22nd INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2020
CAMBRIDGE, MASSACHUSETTS, USA, 13 – 15 October, 2020

Optimizing Distributed Design Processes for Flexibility and Cost

Marco Daub[1], Ferdinand Wöhr[1,2], Markus Zimmermann[1]

[1]Technische Universität München
[2]BMW Group

Abstract: This paper considers three different modes of design work that is distributed over several design parties: independent design (in parallel, no design interactions, sub-system requirements), dependent design (sequential, one-way design interactions, updated sub-system requirements) and interdependent design (in parallel or sequential, two-way design interactions, only system requirements). Each mode requires particular coordination strategies to be successful. One coordination strategy is based on requirement formulation: Both system and sub-system requirements are expressed as so-called solution spaces. Solution spaces represent sets of permissible designs where sub-system (or component) solution spaces can be deduced from the system solution space. The larger the size of a sub-system solution space, the more options for sub-system design decisions satisfying the overall system requirements exist and thus the larger the design flexibility. The three modes are applied to two industrial design problems and evaluated with respect to total flexibility and cost related to iteration steps, interactions between the design parties and requirement formulation. The resulting framework is applicable to general systems design problems.

Keywords: Systems Design, Process Architecture, Solution Spaces, Concurrent Design

1 Introduction

A goal in systems engineering is to break down complexity of the design process in order to simplify, to accelerate, and to reduce costs. The literature on decomposition strategies is extensive. However, it is often descriptive and cumbersome to apply to mathematical design models. In this field, “decomposition-based design optimization”, within “multidisciplinary design optimization”, in which a systems design problem is usually decomposed into separate design problems, is predominant, see (Papalambros and Wilde, 2017). Unfortunately, this is often associated with tight component targets with little tolerance resulting in little robustness and design flexibility. For the purpose of this paper, flexibility refers to the number (or the size of the set) of design options that satisfy the overall system requirements.

This can be alleviated using a design approach based on solution spaces. The key idea of this approach is to consider a set of permissible system designs from which quantitative, least restrictive subsystem requirements can be derived instead of searching for a single, optimal system design in the first place. Here, flexibility can be provided for design work done in parallel, see (Daub et al., 2020; Zimmermann and Hoessle, 2013), or done

sequentially, see (Funk et al., 2019; Vogt et al., 2018). By using these methods, also no iterations on the subsystem design decisions are required if the underlying design model is accurate enough. For some problems however, flexibility might still be small and coupled design decisions, which allow iterations combined with interactions between the designers, would be preferable in order to increase design flexibility. This raises the question of how to choose and sequence the different types of design decisions for an optimal process architecture.

In this paper, this question is addressed. Therefore, a review on systems design based on solution spaces for the different types of design decisions is presented in Section 2. Then, fundamentals for the assessments of a solution-space based design process are discussed and a framework to yield an optimal process architecture is proposed in Section 3. It is applied to problems from the automotive industry, vehicle crash design and battery pack design, before this paper is concluded in Section 4.

2 Systems Design using Solution Spaces

2.1 Definitions

The focus is put on systems design with mathematical design models and continuous design variables. Here, there are d independent *design variables* $x_i \in \mathbb{R}$, which are collected in $\boldsymbol{x} = (x_1, \dots, x_d)$. The vector $\boldsymbol{x} \in \mathbb{R}^d$ contains all relevant design variables and is named a *system design*. In general, the design variable values can be selected, or at least be controlled, by one or multiple designers. The minimum values which can be chosen for x_i are denoted by $x_{\mathrm{ds},i}^{\mathrm{l}}$ and the maximum values by $x_{\mathrm{ds},i}^{\mathrm{u}}$, i.e., $x_i \in \left[x_{\mathrm{ds},i}^{\mathrm{l}}, x_{\mathrm{ds},i}^{\mathrm{u}}\right]$, $i = 1, \dots, d$. These bounds form the *system design space* $\Omega_{\mathrm{ds}} \subset \mathbb{R}^d$ with

$$\Omega_{\mathrm{ds}} = \left[x_{\mathrm{ds},1}^{\mathrm{l}}, x_{\mathrm{ds},1}^{\mathrm{u}}\right] \times \cdots \times \left[x_{\mathrm{ds},d}^{\mathrm{l}}, x_{\mathrm{ds},d}^{\mathrm{u}}\right]. \tag{1}$$

