delta_n^{max}$ have to be set. These uncertainties can be derived e.g. from common manufacturing tolerances for this kind of process. Then, the maximum permissible deviation region for every objective function $\Delta f_1^{max},\dots,\Delta f_m^{max}$ in the objective function space must be defined. As

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

already described, points in the design variable space with the same uncertainty area can cause different deviations in the objective function space. The robustness of a point x_0 can now be tested by checking whether the generated deviation in the objective function space is within the previously defined range. Figure 3 shows an example for a robust point x_0 since the generated deviation in the objective function space is within the robust region.

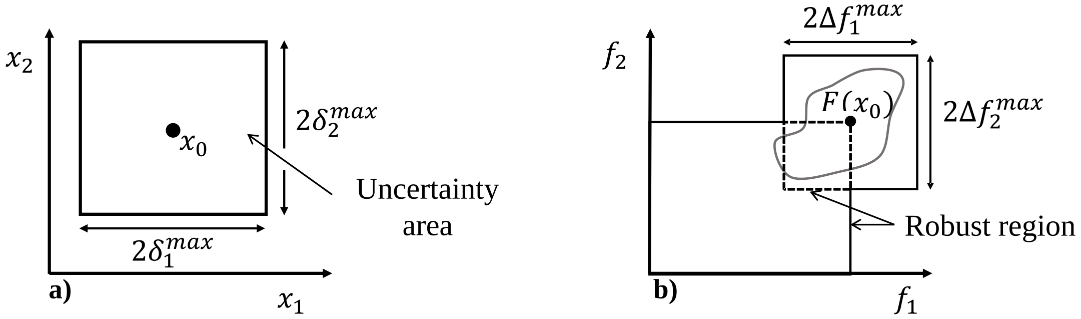

Figure 3. Robustness definition; a) in design variable space; b) in objective function space

In this paper, a further extension of the robust region is defined as a composition of two areas. The first area is defined by the input of the maximal deviation for every objective function, creating a rectangle around the point in the objective function space. The second area results from the consideration, that solution points, which improve for at least one objective function and do not worsen regarding any of the other objective functions offer no disadvantage for the customer. Therefore, points which are out of the maximal deviation range but have better properties are also considered as robust points.

3.2 Algorithm for robustness against manufacturing tolerances

The main goal of the algorithm for determining manufacturing robustness is to check whether uncertainties in the design variable space lead to solutions in the objective function space within a predefined robust region. The algorithm presented in this paper uses LHS to create samples in the uncertainty area in the design variable space around the data point to be evaluated. Subsequently, it is verified sample by sample if the evaluated samples are part of the predefined robust region in the objective function space. Once a sample is outside the specified robust region, the current point is declared as not robust and is removed from the data set. Contrary, a point is said to be robust, if every LHS sample from the design variable space is part of the robust region in the objective function space. Even though sampling is not the most efficient method, since every sample needs to be evaluated by calculating the objective function values, it ensures that even small discontinuities in the objective function space can be detected. However, it is important to use a sufficient number of samples to achieve a solution convergence.

To accelerate the process three extensions are added in comparison to typical sampling methods. Firstly, because of the separate consideration of optimization and robustness check, a predefined quantity of the most promising points from the optimization results are transferred to the manufacturing robustness evaluation. For this, the results in the objective function space of the multi-objective optimization are clustered into different sets, using the non-dominated sorting from (Deb *et al.*, 2002), illustrated in Figure 4a. In this process the Pareto optimal points, get defined as the first set. Then this set is removed and a new Pareto front gets formed from the remaining points, creating set number two, and so on. According to the amount of selected sets by the designer the number of robustness evaluations is reduced.

