

CONFIGURING CUSTOMIZED PRODUCTS WITH DESIGN OPTIMIZATION AND VALUE-DRIVEN DESIGN

Vidner, Olle;
Wehlin, Camilla;
Persson, Johan A;
Ölvander, Johan

Linköping University

ABSTRACT

In order to efficiently design and deliver customized products, it is crucial that the process of translating customer needs to engineering characteristics and into unique products is smooth and without any misinterpretations. The paper proposes a method that combines design optimization with value-driven design to support and automate configuration of customized products. The proposed framework is applied to a case example with spiral staircases, a product that is uniquely configured for each customer from a set of both standard and customized components; a process that is complex, iterative and error-prone. In the case example, the optimization and value-driven design models are used to automate and speed-up the process of delivering quotations and design proposals that could be judged based on both engineering characteristics as well as their added value, thereby increasing the knowledge at the sales stage. Finally, a multi-objective optimization algorithm is employed to generate a set of Pareto-optimal solutions that contain four clusters of solutions that dominate the baseline design. Hence the decision-maker is given a set of optimal solutions to choose from when balancing different economical and technical characteristics.

Keywords: Optimisation, Product architecture, Digital / Digitised engineering value chains, Staircases

Contact:

Vidner, Olle
Linköping University
Division of Product Realisation
Sweden
olle.vidner@liu.se

Cite this article: Vidner, O., Wehlin, C., Persson, J. A. Ölvander, J. (2021) 'Configuring Customized Products with Design Optimization and Value-Driven Design', in *Proceedings of the International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.74

1 INTRODUCTION

In order for the process of configuring and delivering customer unique products to be more efficient and simultaneously keeping it flexible, information handling and verification is crucial. Misinterpretations of customer needs can be a source of errors in later stages, leading to delays, increased cost and wasted resources. The early sales (or quotation) stage is the first important source of information which sets the prerequisites for the process that is to follow. In product development, the design paradox (Ullman, 2010) is often used to address common challenges in product development processes. It illustrates the lack of knowledge that characterizes the early stages of development where the design freedom is at its highest, and the increased knowledge during later stages where costs of changes are increasing exponentially.

In this paper the use of design optimization and value-driven design (VDD) is applied to a case example with spiral staircases, a product that is designed uniquely for each customer, containing a set of both standard and customized components, whose configuration process is complex, iterative and error-prone. In the case example, the optimization and value-driven design models are used to increase the knowledge at the quotation stage, and to support decision-makers. Through optimization with computational models, design alternatives of the staircase are generated according to the customer context and are presented in terms of value for the different stakeholders. This enables the possibility to create fast quotations based on verified configurations, without the need of manual iterations between the sales and design department. This reduces the risk of errors due to misinterpretation and provides competitive advantages.

Spiral staircases can be seen as mass customized products, meaning products developed and designed in a process where each customer need is met from a set solution space through a stable, flexible and responsive process (Piller, 2004). In order for this process to be efficient and reliable, a design automation approach has been applied, where knowledge-based models contain the rule base for generation of feasible designed staircases through an optimization framework. This enables the generation of fast and reliable quotations in a sales context, which is important for avoiding errors and delays often caused by misinterpretations and assumptions based on experiences.

In this paper, we will show how VDD can be integrated into product configurators and how optimal trade-offs can be made through the use of simple value models and multi-objective optimization. Additionally, a number of tools to aid the data processing within such configurators are described. Hence the paper presents a novel approach combining VDD and multi-objective optimization together with tools that support the decision-making process. The methods presented is then applied on an industrial staircase example.

The remainder of the paper is outlined as follows. First a theoretical background is given in order to introduce the different domains that the developed tools and methods are built upon. Thereafter the case of industrial staircase configuration is described, followed by implementation and optimization results. Finally the discussion and conclusion chapter summarizes the paper.

2 THEORETICAL BACKGROUND

The methods and tools elaborated on within this paper relies on four main theoretical areas: configurators, i.e. the ability to develop customized solutions based on existing configurable modules; multi-disciplinary and multi-objective optimization algorithms as drivers in the search for optimal configurations; value-driven design methods to express the goal for the optimization; and finally post-optimization analysis in order to support the decision-making.

2.1 Configurators

Efficient implementations of mass customization in industry are often based on some sort of *product configurator*, as described by e.g. Hvam, Mortensen, and Riis (2008) and Tseng and Piller (2003). When considering a product configurator and its actors together, this unity can be referred to as a *configuration system* (Forza and Salvador, 2002).