For a given design model, each system design $\boldsymbol{x} \in \Omega_{\mathrm{ds}}$ is associated with specific system responses. Hence, s*ystem responses* $z_j \in \mathbb{R}$ can be represented as images of a system design $\boldsymbol{x}$ by *system performance functions* f_j, $j = 1, \dots, m$, where m denotes the number of relevant system responses. It holds

$$f_j \colon \mathbb{R}^d \to \mathbb{R}, \boldsymbol{x} \mapsto z_j = f_j(\boldsymbol{x}). \tag{2}$$

There are requirements on the system responses which must be fulfilled by a system design in order to be *permissible*. These are formulated as upper threshold values $f_{\mathrm{c},j}$. Note that every lower threshold can be transformed into an upper threshold by multiplying both the corresponding system performance function and the lower threshold with -1. The set of all permissible system designs $\boldsymbol{x}$ is called *complete system solution space* $\Omega_{\mathrm{c}} \subset \mathbb{R}^d$ with

$$\Omega_{\mathrm{c}} = \{\boldsymbol{x} \in \Omega_{\mathrm{ds}} \mid f_j(\boldsymbol{x}) \leq f_{\mathrm{c},j}, j = 1, \dots, m\}. \tag{3}$$

2.2 Independent, Dependent, and Interdependent Design Decisions

In this paper, a distributed design process in which each one designer is responsible for specifying the value for one design variable is considered. This can be extended to

situations in which the design variables are grouped as components and component designers are responsible for selecting their associated design variable values, see (Daub et al., 2020; Daub, 2020). Between two designers, there are three different types for the flow of information about their selected design variable values: No flow of information for *independent* design decisions (Mode 1), one-way flow of information for *dependent* design decisions (Mode 2), and two-way flow of information for *interdependent* design decisions (Mode 3), see (Pimmler and Eppinger, 1994). They are visualized in Fig. 1.

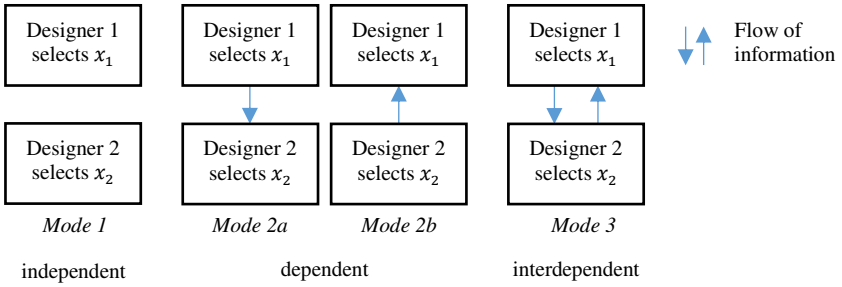


Figure 1. Independent, dependent and coupled design decisions between two designers.

In the following, it is presented how the three modes can be used in the framework of flexible designing based on solution spaces where the system requirements must be fulfilled. For reasons of simplicity, only purely independent, dependent, or interdependent design decisions are considered in this subsection. In the next section, an optimal sequencing of the design work is discussed.

Independent design decisions can be done in parallel, i.e., simultaneously. This can be understood as a concurrent engineering approach. For a flexible and independent design process, independent intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, in which designer i must select the values of x_i, are required. Furthermore, the Cartesian product of these intervals must be a subset of the complete solution space in order to guarantee a permissible system design, meaning

$$\left[x_1^{\mathrm{l}}, x_1^{\mathrm{u}}\right] \times \cdots \times \left[x_d^{\mathrm{l}}, x_d^{\mathrm{u}}\right] \subseteq \Omega_{\mathrm{c}}. \qquad (4)$$

In general, there are various intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, which fulfill Eq. (4). Among them, the ones that provide the most design flexibility are usually preferred. This flexibility is quantified, e.g., as the volume of the Cartesian product in (Zimmermann and Hoessle, 2013), or as the minimum interval length in (Fender et al., 2016). An approach that also considers different uncertainty magnitudes of the design variables is presented in (Daub and Duddeck, 2019).