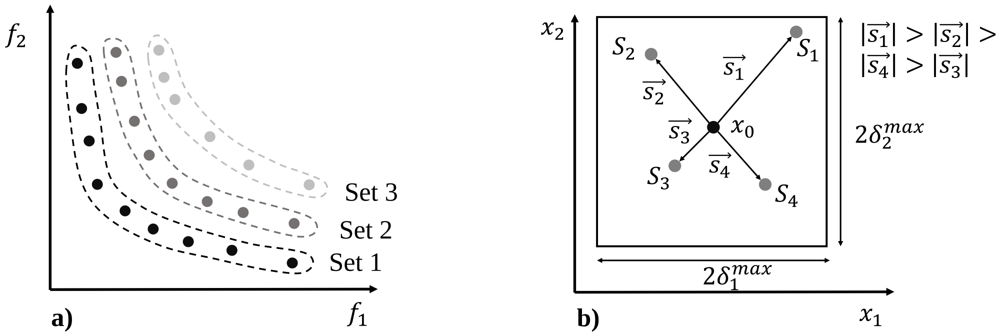

Figure 4. a) Non-dominated sorting. b) Sample distance sorting.

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

The second extension of the algorithm is based on the assumption, that samples within the uncertainty area with a higher uncertainty, i.e. samples which are further away from the initial point x_0 in the design variable space, are less likely to be robust samples. In manufacturing processes, this results in a larger exploitation of tolerances and thus a larger deviation from the ideal value. Hence, prior to the objective function calculation in the robustness evaluation, the samples are sorted by the Euclidean distance from far to near, in regard to the initial point x_0. As a result, samples with the largest Euclidean distance are evaluated first, increasing the possibility of finding non-robust samples earlier and terminating further sample evaluations. Figure 4b gives an example for this type of sorting for four samples S_1-S_4 in the uncertainty area around x_0. After the distances $\left|\vec{s_1}\right|$-$\left|\vec{s_4}\right|$ have been calculated and sorted from high to low, the processing of the samples results in the order $S_1 \rightarrow S_2 \rightarrow S_4 \rightarrow S_3$.

The third extension refers to the evaluation of every point by two sampling loops. The first loop contains only a few samples generated by LHS for a quick check of the whole uncertainty area. This way large gaps and non-robust areas can be detected. Every point passing this loop is transferred to the next loop, where a detailed check with a higher number of samples is conducted. The effect of the extensions is shown in the case study in section 4. Additionally, Figure 5 gives an overview of the algorithm in the form of a UML diagram.

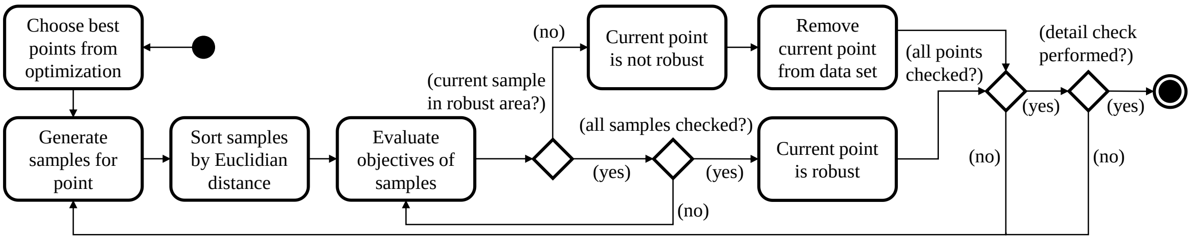
Choose best points from optimization
Generate samples for point
Sort samples by Euclidian distance
Evaluate objectives of samples
(current sample in robust area?)
(no)
Current point is not robust
Remove current point from data set
(all points checked?)
(detail check performed?)
(all samples checked?)
(yes)
(yes)
(no)
Current point is robust
(no)
(yes)
(yes)
(no)

Figure 5: UML diagram of the algorithm for manufacturing-robustness.

3.3 Algorithm for robustness against user-scenario deviation

When user-scenarios are used for a product optimization process the optimized system is based on an assumption of use or on experience data. An example of such a user-scenario is a distribution of events throughout a time period, as shown in the initial peak assumption in Figure 6b. Optimizing a product for one user-scenario assumption is insecure, as the user distribution may vary, possibly leading to high changes in the objective functions. For this reason, samples have to be created where the user-scenarios change randomly, so that the system is checked for robustness against such changes.