In a case study conducted by Hvam, Haug, et al. (2013), it is concluded that the introduction of product configuration systems could reduce the lead-time with as much as 90 percent (partly because of more correct and timely specifications) and lead to increased sales, reduced costs, and formalized engineering knowledge among other effects.

In the context of a product that is the subject of a configuration system, [Forza and Salvador \(2002\)](#) distinguish a product's attributes into two main categories:

- *Commercial characteristics*: the product attributes (or requirements) that are to be defined by the customer.
- *Technical characteristics*: the product attributes that are to be manually or automatically defined by/through the configuration system, to achieve the given commercial characteristics.

This paper focuses on creating a configuration system for supporting the quotation process in the early sales stage, where geometric models are simple mathematical relations that represents the connection between the technical and commercial characteristics of the product. In order to conduct more detailed modeling, alternative approaches could be used, such as the principle of High-level CAD templates (HLCT:s) as described by [Amadori et al. \(2012\)](#) or the CAD templates approach integrating Knowledge Based Engineering environments as described by [Verhagen, Stjepandić, and Wognum \(2015\)](#). Furthermore, [Verhagen, Stjepandić, and Wognum \(2015\)](#) also discuss the use of product configurators as a means to achieve mass customization based on product modularity and supported by traditional design engineering methods. This paper extends the methods described above by the inclusion of formal optimization methods and the utilization of value-driven design to guide the product configuration process, however using more simple geometric representations.

2.2 Optimization

Many engineering problems can be considered as optimization problems, where the engineer has a set of parameters to tune in order to reach a certain goal. However, in practice many problems are much more complex and involves not just one engineer but multiple departments within a company, where each may have different conflicting objectives. Hence a problem may become multi-disciplinary ([Simpson and Martins, 2011](#)) and multi-objective ([Hwang, Paidy et al., 1980](#)). Furthermore, in the general case it might not be possible to obtain derivatives of the objective function, so methods that do not rely on analytical derivatives and are able to handle multi-disciplinary and multi-objective problems are of special interest for these types of problems. An example of such a method is the multi-objective genetic algorithm NSGA-II ([Deb et al., 2002](#)), that is widely used in this type of application.

2.3 Value-driven design

Value-driven design is a methodology for considering technical systems under design through the perspective of economic theory ([Collopy and Hollingsworth, 2011](#)). Being a methodological framework, it has several different interpretations and modes of implementation ([Lee and Paredis, 2014](#)). Within the scope of this paper, it will however be referred to as the use of mathematical models, for transforming technical traits of systems into basis for decision-making in monetary terms. Within the framework presented by [Lee and Paredis \(2014\)](#), this would represent *Artifact-Focused Decision Making*, where the focus lies in maximizing the value of a given artifact under consideration. Value-driven design in this regard has been utilized mainly within the aerospace industry (in examples such as in [Castagne, Curran, and Collopy \(2009\)](#)), where design decisions have potentially large and complex consequences throughout the product's complete life-cycle ([Price et al., 2012](#)) but there is nothing in the term's definition that limits its utilization within other fields.

In this paper, value-driven design will serve as the basis for formulating objective functions and for guiding the optimization algorithm in the search for product configurations with optimal value. The main concept in use here is that of *surplus value* (V_S) in its basic form, where it represents the arithmetic difference between the *reservation price* (P_R) and the total incurred *costs* of manufacturing the product (C_{man}), as formulated by [Price et al. \(2012\)](#):

$$V_S = P_R - C_{man} \quad (1)$$

2.4 Post-optimization analysis

When running optimization studies, the obtained results might need to be considered as a *set* of solutions, rather than one single optimal solution. This is especially true for multi-objective optimization formulations, where it is generally difficult to select one single best design unambiguously out of the Pareto-optimal set of solutions. Instead, a decision-maker may need to establish relations between design variables, objectives and constraints, referred to as *post-optimal* or *post-optimization* analysis by [Krus](#)

(2000). To establish these relations and to study the properties and trade-offs between different solutions before deciding upon a solution to investigate further, a number of activities can be carried out, for instance:

- filtering for feasible solutions in a solution set, i.e. discarding infeasible solutions;
- selecting subsets of a solution set, such as the Pareto-optimal solutions;
- grouping (or clustering) solutions with respect to certain variables;
- ranking solutions to obtain one "optimal" point, using e.g. TOPSIS (Hwang and Yoon, 1981), Grey Relational Analysis (Martínez-Morales, Pineda-Rico, and Stevens-Navarro, 2010) or ELECTRE (Wang and Triantaphyllou, 2008) methods.

