INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED23
24-28 JULY 2023, BORDEAUX, FRANCE

SUPPORTING CHANGEABILITY QUANTIFICATION IN PRODUCT-SERVICE SYSTEMS VIA CLUSTERING ALGORITHM

Machchhar, Raj Jiten;
Aeddula, Omsri Kumar;
Bertoni, Alessandro;
Wall, Johan;
Larsson, Tobias

Blekinge Institute of Technology

ABSTRACT

The design of Product-Service Systems (PSS) is challenging due to the inherent complexities and the associated uncertainties. This challenge aggravates when the PSS being considered has a longer lifespan, is expected to encounter a dynamic context, and integrates many novel technologies. From systems engineering literature, one of the measures for mitigating the risks associated with the uncertainties is incorporating means in the system to change internally as a response to change externally. Such systems are referred to as value-robust systems, and their development largely relies on Tradespace exploration and synthesis. Tradespace exploration and synthesis can be challenging and a time-consuming task due to dimensionality. In this light, this paper aims to present an approach that enables the population of the Tradespace and then, supports the synthesis of such a Tradespace using a clustering algorithm for support changeability quantification in PSS. The proposed method is also implemented on a demonstrative case from the construction machinery industry.

Keywords: Product-Service Systems (PSS), Systems Engineering (SE), Decision making, Changeability quantification, Early design phases

Contact:
Machchhar, Raj Jiten
Blekinge Institute of Technology
Sweden
raj.jiten.machchhar@bth.se

Cite this article: Machchhar, R. J., Aeddula, O. K., Bertoni, A., Wall, J., Larsson, T. (2023) 'Supporting Changeability Quantification in Product-Service Systems via Clustering Algorithm', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.323

1 INTRODUCTION

It is widely acknowledged that the increased competition and volatile market have pushed the manufacturing industries to a servitized business model. Academic literature describes this transition towards servitization from offering pure products as offering Product-Service Systems (PSS) (Pirola et al., 2020). Product-Service Systems (PSS) design inherently deals with high complexity due to its multidimensionality (Mourtzis et al., 2018). This challenge aggravates when the PSS being considered has a longer lifespan, is expected to encounter a dynamic context, and integrates many novel technologies. Literature in systems engineering (SE) and PSS agree that the development of complex solutions deal with many uncertainties during the design stage (Erkoyuncu et al., 2011; McManus and Hastings, 2005). A strategy to mitigate the risks associated with such complexity is system architecting, where one of the goals is generalizing a solution to be valuable under external changes in the form of requirements or contexts (Crawley et al., 2016; Mekdeci et al., 2012). The system is inherently value-robust if the value is largely unaffected by such external changes. Else, this goal is achieved by incorporating means and mechanisms in the design to change either the form or function under value-deteriorating events during its lifecycle. A system property that can alter its state as a response to value-deteriorating events is generically referred to as "changeability" (Ross et al., 2008).

The theme of changeability and its quantification is one of the core aspects of PSS development (Bertoni and Bertoni, 2019). This argument is partly strengthened by the fact that a PSS is typically subjected to several external changes (such as requirements, users, environment, legislation, etc.) and partly by the fact that the incentive of intentionally developing a value-robust PSS is higher as the provider in leasing-oriented business models retains the ownership. One of the methods to deal with changeability quantification is "Filtered Outdegree" (Ross et al., 2008), which is a measure of how changeable a system is based on the number of acceptable paths the system can utilize to incur a state change. This method is considered viable for developing value-robust PSS, where change capability is attributed to the entire domain of PSS elements such as products, services, and infrastructure (Machchhar and Bertoni, 2022). Filtered Outdegree relies on Tradespace exploration and synthesis, that is, a systematic calculation of the value metrics in the form of benefit/utility versus costs for changing externalities (Rhodes and Ross, 2010). Tradespace exploration and synthesis can be challenging and time-consuming (Specking et al., 2019). The open-ended problem of assessing and understanding multiple system-context interactions in changing requirements and contexts, along with extracting the desired knowledge from much simulation data poses a significant challenge to the decision-makers. For example, many context variations would generate a multitude of Tradespaces that would need additional support for knowledge extraction. State-of-the-art machine learning techniques mainly support decision-making rather than changeability quantification (Das and Pratihar, 2019), while dedicated information visualization (Midway, 2020), such as graphs, networks, heatmaps, etc., do not confer the desired level of support for changeability quantification. The main reason is that systems as complex as PSS, undergoing many requirements and contextual changes in their operational phase, need a more concrete approach to understand their value under all external changes.