Dependent design decisions are taken sequentially, what can be understood as a traditional or over-the-wall engineering approach. Assuming a consecutive ordering of the design decisions according to their indices, the required interval $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, in which designer i must select the values of x_i, depend on the first $i-1$ design decisions. Moreover, the interval $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ must be a subset of the projection along the ith coordinate axis of the updated

complete system solution space $\Omega_c(x_1, \dots, x_{i-1})$, in order to guarantee a permissible system design regardless of the decision for $x_i \in [x_i^l, x_i^u]$, i.e.,

$$[x_i^l, x_i^u] \subseteq \text{proj}_i\,(\Omega_c(x_1, \dots, x_{i-1})). \qquad (5)$$

Again, there are various intervals which fulfill Eq. (5), and among them, maximum flexibility is preferred. If flexibility is quantified as the volume, which corresponds to the interval length here, $[x_i^l, x_i^u]$ is defined by using an equal sign in Eq. (5) if Ω_c is connected. However, other approaches are also conceivable, e.g., when different uncertainty magnitudes of the design variables are considered, see (Daub, 2020).

In contrast to independent and dependent design decisions, interdependent design decisions allow each designer to adapt their selection for the design variable values in dependence of the selection of the other designers. In theory, this may result in arbitrarily many iterations for the system design. This can be seen as a series of single or parallel design decisions in which every design variable occurs more than once, see (Devendorf and Lewis, 2011). The iterations can terminate if the updated system design is permissible.

3 Optimal Architecture for a Solution-Space-Based Design Process

3.1 Problem Statement

There are various decomposition strategies which sequence independent, dependent, and interdependent design decisions for a solution-space-based design process. In the following, such a *decomposition strategy* is denoted by $D \in \{D_1, \dots, D_{n_{ps}}\}$, where n_{ps} denotes the number of possible strategies. If a system design consists of two design variables for example, there are four possible strategies which are closely related to the design modes shown in Fig. 1: Designer 1 and designer 2 design independently (D_1, mode 1), dependently where either designer 1 (D_2, mode 2a) or 2 (D_3, mode 2b) starts, or interdependently (D_4, mode 3). The number of strategies n_s increases rapidly with the number of design variables. In Fig. 2, an example for D to obtain a system design with eight design variables is visualized.

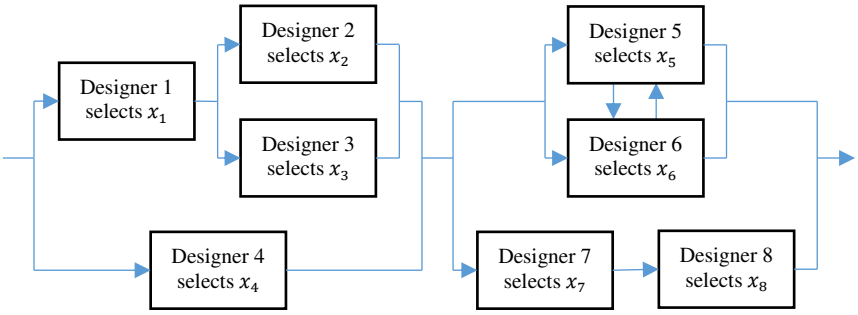


Figure 2. Example to for a decomposition strategy to sequence eight design decisions.

In addition to a decomposition strategy, a *flexibility strategy* F for optimizing the intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \dots, d$, of a solution-space based design process is required. This can for example be done by combining the flexibility measures from above. Overall, a design process that depends on both the decomposition and the flexibility strategy is obtained. As there are usually reasons or preferences for a specific flexibility strategy F, like using the volume as a natural flexibility measure, the focus is placed on finding an optimal decomposition strategy D for given flexibility strategy F. Here, the subsequent framework is proposed:

1. **Determine the feasible decomposition strategies**: Due to the rapidly growing number of decomposition strategies, with the number of design variables, not all of the strategies for a solution-space-based design process should be assessed. This is often not even necessary as the organizational structure with different designers may not allow to access the full spectrum of strategies. For example, it can be preferable that designer 1 starts designing if he is required in another project soon, or it might be difficult for designers to design interdependently due to a complicated communication infrastructure. The remaining, feasible decomposition strategies are collected in $\{D_1, \dots, D_{n_{\mathrm{fs}}}\}$, where n_{fs} denotes their number with $n_{\mathrm{fs}} \leq n_{\mathrm{ps}}$.