These manipulations and analyses of different solutions posterior to the optimization run-time are all here referred to as *post-optimization analysis*.

3 APPLICATION OF THE PROPOSED FRAMEWORK

Spiral staircases are products that are customer-unique and adapted for each sales case. They are typically configured with a set of standard components; steps in a set of varying radii and in some cases standard landings and railings types, and a set of customized components; center pillar, railings, handrails and landings, all seen in figure 1.

The design of the spiral staircase is dependent upon several factors. For one, the environment and the surrounding, which limits the possible entry and exiting angles for the staircase, and is concluded with certain intervals of angles for each floor's entry and exit, and additionally for each case of the preferred turning of the staircase; clockwise and counter-clockwise turning. Secondly, a free head clearance (headroom) is required for a person to enter and walk the stair properly. These factors together limit the possible rises of the stair and thus the step depth and height in the walking line, whose relational values are considered more or less comfortable to walk in. Beyond this, the staircase must satisfy varying national legislative requirements, e.g. width and lengths between landings dependent on building capacity and intention of use and safety requirements of minimal railing height and maximum gaps for child safety, etc.



Figure 1. Photo of a manufactured spiral staircase, consisting of 3 segments. Reproduced from Weland AB (2016) with permission.

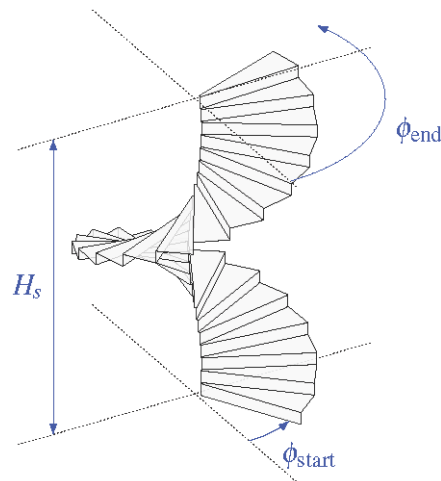


Figure 2. Schematic representation of a spiral staircase segment and some important design parameters and variables.

The design challenge lies in finding feasible combinations of values, representing the different design factors mentioned above. These can be considered as a number of design variables, represented by \bar{x} in table 1 and visualized through figure 2.

Many staircases (such as the one depicted in figure 1) consist of multiple (N) segments in sequence, where each segment (indexed by s) consist of n_s steps. The problem complexity is further enhanced for each introduced segment, since the segments cannot be considered in isolation. Modified characteristics for one segment must be accommodated for in another segment.

One way of taking on these challenges is by introducing a configurator, as has been previously demonstrated by Poot et al. (2020). Here, the configuration process is partly manual and partly automated – the user is responsible for inputting their requirements (i.e. the *commercial characteristics*) into the configurator, while the configurator is responsible for the automatic search and evaluation of feasible design solutions (i.e. the *technical characteristics*) that realizes the specific commercial characteristics. That is, instead of letting the user define the *solution* to their specific problem (as is fairly common for existing configuration systems), we propose an approach where the user defines the *problem* and the configuration system automatically finds the solutions using optimization.

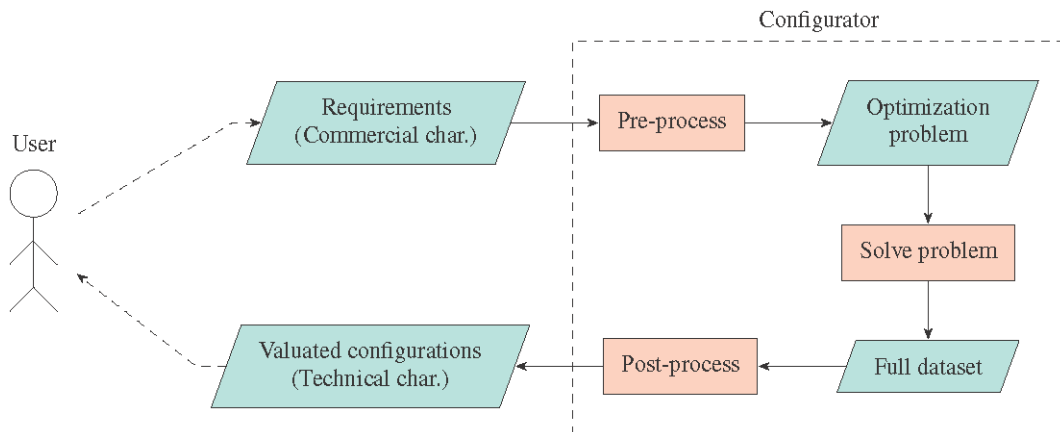


Figure 3. Conceptual overview of proposed product configuration system framework.