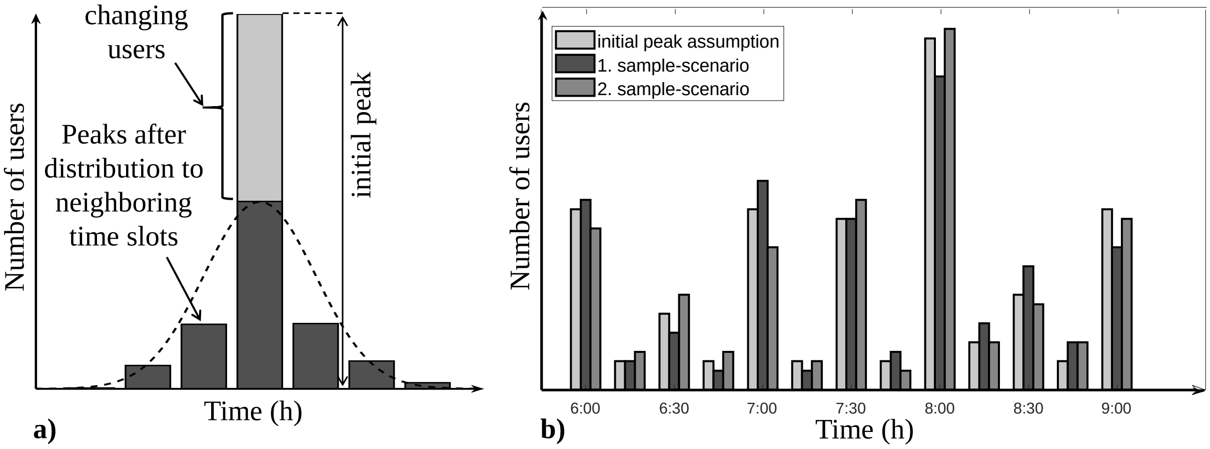

Figure 6. a) Distribution of users to neighboring time slots using Gauss-distribution.
b) Example of an initial scenario assumption with two sample-scenarios.

For the algorithm, the extent of uncertainty can be chosen by defining the percentage on how many users of the total number of users will change their time slot. For the algorithm it is assumed, that users will not change their time slots in a completely random way but are more likely to change around their initial time slot. For this, every time slot is varied using a Gaussian-distribution with a mean at the peak of interest and a standard deviation of one time slot. As a result, the probability for a user to change one,

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

two or three time slots forward or backward is approximately 68%, 27% and 5% (Figure 6a). In addition, two restrictions for this process have been added. The first restriction ensures that at most, only the number of users of the initial peak can change their time slot, so that no negative peaks occur. The second restriction refers to the border peaks, where the users can only change in one direction away from the outer border. Figure 6b shows an example of two variations of the initial user-scenario, with a defined change-percentage of 20% of the users.

This distribution randomly generates differing user-scenarios, representing the uncertainty of the user distribution. The number of necessary differing user-scenarios to be checked for one point has to be high enough to obtain convergence. An example of the convergence behavior is given in the case study. Figure 7 gives an overview of the process of creating a sample of the user scenario with uncertainty in the form of a UML diagram.

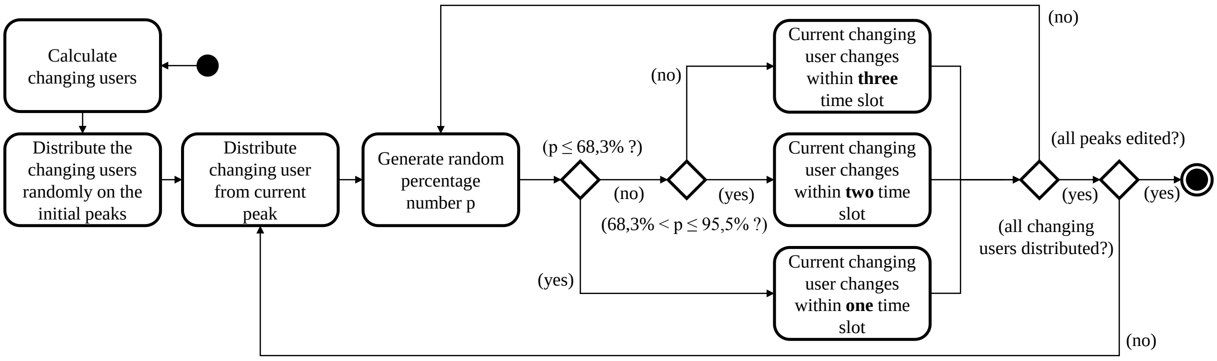
Calculate changing users
Distribute the changing users randomly on the initial peaks
Distribute changing user from current peak
Generate random percentage number p
(p ≤ 68,3% ?)
(no)
(yes)
(no)
(yes)
(68,3% < p ≤ 95,5% ?)
Current changing user changes within **three** time slot
Current changing user changes within **two** time slot
Current changing user changes within **one** time slot
(no)
(all peaks edited?)
(yes)
(yes)
(all changing users distributed?)
(no)