2. **Define assessment criteria for the decomposition strategies**: In order to assess the feasible decomposition strategies for a design process with a given flexibility strategy, one or multiple assessment criteria must be defined. In this paper, the *(total) design flexibility* $\mu(D, F)$, for which a large value is desired, and the *process cost* $C(D)$, for which a small value is desired, are taken into account. As these two measures should be defined with respect to the specific application in general, no specific definitions are provided here. As an example, the total design flexibility can be measured for example by the average, minimum, or maximum flexibility provided by F. The quantification of the process cost can be understood more general, for which, e.g., the considerations from the first step can be extended. Furthermore, it is also conceivable to quantify the involved design modes in terms of design flexibility and process cost directly, see the example in the next subsection. Usually, independent decisions are characterized by small flexibility and small cost, dependent design decisions by medium flexibility and medium cost, and interdependent decisions by large flexibility and large cost.

3. **Choose an optimal, feasible decomposition strategy:** For $n_{\mathrm{fs}} \geq 1$, it is necessary to assess the feasible decomposition strategies in terms of total design flexibility and process cost to choose an optimal decomposition strategy. As two assessment criteria are involved here, the goal becomes to find a Pareto optimal, feasible decomposition strategy that maximizes $\mu(D, F)$ and minimizes $C(D)$.

In the following, the framework is applied to two simple vehicle design problems, one from the field of crash design and the other one from the field of battery pack design.

3.2 Application to Industrial Examples

(a) Crash design problem, from (Zimmermann and Hoessle, 2013)

The problem of designing a vehicle front structure to account for a front crash load case against a rigid wall at full overlap is considered, see Fig. 3(a). Here, the structure is modeled as two sections with deformation lengths s_1 and s_2. For each section, it is assumed that there is a constant force which is necessary to deform this section, and which can be specified by the designer. This means that there are two designers responsible for two design variables, F_1 and F_2. In order for a system design to be permissible, it must fulfill requirements regarding the energy absorption, maximum acceleration, and the order of deformation, i.e.,

$$-s_1F_1 - s_2F_2 \leq -\frac{1}{2}mv_0; F_1, F_2 \leq ma_\mathrm{c}; F_1 - F_2 \leq 0, \qquad (6\text{-}8)$$

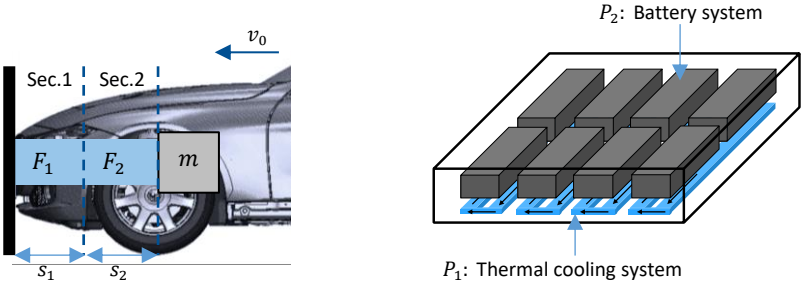
where m is the vehicle mass, v_0 the initial velocity, and a_c is the critical acceleration. The design space is set as $\Omega_\mathrm{ds} = [0\mathrm{N}, 500\mathrm{kN}] \times [0\mathrm{kN}, 500\mathrm{kN}]$, and the parameters as $s_1 = s_2 = 0.3\mathrm{m}$, $m = 1500\mathrm{kg}$, $v_0 = 15.6\frac{\mathrm{m}}{\mathrm{s}}$, $a_\mathrm{c} = 300\frac{\mathrm{m}}{\mathrm{s}^2}$, cf. (Daub et al., 2020).