The spiral staircase design problem is however characterized such, that for each defined set of commercial characteristics, there is a large candidate set of feasible technical characteristics. Getting an overview of this set and making a decision is therefore hard. While it would be possible to let the configurator choose one candidate automatically, this would extend the responsibility of the system to correctly identify the *single best solution*. However, as pointed out by Collopy (2012), design is inherently uncertain and deterministic system models might not be fully able to manage these uncertainties and make good decisions autonomously. Instead, we want the human actors of the configuration system to be able to make the final decisions on technical characteristics, with constructive *support* from the configurator. Thus, the configurator is responsible for finding and presenting *good* solutions, while the human actors are responsible for selecting the *best* solution. More importantly, the basis on which the decision-maker acts is not only of a technical nature, but it is also coupled with the configuration's anticipated *value*. The architecture of this proposed product configuration system framework is summarized and represented in figure 3.

3.1 Computation and optimization model

To evaluate solutions to the design problem, a number of computational models are employed to give quantitative measures on dependent product properties, given the values of a set of independent design properties. All models are defined by relatively simple mathematical expressions or database lookups, implemented in the Python programming language.

The different computational models are integrated by using the OpenMDAO software library (Gray et al., 2019) to define components, variables and connections between these. Here, certain model inputs are defined as design variables while certain outputs are defined as responses, i.e. constraints and objectives (as seen in table 1). The computational models are connected according to figure 4, where \bar{x} is the vector of design variables and \bar{y}_{\dots} are vectors of the corresponding model's output variables. The meaning and content of the variables used are summarized in table 1 and selectively described below.

The *geometry* discipline model encapsulates the calculation of the complete staircase trajectory and step properties (e.g. depth, height), represented by $\bar{y}_{\text{geometry}}$. All these estimations are carried out through explicit mathematical rules, for instance relating the step depths and heights to the number of steps and the given start/end angles. A concrete example is the definition of the *segment sweep angle* ϕ_{sweep} for a

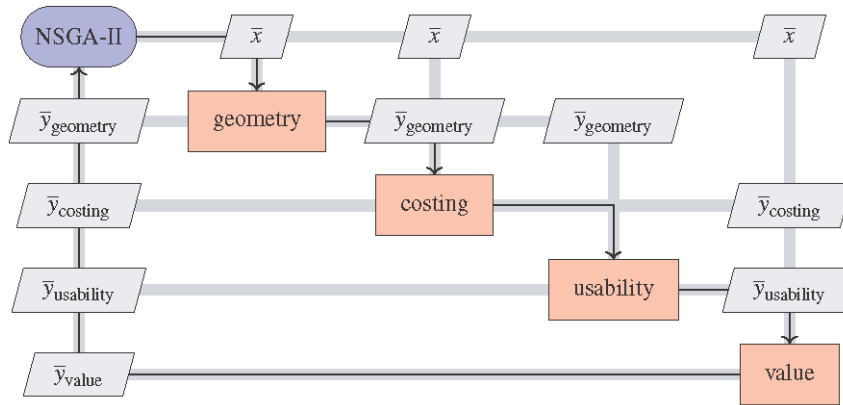


Figure 4. Schematic XDSM diagram (Lambe and Martins, 2012) for the calculation and optimization model, showing the NSGA-II driver controlling the four computational models along the main diagonal, executed in series (as indicated by the process arrows). Symbols in the upper and lower diagonals signify dependencies of the computational models and the optimization driver, respectively.

Table 1. Overview of input and output variables, along with their roles in the optimization formulation.

Name	Description	Role
\bar{x}	Design variables: <ul style="list-style-type: none"> Start and end angle ($\phi_{start,s}$ and $\phi_{end,s}$) of each segment Extra revolution angles ($\phi_{extra,s}$) for each segment Number of steps per segment (n_s) Outer radius (r) Turning direction 	Design variable Design variable Design variable Design variable Design variable
$\bar{y}_{geometry}$	Geometry representation: <ul style="list-style-type: none"> Sweep angle ($\phi_{sweep,s}$) of each segment Step depth (d_s) in each segment Step height (h_s) in each segment 	Constraint Constraint
$\bar{y}_{costing}$	Costing estimations: <ul style="list-style-type: none"> Material cost (C_{mat}) Manufacturing cost (C_{man}) 	Objective
$\bar{y}_{usability}$	Usability measurements: <ul style="list-style-type: none"> Minimum headroom (HR_{min}) Violation of usability ideals (IV_s) 	Constraint
\bar{y}_{value}	Product value estimations: <ul style="list-style-type: none"> Reservation price (P_R) Surplus value (V_S) 	Objective

given segment s , where $\phi_{extra,s}$ is divisible by 360 (i.e. complete revolutions):

$$\phi_{sweep,s} = \phi_{end,s} - \phi_{start,s} + \phi_{extra,s} \quad \phi_{extra,s} | 360 \quad (2)$$