Figure 7: UML diagram for creating one sample of the user-scenario under uncertainty.

4 CASE STUDY

The case study is performed on the development of a robust design of a customer-specific coffee machine, optimized regarding the operating time and the water consumption. The basis for the optimization is a user-scenario of hotel guests ordering cups of coffee in a three-hour breakfast time period. This scenario is shown schematically in the initial peak assumption in Figure 6b. The analysis is performed using a functional model, calculating the brewing process as well as the transfer of the coffee from a brewing unit to a buffer tank and finally to a tap element. Additionally, a parametric geometry model in the form of a generative-design-approach (GDA) model (Brockmöller *et al.*, 2020) is used to evaluate all design variable changes from the optimization algorithm, which have an influence on the physical representation of the coffee machine. This physical representation is used as a restriction check for the overall size of the machine and the installation space of the components. For a more detailed description refer to (Wolniak *et al.*, 2020b; Wolniak *et al.*, 2020a).

The robustness region in the objective function space is defined by ± 1 liter for the water consumption and ± 15 minutes for the operating time. Another uncertainty considered is a deviation in the distribution of the customers in the user-scenario of 15%. The multi-objective optimization is performed using the Non-dominated Sorting Genetic Algorithm-II (NSGA2) from (Deb *et al.*, 2002) with a defined maximum number of 3000 evaluations. For the manufacturing robustness the best five sets of the resulting optimal points are chosen. The deviation of the design variables is set to ±0.5 mm as continuously changing geometric values of the brewing unit and the buffer tank. Additionally, an uncertainty of the water pump power is set to ±5%. For the LHS, initially the relatively high number of 500 samples for every point are used to obtain convergence in the manufacturing robustness and 100 samples are used for convergence in the user-scenario robustness. Moreover, two sampling loops are used in the manufacturing robustness, whereby the first loop contains 25 samples for the quick check and the second loop contains 475 samples. The impact of the extensions in the robustness algorithms on this high number of initial samples is presented in the following.

For a detailed overview, the results of the whole robust optimization process for the exemplary user-scenario are divided according to the proposed steps in Figure 2. Figure 8a shows the 2480 valid points from the multi-objective optimization after excluding the points violating the geometrical restrictions imposed by the GDA model. The best five sets for the robustness evaluation result in 347 points for the following process. In Figure 8b, the results of the algorithm for robustness against manufacturing tolerances is displayed. Here, 46 points showed a robust behavior according to the previously defined

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

robust region. Especially in three areas robust points can be detected: An area with low operating time and high water consumption, an area with high operating time and low water consumption and an area in between, containing solutions that offer a good compromise between the objectives. Finally, the points from the second step are passed to the robustness algorithm for user-scenario deviation, showing that 7 of these points additionally have robust behavior (Figure 8c). Thus, these points fulfill all robustness requirements of the considered case study and can therefore be defined as robust designs (Figure 8d).