(b) Battery pack design problem, simplified from (Wöhr et al., 2020)

The problem of designing a vehicle battery pack, consisting of a thermal cooling system and a battery system, is considered, see Fig. 3(b). Here, it is assumed that each the mechanical power of the thermal cooling system P_1 and the electrical power of the battery system P_2 can be specified by the designer. This means that there are two designers responsible for two design variables, again. In order for a system design to be permissible, it must fulfill two requirements regarding the power difference, i.e.,

$$-P_1 + P_2 \leq P_\mathrm{c}^\mathrm{u}; P_1 - P_2 \leq -P_\mathrm{c}^\mathrm{l}, \qquad (9)$$

where P^u is the critical upper power limit and P^l is the critical lower power limit. The design space is set as $\Omega_\mathrm{ds} = [0\mathrm{N}, 200\mathrm{kW}] \times [0\mathrm{kN}, 500\mathrm{kW}]$, and the power limits as $P_\mathrm{c}^\mathrm{l} = 150\mathrm{kW}$ and $P_\mathrm{c}^\mathrm{l} = 200\mathrm{kW}$, cf. (Wöhr et al., 2020).



(a) (b)

Figure 3. Vehicle crash design problem (a) and battery pack design problem (b).

As both the vehicle crash design problem and battery pack design problem have two design variables, there are four possible decomposition strategies $D_1, \dots, D_4$ for each problem, see above. As above, D_1 represents independent design decisions (mode 1), D_2 and D_3 represent dependent design decisions (mode 2a and 2b), and D_4 represents interdependent design decisions (mode 3). All are considered as feasible here. Moreover, the same flexibility strategy F is considered for the two problems: the volume of $\left[x_1^{\mathrm{l}}, x_1^{\mathrm{u}}\right] \times \left[x_2^{\mathrm{l}}, x_2^{\mathrm{u}}\right]$ is maximized for independent design decisions, and $\left[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}\right] = \mathrm{proj}_i\,(\Omega_{\mathrm{c}})$ is set for the first decision for dependent design decisions. In order to assess the total design flexibility $\mu(D, F)$ for the different decomposition strategies, the volume of the set of system designs that can be realized is used. It is defined as

$$\mu(D,F) = \begin{cases} \mathrm{vol}\left(\left[x_1^{\mathrm{l}}, x_1^{\mathrm{u}}\right] \times \left[x_2^{\mathrm{l}}, x_2^{\mathrm{u}}\right]\right) & \text{for } D_1, \\ \mathrm{vol}(\Omega_{\mathrm{c}}) & \text{for } D_2, D_3, D_4 \end{cases} \qquad (10)$$

In Fig. 4, maximum-volume inner boxes $\left[x_1^{\mathrm{l}}, x_1^{\mathrm{u}}\right] \times \left[x_2^{\mathrm{l}}, x_2^{\mathrm{u}}\right]$ are visualized along with the complete system solution space Ω_{c}. In addition, minimum outer boxes that represent $\left[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}\right] = \mathrm{proj}_i\,(\Omega_{\mathrm{c}})$ via $\mathrm{proj}_1\,(\Omega_{\mathrm{c}}) \times \mathrm{proj}_2\,(\Omega_{\mathrm{c}})$ are shown.

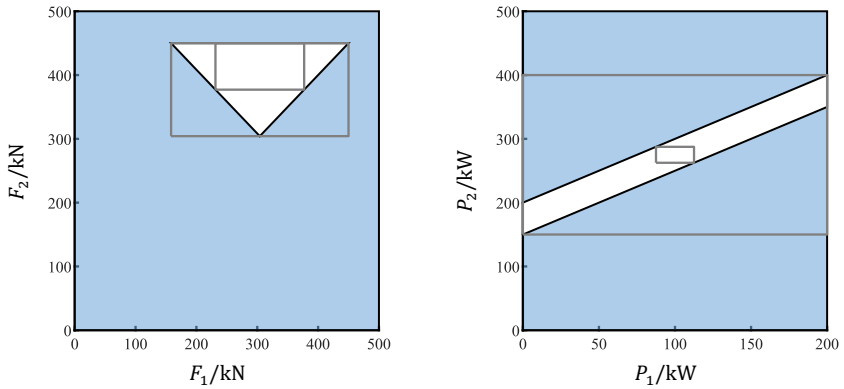


Figure 4. Complete system solution space (white) and system design space (blue) including maximum-volume inner boxes and minimum outer boxes (gray bounds) for problem (a) and (b).