To make cost estimations, the *costing* model looks up material costs (C_{mat}) for different components in a database. For instance, knowing the staircase radius and the number of steps (from \bar{x}) makes it possible to estimate an important part of the material costs. Based on the material costs, the total manufacturing costs are estimated simply through scaling by a factor $k_{C_{man}}$:

$$C_{man} = C_{mat} * k_{C_{man}} \quad (3)$$

The *usability* model evaluates design solutions from a usability (or ergonomics) perspective, giving metrics such as the minimum occurring headroom HR_{min} and the relations between step depth and height in each segment of the staircase. Furthermore, it calculates a measurement IV_s of how much a certain segment s violates certain *ideal* usability measurements.

With all calculations above carried out, the product value can be estimated by the *value* model, based upon the surplus value definition, see equation (1). The reservation price P_R of a given design solution is given by the sum of all N segments' corresponding price contributions. These contributions are in turn based on the assumption that each segment s has a *nominal* reservation price $P_{\text{nom},s}$ that can be *degraded* by a certain factor $k_{\text{degrad},s}$:

$$P_R = \sum_{s=1}^N P_{\text{nom},s} * (1 - k_{\text{degrad},s}) \quad (4)$$

One segment's nominal reservation price consists of two terms: (1) a fixed unit price term P_s [SEK] and (2) a variable term $p_{H,r} * H_s * r$, based on height and radius where $p_{H,r}$ [SEK/m²] is a constant, H_s [m] the height of the corresponding segment and r [m] the radius of the staircase. That is, a segment's nominal reservation price increases monotonously with an increased height and radius:

$$P_{\text{nom},s} = P_s + p_{H,r} * H_s * r \quad (5)$$

The price degradation factor $k_{\text{degrad},s}$ is based on the assumption that bad usability will impair the staircase's utility and therefore also its reservation price. Thus, the ideal violation IV_s (from $\bar{y}_{\text{usability}}$) is used to calculate this degradation:

$$k_{\text{degrad},s} = \min(IV_s, k_{\text{degrad}}^a) * k_{\text{degrad}}^b \quad (6)$$

Assuming that $IV_s \in [0, \infty)$, the constants k_{degrad}^a and k_{degrad}^b can be selected so that a certain violation yields the anticipated price degradation. In the examples evaluated within the context of this paper, $k_{\text{degrad}}^a = 1$ and $k_{\text{degrad}}^b = 0.9$, meaning that the reservation price degrades linearly such that

$$IV_s = 0 \Rightarrow k_{\text{degrad},s} = 0.0 \quad IV_s \geq 1 \Rightarrow k_{\text{degrad},s} = 0.9 \quad (7)$$

That is, segments with "bad" usability ($IV_s \geq 1$) lose 90 percent of their nominal reservation price.

3.2 Optimization formulation

To find solutions that minimizes/maximizes the objectives and satisfies the constraints defined, an optimization driver is set to control the model. In this case, an NSGA-II (Deb et al., 2002) implementation for OpenMDAO (Vidner, 2020a) has been developed and utilized to solve the problem formulation. Since the overall goal of the optimization strategy is geared towards *design exploration* (i.e. finding multiple "good" designs rather than one "best" design), the optimization objective is not to only maximize V_S (as would perhaps be the obvious choice). Rather, a bi-objective formulation is used, defined to maximize P_R and to minimize C_{man} , as $V_S = P_R - C_{\text{man}}$. This is for increasing the selection pressure towards design exploration in the genetic algorithm and has been observed to yield solutions that dominate those of a single-objective formulation for this specific problem, in the same number of function evaluations.

Thus, the optimization formulation used is summarized by equation (8).

$$\begin{aligned} & \text{maximize} && P_R(\bar{x}) && \text{and} \\ & \text{minimize} && C_{\text{man}}(\bar{x}) \\ & \text{with respect to} && \bar{x} \\ & \text{subject to} && h_{\text{lower}} \leq h_s(\bar{x}) \leq h_{\text{upper}} && \text{for all } s \in [1, N] \\ & && d_{\text{lower}} \leq d_s(\bar{x}) \leq d_{\text{upper}} && \text{for all } s \in [1, N] \\ & && HR_{\text{min}}(\bar{x}) \geq HR_{\text{min,lower}} \end{aligned} \quad (8)$$

where $()_{\text{lower}}$ and $()_{\text{upper}}$ are given bounds, adjustable for different cases.