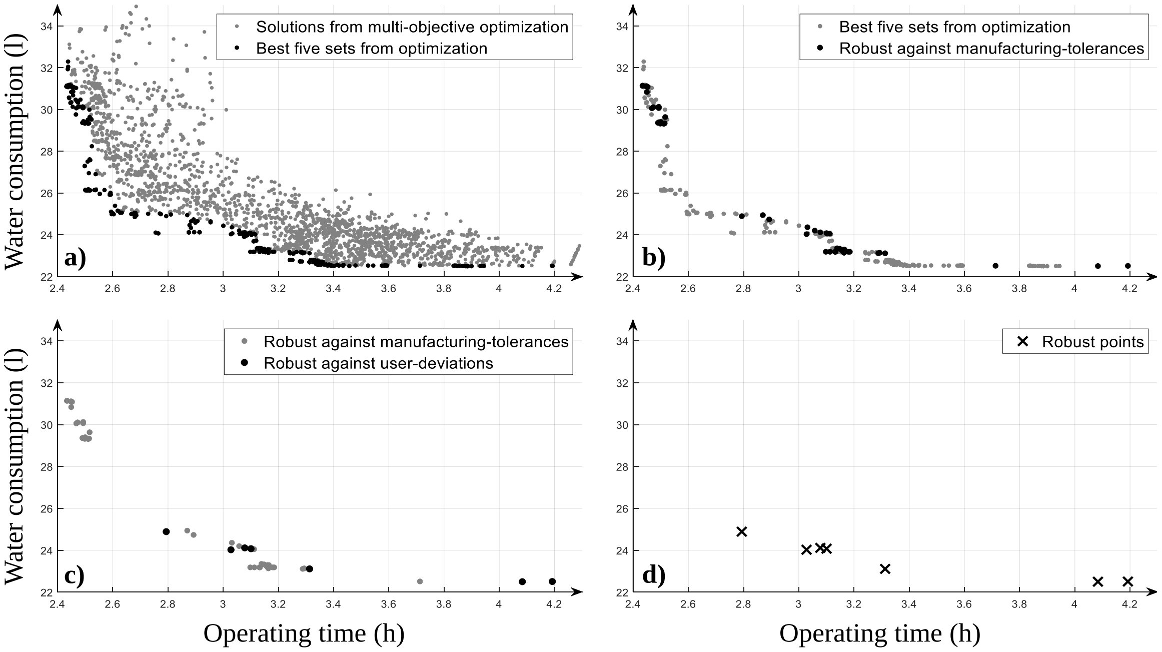

Figure 8: Results from the robust optimization process.

The process shows that with every step more points are removed from the data set, resulting in only a few points which satisfy all robustness requirements. However, due to the two robust areas in the objective function space a coffee machine with low water consumption and high operating time or a solution with a good compromise can be chosen, according to the preferences of the customer. For the case that no satisfying robust solution can be found two possibilities arise. Firstly, more sets can be evaluated to check, if there are robust solutions in a worse area of the multi-objective optimization results. Secondly, a readjustment of the requirements can be considered, by enlarging the robust regions of the objective functions.

This case study also demonstrates the time saving effect of the sample sorting before the objective function evaluation and using two sampling loops with the underlying idea of a first quick robustness check for manufacturing tolerances. Table 1 shows the percentage of the eliminated non-robust points in conjunction with the number of samples to detect these non-robust points for different sampling variations.

Table 1. Effect of different sampling variations in manufacturing robustness.

Needed number of samples	Percentage of removed non-robust points		
	one loop, no sorting	two loops, no sorting	two loops with sorting
1	28.6	29.2	34.5
5	67.1	67.4	78.4
20	85.7	89.0	95.0
100	95.4	96.7	98.0

It can be noted, that the use of two sampling loops with a first quick and a second detailed check increases the percentage of removed non-robust points during the first 20 samples slightly by 3.3%. A greater effect can be seen when combining two sampling loops and an additional sorting. After 20 from a total number of 500 sample evaluations, 95% of the non-robust points are detected and removed from the data set. Consequently, in 95% of the non-robust points, 480 or more sample evaluations are saved. This also displays almost a 10% increase compared to the method with only one sampling loop without sorting. In addition, due to the sorting, almost 6% more points are removed after the first sample. Since

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

after 100 samples still not all non-robust points are found, this is an indication of the difficult objective function space resulting from the gaps due to restrictions.

The resulting effect of changing requirements for more robust solutions can be taken from Table 2, where the variation of the percentage of changing users in the user-scenario robustness check is exemplarily listed.