In order to assess the process costs $C(D)$ of the different decomposition strategies, the cost for the number of requirement updates n_{r}, *the* number of design selections n_{s}, and the number of interactions between the designers n_{i} are considered, i.e.,

$$C(D) = n_{\mathrm{r}}(D)\mathrm{C}_{\mathrm{r}} + n_{\mathrm{s}}(D)C_s + n_{\mathrm{i}}(D)C_{\mathrm{i}}. \qquad (11)$$

For D_1, the requirements are updated for each design variable in the beginning, each design variable value is selected once, and there is no interaction between the designers, i.e., $n_{\mathrm{r}} = 2$, $n_{\mathrm{s}} = 2$, $n_{\mathrm{i}} = 0$. For D_2 and D_3, the requirements are updated for each design variable before selecting their values, each design variable value is selected once, and there is one interaction in between, i.e., $n_{\mathrm{r}} = 2, n_{\mathrm{s}} = 2, n_{\mathrm{i}} = 1$. For D_4, there is no requirement update,

and it is assumed that each design variable value is readjusted once with respect to the decision for the other design variable value, which requires four interactions to ensure a permissible system design, i.e., $n_r = 0, n_s = 4, n_i = 4$. Note that this assumption does not necessarily reflect the reality and is only made for the purpose of comparison here. Regarding the values of C_i, C_s, and C_r, two different cases are considered:

Case 1: $C_r = C_s = C_i = 1$ where n_i, n_s, and n_r are weighted the same,

Case 2: $C_r = 3, C_i = C_s = 1$ where n_r is weighted three times.

For each problem (a) and (b), the values of $(\mu(D,F), C(D))$ are visualized in Fig. 5 for the two cases depending on the decomposition strategy D. Note that all values for D_2 and D_3 are the same because of the definitions of $\mu(D,F)$ and $C(D)$, i.e., the order of the sequencing of dependent design decisions is irrelevant here.

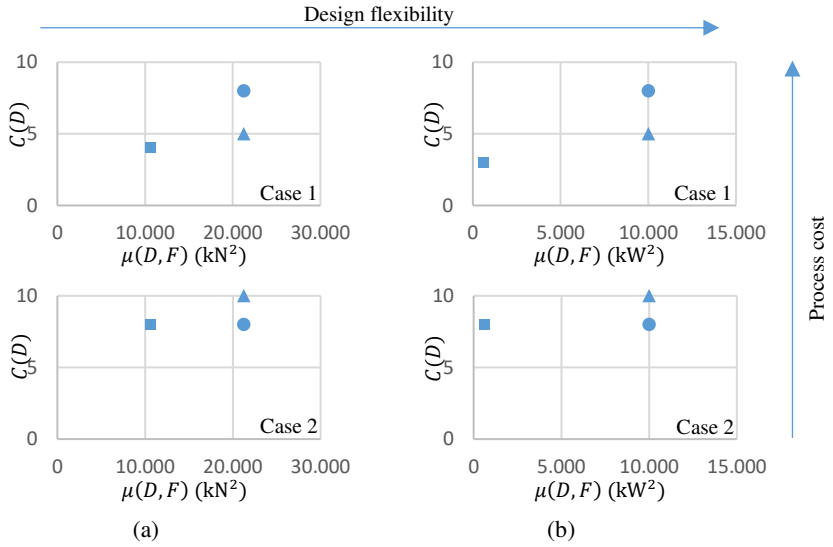


Figure 5. Flexibility and cost $(\mu(D,F), C(D))$ for case 1 (top row) and 2 (bottom row) depending on the decomposition strategy D: independent (D_1, square markers), dependent (D_2 and D_3, triangle markers), and interdependent (D_4, circle markers) for the industrial problems related to crash design (a, left column) and battery pack design (b, right column). Utopia point is bottom right.

For both problems (a) and (b), the decomposition strategies D_1, D_2, and D_3 are Pareto optimal in case 1 with $C(D_1) < C(D_2) = C(D_3) < C(D_4)$ and $\mu(D_2,F) = \mu(D_3,F) = \mu(D_4,F) > \mu(D_1,F)$, and the decomposition strategy D_4 is Pareto optimal in case 2 with $C(D_4) = C(D_1) < C(D_2) = C(D_3)$ and $\mu(D_4,F) = \mu(D_2,F) = \mu(D_3,F) > \mu(D_1,F)$. Using this setting for two design variables, there are always two critical cost thresholds for the requirement updates (C_r) above which interdependent design is preferred against dependent design (threshold 1 for formulae), and preferred against both dependent and independent design (threshold 2 for formulae, see case 2). Below threshold 2, there are always two Pareto optimal decomposition strategies, i.e., it is a trade-off between maximum flexibility (interdependent design, between threshold 1 and 2; dependent design, below threshold 1, see case 1) and minimum process cost (independent design).