Overall, a value-robust PSS entails a better market success probability rendering changeability quantification a practical design decision-making aspect, especially in the early design phase because much of the resources are committed during that phase. Understanding the overall value of a PSS and its variance with external changes in a Tradespace of several possible designs from a range of design variables is challenging, mainly due to dimensionality. Clustering algorithms have been shown to enhance Tradespace synthesis in multiple ways, for instance, Pareto space reduction by creating a family head for the given solution (Zio and Bazzo, 2011), design space reduction for an effective metamodeling routine (Qiu et al., 2016), or establishing crucial input-output relationships from a multiobjective problem (Das and Pratihar, 2019). Thus, clustering can complement Tradespace synthesis, but it has not been thoroughly investigated for assisting changeability quantification. In this light, this paper aims to:

- Present an approach that enables the population of the Tradespace in the early design stages of a PSS and then supports its synthesis using a clustering algorithm based on user inputs.
- Demonstrate applicability of the approach on a case study in the construction machinery industry.

The paper is structured as follows: Section 2 presents the research method, section 3 briefly overviews the related literature, section 4 details the proposed approach supporting changeability quantification, section 5 demonstrates the proposed method on a construction machinery industry vehicle, and finally, section 6 concludes the paper with highlighting the prospects.

2 RESEARCH METHOD

The work presented in this paper is an outcome of Participatory Action Research (PAR) (Avison et al., 1999). Multiple data collection strategies were applied in collaboration with the industrial partners in the construction equipment manufacturing sector to understand the needs of electrified future mining sites. Both real-time and retrospective methods were utilized (Blessing and Chakrabarti, 2009). Regarding real-time, data were mainly gathered by visiting a mining site to comprehend the functioning of different aspects of a mining scenario. Notes were taken while the site was showcased, and the challenges were discussed. Regarding retrospective methods, the existing literature in the domain of PSS and SE was reviewed in narrative style due to the vastness of these streams, finding synergies between both disciplines, especially concerning changeability and its quantification. Snowballing has been particularly helpful during the literature review due to the breadth of these fields. Besides literature, simulation data has been one of the cores of the results presented in this paper. Also, data were collected via semi-structured interviews with company partners along with referring to internal documents and documents available on the web pages. These data were supplemented by regular bi-weekly virtual meetings where the issues were exemplified further. These meetings also served as a basis for demonstrating preliminary results of the prescribed "support" (Blessing and Chakrabarti, 2009) to verify the results from a logic and consistency point of view and gain valuable feedback. This process was repeated iteratively, allowing the researchers to close the look-think-act learning circles prevalent in PAR. From a design research methodology perspective (Blessing and Chakrabarti, 2009), this research is partially descriptive and partially prescriptive in nature based on the need and expectation findings.

3 CHANGEABILITY QUANTIFICATION IN PRODUCT-SERVICE SYSTEMS

The notion of value is considered the core of decision-making concerning the development of PSS (Rondini et al., 2020). Literature is PSS and SE may have a different taxonomy for defining value, but the central idea is the same. Value is seen as a ratio of benefits and costs in PSS literature (Rondini et al., 2020), while a ratio of utility and cost in SE literature (Ross et al., 2008), and this definition of value is preserved in this work. As mentioned previously, the development of a PSS inherits many complexities that need to be addressed during the design phase. These complexities are distinguished as internal complexity and external complexity based on the system boundary definition (Heydari and Herder, 2020). Internal complexity is associated with the artifact being developed, while external complexity is associated with various contexts in different scenarios. Both these complexities must be balanced so that the solution is not under-prepared to respond to external value-deteriorating events and not over-prepared, leading to unnecessary costs (Heydari and Herder, 2020). To address the complexity aspects of design, McManus and Hastings (2005) presented one of the first frameworks for understanding the uncertainties associated with the complexities and the strategies to mitigate them. These strategies are typically system properties that provide means for sustaining value over the lifetime, and changeability is one such property that can be applied to both, form and function. As mentioned before, a method for changeability quantification is Filtered Outdegree, assessed by counting the number of possible transition path the system can adopt to reach a new state in a Tradespace. Tradespace can be multidimensional for gaining better insights about time and change, achieved by Epoch-Era Analysis. Here, Epoch is a specific period where the requirements and context are fixed. A set of such epochs are referred to as an Era that can be utilized for decision-making about future uncertainties.