Table 2. Effect of changing requirements on the number of robust solution points.

Percentage of changing users	Number of robust points
0	46
5	11
10	8
20	5

It can be seen, that as expected, a higher percentage of changing users, representing a higher uncertainty, results in less robust points. Furthermore it is noticeable that many points are already eliminated at a low percentage of changing users. This means that these coffee machine configurations are very sensitive to changing user-scenarios.

5 CONCLUSION AND OUTLOOK

In this paper we present an applied example of a robust multi-objective numerical optimization in the context of a user-scenario as the basis for the product development. The robustness evaluation includes the impact of manufacturing uncertainties as well as uncertainties in the user-scenario. Typically, due to discrete parameter changes or restrictions there is a discontinuous solution space in product development. To cope with this difficulty, the robustness evaluation is performed using the well-known method of Latin Hypercube Sampling (LHS). The often necessary high number of samples is alleviated by adding extensions considering the number and order of the objective function evaluations. In the case study of the robust design optimization of a coffee machine we could show, that these extensions reduce the number of necessary samples for convergence from 500 initial samples to 100 or less samples in 98% of the cases. Additionally, we propose a second robustness evaluation for the user-scenario uncertainties. By using a Gaussian distribution we could show the possibility of creating differing user-scenarios according to a predefined uncertainty. With this, starting from a high number of 3000 evaluations a total of 7 robust and Pareto optimal points were found in the case study.

The robustness assessment is executed as a serial process, which makes it possible to use any optimization algorithm as a basis and to perform the individual assessments independently. However, it has to be noted, that for time intensive simulations the number of samples due to LHS may be very high. In future work more case studies have to be investigated to further analyze the impact of the implemented extensions on different types of solution spaces.

REFERENCES

Alinejad, F. and Botto, D. (2019), “Innovative adaptive penalty in surrogate-assisted robust optimization of blade attachments”, *Acta Mechanica*, Vol. 230 No. 8, pp. 2735–2750.

Branke, J. (1998), “Creating robust solutions by means of evolutionary algorithms”, in Eiben, A.E. (Ed.), *Parallel problem solving from nature: 5th international conference, Amsterdam, The Netherlands, September 27 - 30, 1998 ; proceedings*, *Lecture Notes in Computer Science*, Vol. 1498, Springer, Berlin, pp. 119–128.

Brockmöller, T., Li, H., Gembarski, P.C. and Lachmayer, R. (2020), “An Investigation of a Generative Parametric Design Approach for a Robust Solution Development”, *Proceedings of the Design Society: DESIGN Conference*, pp. 315–324.

Chalupnik, M.J., Wynn, D.C. and Clarkson, P.J. (2009), “Approaches to Mitigate the Impact of Uncertainty in Development Processes”, *DS 58-1*: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 1, Design Processes, Palo Alto, CA, USA, 24.-27.08.2009, pp. 459–470.

Chang, P.B., Williams, B.J., Santner, T.J., Notz, W.I. and Bartel, D.L. (1999), “Robust optimization of total joint replacements incorporating environmental variables”, *Journal of biomechanical engineering*, Vol. 121 No. 3, pp. 304–310.

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press

Chen, W., Allen, J.K., Tsui, K.-L. and Mistree, F. (1996), "A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors", *Journal of Mechanical Design*, Vol. 118 No. 4, pp. 478–485.

Deb, K. and Gupta, H. (2005), "Searching for Robust Pareto-Optimal Solutions in Multi-objective Optimization", in Coello Coello, C.A., Hernández Aguirre, A. and Zitzler, E. (Eds.), *Evolutionary multi-criterion optimization: Third international conference, EMO 2005, Guanajuato, Mexico, March 9 - 11, 2005 ; proceedings*, *Lecture Notes in Computer Science*, Vol. 3410, Springer, Berlin, pp. 150–164.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6 No. 2, pp. 182–197.

Filippidou, D. (1998), "Designing with scenarios: A critical review of current research and practice", *Requirements Engineering*, Vol. 3 No. 1, pp. 1–22.

Gunawan, S. and Azarm, S. (2005), "Multi-objective robust optimization using a sensitivity region concept", *Structural and Multidisciplinary Optimization*, Vol. 29 No. 1, pp. 50–60.