The presented results were obtained using strong assumptions about the cost coefficients C_r, C_s and C_i, and the number of required design actions n_r, n_s and n_i. In particular the latter will probably be highly dependent on the actual design problem under consideration. A strongly nonlinear problem subject to many constraints with an intricate solution space geometry is more likely to require a large n_s and n_i. This is currently being explored in parallel work, see e.g. (Wöhr et al., 2020). Regardless of the concrete numbers, these two very simple examples show a mechanism that makes the most structured design approach, i.e., independent design, preferable: when the cost of iteration and interaction outweighs the cost of requirement formulation and loss flexibility due to decomposition. The most unstructured approach, i.e. interdependent design, is preferable, when iteration and interaction are cheap.

4 Conclusion

In this paper, a framework to obtain optimal process architectures for a solution-space-based design process was proposed. Here, feasible sequencings of independent, dependent, and interdependent design decisions are viewed as decomposition strategies and are assessed. This assessment is done by considering maximum design flexibility and minimum process cost, which yields Pareto optimal decomposition strategies, i.e., Pareto optimal process architectures. The framework was applied to two simple industrial design problems for which a dependence of the Pareto optimal decomposition strategies on the cost of the requirement updates was detected. The general structure of the framework offers its application to more complex design problems with multiple design variables for which more differentiable results are expected, as well. Furthermore, realistic costs of the different decomposition strategies should be investigated, and the quantification of the total design flexibility must be clarified in order to obtain useful Pareto optimal process architectures.

References

Daub, M., 2020. Optimizing Flexibility for Component Design in Systems Engineering under Epistemic Uncertainty. Ph.D. thesis. Munich, Germany.

Daub, M., Duddeck, F., 2019. Maximizing Flexibility for Complex Systems Design to Compensate Lack-of-Knowledge Uncertainty. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems: Part B. Mechanical Engineering 5 (4), 41008.

Daub, M., Duddeck, F., Zimmermann, M., 2020. Optimizing Component Solution Spaces for Systems Design. Structural and Multidisciplinary Optimization.

Devendorf, E., Lewis, K., 2011. The Impact of Process Architecture on Equilibrium Stability in Distributed Design. J. Mech. Des. 133 (10), 1.

Fender, J., Duddeck, F., Zimmermann, M., 2016. Direct computation of solution spaces. Struct Multidisc Optim 55 (5), 1787–1796.

Funk, M., Jautze, M., Strohe, M., Zimmermann, M., 2019. Sequential Updating of Quantitative Requirements for Increased Flexibility in Robust Systems Design. Proc. Int. Conf. Eng. Des. 1 (1), 3531–3540.

Papalambros, P.Y., Wilde, D.J., 2017. Principles of optimal design: Modeling and computation. Cambridge University Press, Cambridge, UK, 1 volume.

Pimmler, T.U., Eppinger, S.D., 1994. Integration analysis of product decompositions. Proceedings of the 6th International Conference on Design Theory and Methodology Conference, Minneapolis, USA, September 1994.

Vogt, M.E., Duddeck, F., Wahle, M., Zimmermann, M., 2018. Optimizing tolerance to uncertainty in systems design with early- and late-decision variables. IMA Journal of Management Mathematics 30 (3), 269–280.

Wöhr, F., Königs, S., Ring, P., Zimmermann, M., 2020. A Role-Activity-Product Model to Simulate Distributed Design Processes. Proceedings of the 22nd International DSM Conference (DSM2020), Montreal, Canada.

Zimmermann, M., Hoessle, J.E. von, 2013. Computing solution spaces for robust design. Int. J. Numer. Meth. Engng 94 (3), 290–307.

Contact: M. Daub, Technische Universität München, Arcisstr. 21, 80333 Munich, Germany, marco.daub@tum.de, ferdinand.woehr@tum.de, zimmermann@tum.de