Dwyer and Efatmaneshnik (2020) argue that Filtered Outdegree is still one of the primary methods for changeability quantification and mainly relies on Tradespace exploration and synthesis, further proposing an enhancement by the "Dijkstra" algorithm in the quantification by finding the shortest distance between nodes in the Tradespace. Several methods are also proposed in the literature for a better exploration and synthesis of the Tradespace that involves some form of machine learning. On the exploration side, surrogate modeling is a common technique supporting dimensionality reduction, projection-based modeling, sensitivity analyses, and optimization (Yondo et al., 2018). On the synthesis side, methods are more oriented towards analysis of the Tradespace either in the form of Pareto or complete Tradespace to gain comprehensible insights about the design variables, dominantly using clustering algorithms. For instance, Zio and Bazzo (2011) proposed a clustering procedure on full

Tradespace to reduce the design options. Das and Pratihar (2019) used a neuro-fuzzy system to establish the relationship between the design variables and objectives after getting an initial set of Pareto front. Clustering is a technique of categorizing numerical data into multiple mutually exclusive clusters or groups. Various techniques can be employed for clustering; the most used methods are K-means, hierarchical method, and neural networks. Each method has pros and cons based on data characteristics like skewness, the number of clusters, sample size, error in data, etc. K-means is a centroid-based method, and it is observed to have an optimal performance compared to other methods for variations in the number of clusters concerning convergence and complexity (Kumar and Reddy, 2017). However, all these techniques are mainly geared toward design decision support rather than changeability quantification. In this work, K-means clustering has been utilized with the purpose of categorizing the simulation data into K mutually exclusive clusters. Such a categorization is achieved by an iterative process that minimizes the sum of distances from each data point to its centroid point until no further sums are possible to obtain the set of clusters.

4 A METHOD FOR SUPPORTING CHANGEABILITY QUANTIFICATION

SE literature often refers to the "development control" (Kossiakoff and Sweet, 2003) of the engineering team as a means for establishing a boundary between the system and the context. Development control implies that all the aspects that the engineering team cannot control but still influence the operation of the system are referred to as context. Aligning with this perspective, an internal change is a change within the system in response to external changes such as context. With such a distinction, an operational scenario is a reference frame of the system and the changing contexts. In the design stages, the changing contexts are selected based on the likelihood of occurrence and not as a forecast. Also, the state of any system is a function of design variables. In optimization runs, a disciplinary analysis computes the responses of the system, which are the states of the system (Martins and Ning, 2021). A design variable can either define a system's form (configuration) or function (control input). The control policy is a series of such control inputs (Bertsekas, 2019). The state of a system in each context is, thus, a function of system configuration and control policy. During optimization studies, depending on the problem formulation and the method of aggregation of system responses as a time history, a control input may not be explicitly necessary at each time step. However, it is argued that a simultaneous evaluation of different configurations and control policies is necessary for a better understanding of system-context interactions. In this light, the Tradespace shall comprise design points that may have the same configuration but vary atleast in the control policy. Also, design variables can be referred to as configuration variables and control variables for an explicit distinction in this work.

The proposed method for supporting changeability quantification has been illustrated in modules A and B (refer to Figure 1 and Figure 2), where the blue boxes represent the steps, and the grey boxes represent user inputs. Besides facilitating a thorough comprehension of the proposed method, the reason for such a split is that module A is optional and can be skipped if the Tradespace has been generated previously. In all other cases, generating such a Tradespace is an engineering problem, and design optimization (DO) is one of the prominent methods that enables finding viable solutions under a variety of necessary conditions (Martins and Ning, 2021). In this work, the problem is multiobjective, contains discrete variables, is believed to be multimodal, and has function discontinuities. Thus, an evolutionary algorithm is utilized to solve a multiobjective problem. To simulate different control policies, dynamic programming (DP) (Bertsekas, 2019) is a proven technique that enables benchmarking different policies based on value or cost function. Simulation of control policies is necessary as some contextual changes can be responded to by a change of control policy rather than a change of configuration under the pretext that configuration changes are more expensive than control policy changes. However, a combination of DO and DP for generating Tradespace can incur high computational costs. A local optimizer would need multiple iterations with several starting points, while a global optimizer (such as evolutionary algorithms) naturally inherits a longer convergence time (Martins and Ning, 2021). Thus, design space exploration using Design-of-Experiments (DoE) (Yondo et al., 2018) has been suggested and utilized in this work instead of DO. Besides, finding the global optimum is seldom the goal in early design when many uncertainties exist. However, the proposed DoE routine can be replaced by a full DO run, but in such a case, it is advised to store the iteration history data for a more robust quantification. The step numbers indicated in yellow circles in Figure 1 and Figure 2 have been detailed as follows:

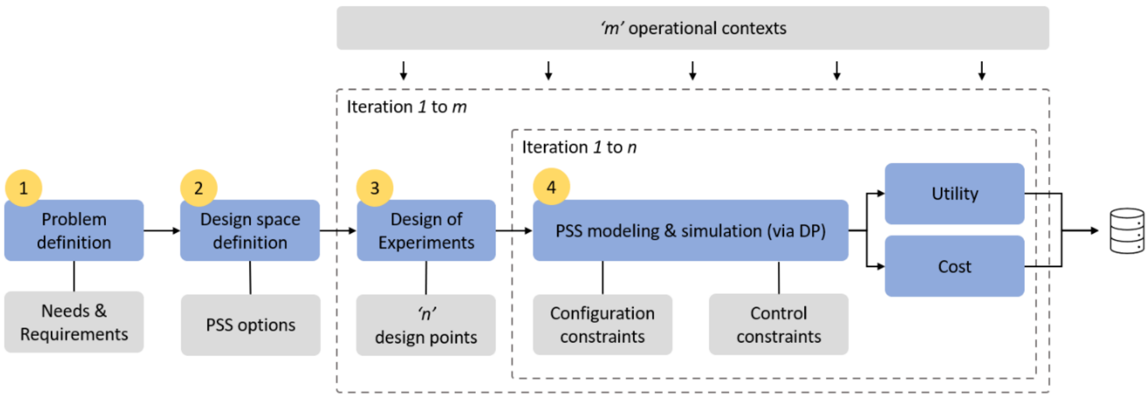

Figure 1: Module A of the proposed method, geared towards Tradespace exploration.

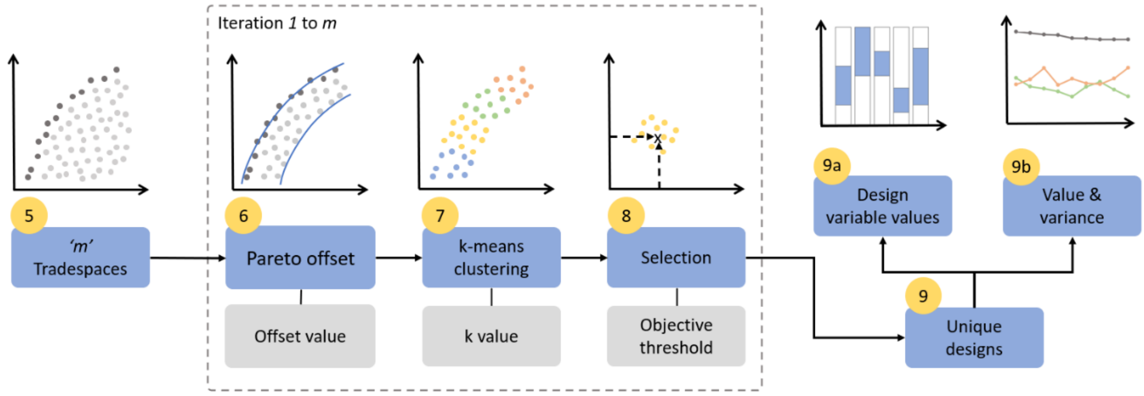

Figure 2: Module B of the proposed method, geared towards Tradespace synthesis.