He, Z., Yen, G.G. and Lv, J. (2020), "Evolutionary Multiobjective Optimization With Robustness Enhancement", *IEEE Transactions on Evolutionary Computation*, Vol. 24 No. 3, pp. 494–507.

Heling, B., Schleich, B. and Wartzack, S. (2018), "Robust-Design-Optimization of mechanisms based on kinematic requirements considering uncertainties", *Procedia CIRP - 15th CIRP Conference on Computer Aided Tolerancing – CIRP CAT 2018*, Vol. 75, pp. 27–32.

Jensen, F.V. and Nielsen, T.D. (2007), *Bayesian networks and decision graphs*, *Information Science and Statistics*, Softcover reprint of the hardcover 2nd ed. 2007, Springer, Berlin.

Jin, R., Du, X. and Chen, W. (2003), "The use of metamodeling techniques for optimization under uncertainty", *Structural and Multidisciplinary Optimization*, Vol. 25 No. 2, pp. 99–116.

Lachmayer, R., Mozgova, I. and Da Silva de Siqueira, R. (2017), "Development of a Topology Optimization Method for Tailored Forming Multi-material Design", in *Procceedings of the 24th ABCM International Congress of Mechanical Engineering, 12/03/2017*, ABCM.

Lee, K.-H. and Park, G.-J. (2001), "Robust optimization considering tolerances of design variables", *Computers & Structures*, Vol. 79 No. 1, pp. 77–86.

Mckay, M.D., Beckman, R.J. and Conover, W.J. (2000), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code", *Technometrics*, Vol. 42 No. 1, pp. 55–61.

Meier, H., Roy, R. and Seliger, G. (2010), "Industrial Product-Service Systems—IPS 2", *CIRP Annals*, Vol. 59 No. 2, pp. 607–627.

Paenke, I., Branke, J. and Jin, Y. (2006), "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation", *IEEE Transactions on Evolutionary Computation*, Vol. 10 No. 4, pp. 405–420.

Park, G.-J. and Lee, K.-H. (2005), "A Global Robust Optimization Using the Kriging Based Approximation Model", *Transactions of the Korean Society of Mechanical Engineers A*, Vol. 29 No. 9, pp. 1243–1252.

Renzi, C., Leali, F. and Di Angelo, L. (2017), "A review on decision-making methods in engineering design for the automotive industry", *Journal of Engineering Design*, Vol. 28 No. 2, pp. 118–143.

Riesenfeld, R.F., Haimes, R. and Cohen, E. (2015), "Initiating a CAD renaissance: Multidisciplinary analysis driven design", *Computer Methods in Applied Mechanics and Engineering*, Vol. 284, pp. 1054–1072.

Schreiber, D., Gembarski, P.C. and Lachmayer, R. (2018), "Developing a Constraint-Based Solution Space for Product Service Systems", *Proceedings of the 8th* International Conference on Mass Customization and Personalization - Community of Europe (MCP-CE 2020).

Venanzi, I. and Materazzi, A.L. (2013), "Robust optimization of a hybrid control system for wind-exposed tall buildings with uncertain mass distribution", *Smart Structures and Systems*, Vol. 12 No. 6, pp. 641–659.

Wolniak, P., Kloock-Schreiber, D., Sauthoff, B. and Lachmayer, R. (2020a), "Integrating Architectural Design Changes in Computer-Aided Design Optimization", *Proceedings of the 9* International Conference on Mass Customization and Personalization - Community of Europe (MCP-CE 2020).

Wolniak, P., Sauthoff, B., Kloock-Schreiber, D. and Lachmayer, R. (2020b), "Automated Product Functionality and Design Optimization Instancing a Product-Service System", *Proceedings of the Design Society: DESIGN Conference*, Vol. 1, pp. 1405–1414.

Wynn, D.C. and Eckert, C.M. (2017), "Perspectives on iteration in design and development", *Research in Engineering Design*, Vol. 28 No. 2, pp. 153–184.

https://doi.org/10.1017/pds.2021.520 Published online by Cambridge University Press