3.3 Execution and results

The optimization application has been tested with input data based on an actual customer case for a three-floor staircase. A baseline design in the form of a manual solution for the same case has been used as a basis for comparison and as reference for relative measurements presented below. The optimization driver has been set to evaluate 52 individuals through 50 generations. During optimization runtime, all

variable values have been automatically stored in an *xarray* dataset (Hoyer and Hamman, 2017) through the use of the *Scop* extension for OpenMDAO (Vidner, 2020b).

The left part of figure 5 shows this full dataset in objective space, indicating that the optimization algorithm has put a majority of its effort towards evaluating designs leading up to, and along, the emerging Pareto-frontier in the lower-right part of the diagram.

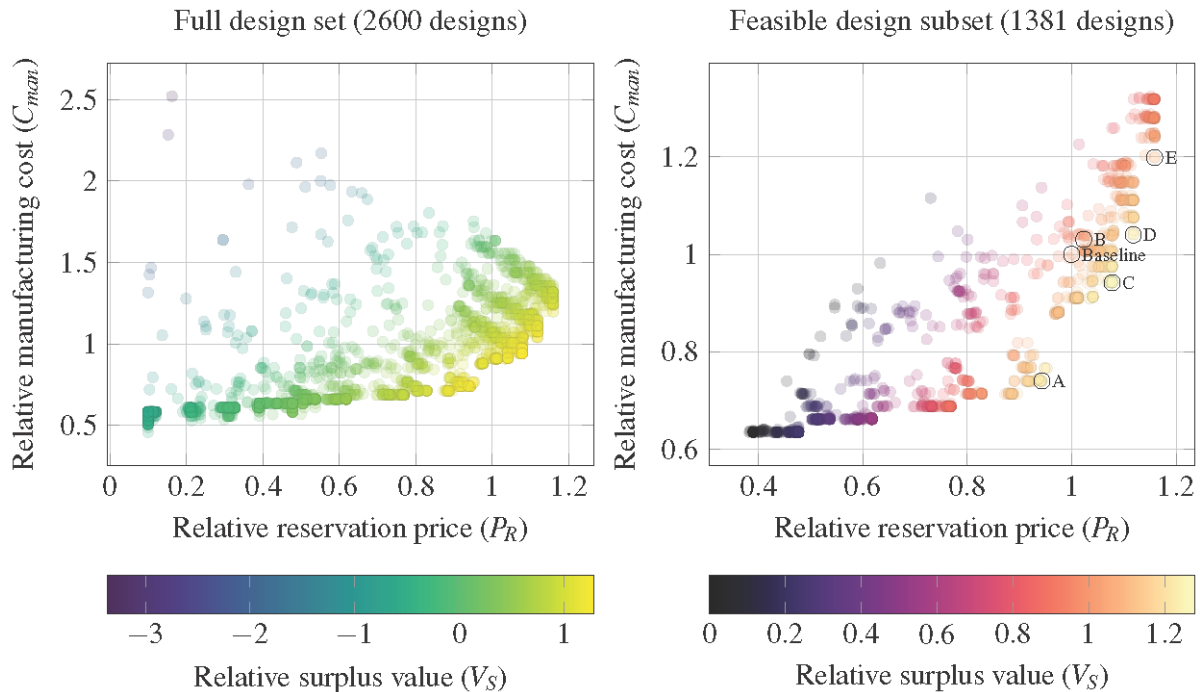


Figure 5. Scatter plots of design sets from optimization execution, in objective space. Selected designs are marked with a circle and Design ID (see table 2). One point is one evaluated design and all points have the same opacity. All three dimensions are relative to the baseline design. The utopian point lies towards the lower-right corner.

From the full dataset, the feasible subset has been extracted, where *feasible* refers to designs that (1) satisfy all optimization constraints and (2) have a positive surplus value. These 1381 designs are represented in the right part of figure 5.

In order to reduce the amount of data used for decision-making, while still maintaining representative information on the solution diversity, the feasible solutions have been grouped with respect to the radius (r) value. Within each one of these groups, the design with the highest surplus value (V_S) have been selected. These designs along with the baseline design are represented in table 2 and marked in figure 5 to visualize their relative positions in objective space. While design B is similar to the baseline design, they are both dominated by designs A, C, D and E.

Table 2. Selected designs for decision-maker to consider, along with baseline design.

Design ID	Radius [mm]	Sweep angle per segment [°]	Step count per segment	Rel. P_R [-]	Rel. C_{man} [-]	Rel. V_S [-]
A	900	300, 285	14, 15	0.94	0.74	1.25
Baseline	1000	260, 280	14, 15	1.00	1.00	1.00
B	1000	281, 284	14, 16	1.02	1.03	1.01
C	1100	241, 277	14, 16	1.08	0.94	1.28
D	1200	235, 245	14, 16	1.12	1.04	1.24
E	1300	243, 249	15, 16	1.16	1.20	1.10

4 DISCUSSION AND CONCLUSIONS

A value-driven design based formulation is here used in a two-fold sense: to guide the optimization (*computerized* decision-making) and to give value-based insights on design decisions to humans (*computer-supported* decision making). It is shown in section 3.3 that this formulation and the method described successfully finds and selects designs that are similar to the baseline, and additionally, it succeeds in finding solutions that dominate the baseline, indicating that the method have the potential to find even more valuable solutions in less time than what it would take for an engineer to do the same.

Through implementation of this method in a sales context, the uncertainty associated with the initial development stages can be reduced already at the quotation stage, and also lead to the build-up of product knowledge (as the design space is searched and confined). This could imply a number of effects for the company adopting the approach. First off is the competitive advantage of being able to quickly generate reliable sales support and quotations to provide to the customer. Secondly, this potentially implies elimination of errors, otherwise prone to occur later in the process, due to assumptions made in the quotation stage, leading to increased costs. Finally, the amount of resources needed in the process could potentially be significantly reduced through the automation of repetitive tasks.

This first implementation of the value formulation as objective in the optimization is a simplified representation and is by that an obvious candidate for improvement of the overall framework. The optimization could also be performed on a higher level than only for one isolated sales process, meaning that the objective would be to maximize the overall surplus value for several or all current sales processes in the company. In either case, combining the optimization and configurator architecture with a visualization solution could further improve the usability of the proposed framework, by providing the user with accurate representations of the calculated solutions.

While even the simplistic value model presented here may give some indications on the utility of value models in optimization-based product configurators, further investments must be made into the validation of the specific model. It does not take the product's lifecycle into account and might therefore be fairly limited in its applicability. On the other hand, the lifecycle of a spiral staircase is not very complex once it has been delivered. The product may be highly usable and sometimes even mandatory due to safety regulations, but may not always bring that much tangible value in return for the investments that it incurs. Yet still, it is clear that some designs can be more valuable than others and that *no* design is much less valuable than *a* design – having a design in place might enable the use of a facility. However, adapting and applying the presented approach onto categories of products with more complex relations between their lifecycle and their value, is seen as an even more exciting area of further research.

In the case example demonstrated above, optimization and value-driven design are used to systematically explore the design space and generate design alternatives for spiral staircases. The design alternatives are presented in terms of value for the main stakeholders (i.e. supplier and customer), as an interpretation of the technical specifications of the staircases. This abstraction can be a support for both technically-oriented and business-oriented users, as it might help in viewing the product in its bigger, economic perspective. By developing the approach further and taking it into new fields of the industry, it might serve as valuable support for human decision-makers in the very early stages of product development.

REFERENCES

- Amadori, K. et al. (2012). "Flexible and Robust CAD Models for Design Automation". *Advanced Engineering Informatics* 26.2, pp. 180–195. DOI: 10/f228fq.
- Castagne, S., R. Curran, and P. D. Collopy (2009). "Implementation of Value-Driven Optimisation for the Design of Aircraft Fuselage Panels". *International Journal of Production Economics* 117.2, pp. 381–388. DOI: 10/dsqw6n.
- Collopy, P. D. (2012). "A Research Agenda for the Coming Renaissance in Systems Engineering". *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. Nashville, Tennessee: American Institute of Aeronautics and Astronautics. DOI: 10/gjsffd.
- Collopy, P. D. and P. M. Hollingsworth (2011). "Value-Driven Design". *Journal of Aircraft* 48.3, pp. 749–759. DOI: 10/dmwprn.
- Deb, K. et al. (2002). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. DOI: 10/bnw2vv.

- Forza, C. and F. Salvador (2002). “Managing for Variety in the Order Acquisition and Fulfilment Process: The Contribution of Product Configuration Systems”. *International Journal of Production Economics* 76.1, pp. 87–98. DOI: 10/b42trp.
- Gray, J. S. et al. (2019). “OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization”. *Structural and Multidisciplinary Optimization* 59.4, pp. 1075–1104. DOI: 10/gfwctcb.
- Hoyer, S. and J. J. Hamman (2017). “Xarray: N-D Labeled Arrays and Datasets in Python”. *Journal of Open Research Software* 5, p. 10. DOI: 10/gdqdmw.
- Hvam, L., A. Haug, et al. (2013). “Observed Benefits from Product Configuration Systems”. *International Journal of Industrial Engineering: Theory, Applications and Practice* 20.5-6, pp. 1–6.
- Hvam, L., N. H. Mortensen, and J. Riis (2008). *Product Customization*. Berlin: Springer.
- Hwang, C.-L., S. Paidy, et al. (1980). “Mathematical Programming with Multiple Objectives: A Tutorial”. *Computers & Operations Research* 7.1-2, pp. 5–31. DOI: 10/bw8cvv.
- Hwang, C.-L. and K. Yoon (1981). *Multiple Attribute Decision Making*. Ed. by M. Beckmann and H. P. Kunzi. Vol. 186. Lecture Notes in Economics and Mathematical Systems. Berlin, Heidelberg: Springer Berlin Heidelberg. DOI: 10/f8rt.
- Krus, P. (2000). “Post Optimal System Analysis Using Aggregated Design Impact Matrix”. *ASME International Design Engineering Technical Conference Computers and Information in Engineering Conference*. American Society of Mechanical Engineers.
- Lambe, A. B. and J. R. R. A. Martins (2012). “Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes”. *Structural and Multidisciplinary Optimization* 46.2, pp. 273–284. DOI: 10/fzg48s.
- Lee, B. D. and C. J. Paredis (2014). “A Conceptual Framework for Value-Driven Design and Systems Engineering”. *Procedia CIRP* 21, pp. 10–17. DOI: 10/gjgn8n.
- Martínez-Morales, J. D., U. Pineda-Rico, and E. Stevens-Navarro (2010). “Performance Comparison between MADM Algorithms for Vertical Handoff in 4G Networks”. *2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*, pp. 309–314. DOI: 10/b8gx99.
- Piller, F. T. (2004). “Mass Customization: Reflections on the State of the Concept”. *International Journal of Flexible Manufacturing Systems* 16.4, pp. 313–334. DOI: 10/cphjxw.
- Poot, L. P. et al. (2020). “Integrating Sales and Design: Applying CAD Configurators in the Product Development Process”. *Proceedings of the Design Society: DESIGN Conference* 1, pp. 345–354. DOI: 10/gjsffr.
- Price, M. et al. (2012). “A Novel Method to Enable Trade-Offs across the Whole Product Life of an Aircraft Using Value Driven Design”. *Journal of Aerospace Operations* 1.4, pp. 359–375. DOI: 10/gjsffq.
- Simpson, T. W. and J. R. R. A. Martins (2011). “Multidisciplinary Design Optimization for Complex Engineered Systems: Report From a National Science Foundation Workshop”. *Journal of Mechanical Design* 133.10, pp. 101002-1–101002-10. DOI: 10/b2s9br.
- Tseng, M. M. and F. T. Piller, eds. (2003). *The Customer Centric Enterprise: Advances in Mass Customization and Personalization*. Berlin ; New York: Springer. 535 pp.
- Ullman, D. G. (2010). *The Mechanical Design Process*. 4th ed. McGraw-Hill Series in Mechanical Engineering. Boston: McGraw-Hill Higher Education. 433 pp.
- Verhagen, W. J. C., J. Stjepandić, and N. Wognum, eds. (2015). *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*. 1st ed. Cham: Springer. 839 pp. DOI: 10/f8rv.
- Vidner, O. (2020a). *OpenMDAO-NSGA: NSGA-II and NSGA-III Implementations for OpenMDAO*. Version v0.1. Zenodo. DOI: 10.5281/ZENODO.4279483.
- (2020b). *Scop: An Opinionated, Self-Contained Data Format and Post-Optimization Toolchain*. Version v0.3. Zenodo. DOI: 10.5281/ZENODO.4263728.
- Wang, X. and E. Triantaphyllou (2008). “Ranking Irregularities When Evaluating Alternatives by Using Some ELECTRE Methods”. *Omega*. Special Issue Section: Papers Presented at the INFORMS Conference, Atlanta, 2003 36.1, pp. 45–63. DOI: 10/fghj9s.
- Weland AB (2016). *Product Manual Edition 6*.