- Step 1: The first step is contextualizing the problem based on the needs and requirements. For instance, if utility theory is used for deriving the value of a solution, agreeing upon what constitutes higher utility along with stakeholders' subjectivities is a part of this step.
- Step 2: In this step, the design space is defined. One of the crucial aspects of this step is to assign bounds to configuration, control, and state variables for generating a confined Tradespace, where the state is a function of the design variables (configuration + control variables). Also, a set of contexts are defined to represent an Era (Rhodes and Ross, 2010) and supplied in this step, resulting in *m* iterations for *m* operational contexts.
- Step 3: Based on the input of required design points *n*, an automated DoE sub-routine exploits the Latin hypercube sampling (LHS) method to generate sample points across the design space ready for simulation. Each context supplied in this step represents an Epoch (Rhodes and Ross, 2010).
- Step 4: Here, the PSS is modeled and then simulated via DP to generate utility vs. cost trade-offs based on input configuration and control constraints. A simulation is performed for the m^{th} context and n^{th} design point. The cost function can have various formulations, such as deterministic, stochastic, min-max, etc., based on the application (Bertsekas, 2019). Constraint violations are treated by a penalty-based approach.
- Step 5: Each simulation result is stored in a database. This database eventually comprises *m* Tradespaces for *m* contexts, and each Tradespace comprises *n* designs for *n* design points.
- Step 6: Each Tradespace is fed to the clustering sub-routine that is iterated for *m* Tradespaces. The Pareto Front is first recognized by finding the non-dominated solutions. Since the Tradespace was generated via DoE, the Pareto may not be (truly) non-dominated. An approximate model of the Pareto Front is built using regression analysis to assess its nature, and an offset of this model enables the user to select a band of designs in the Tradespace. Figure 2 exemplifies how such a model would look for a two-objective DO problem (utility and cost). Such a band enables a more robust changeability quantification support by involving more design points and, at the same time, reduces the design space to a comprehensible limit. For instance, a dominant design point may be a better choice of design cumulatively under an Era than a non-dominant design for a given Epoch. Thus, offset value is a user input based on many factors, such

https://doi.org/10.1017/pds.2023.323 Published online by Cambridge University Press

as the confidence on Pareto, the resolution of quantification needed, the desired level of reduction in the design space, etc.

- Step 7: The band of design points generated by Pareto offset is fed to a clustering algorithm to cluster designs based on the desired number of clusters k as user input. Here, $1 < k < m$ for all iterations as $k = 1$ represents the Tradespace itself, and $k = m$ represents each design point in the Tradespace. The input of k value is a strategic decision based on factors such as business portfolio, platform-based offerings, etc.
- Step 8: A suitable cluster is selected to support changeability quantification once the clusters are formed. The user input of desired objective threshold (utility or cost in this case) serves as a basis for the selection of a cluster. Centroid positions for each cluster are extracted, and based on this input, the centroid that minimizes the distance to the indicated objective value is selected.
- Step 9: Once all the iterations are performed, clusters from each iteration are combined to represent a set of unique design points by eliminating repetitive points. Following, all the unique design points can be re-routed to step 4, which is the modeling and simulation of the PSS for each context to generate a matrix of m x n x *objectives* dimensions. Based on Value-driven design (VDD) literature (Collopy and Hollingsworth, 2011), the *objective* values can be converted into a single scalar value for each design point quantitatively or qualitatively. Such a conversion enables direct comparison of different design points, and the matrix is squeezed to m x n dimensions. This matrix now represents a set of designs that fulfill the desired input objective with the best possible trade-offs with respect to other objectives as the design points were extracted close to the Pareto Front. Such a matrix can be used in two aspects:
  - Step 9a: From the matrix, a bar graph can be plotted to indicate which design variables are essential for achieving the desired value. The design variables shall be normalized, as seen in Figure 2, to visualize all variables at once. This bar graph serves as a basis for changeability quantification by raising awareness about the sensitivities of all design variables for achieving the desired objective with the best possible trade-offs in an Era. Also, a design variable can be assigned a fixed metric, and the change in expected value can be analyzed.
  - Step 9b: A VDD-based weighted sum of value across an Era highlights the "best" design point to the design team. However, value-robustness implies minimizing negative value variance and maximizing positive value variance compared to a benchmark, i.e., minimizing and maximizing value depreciation and appreciation due to external changes, respectively. Thus, as seen in Figure 2, three graphs are plotted, value and variance to support changeability quantification.

5 DEMONSTRATIVE CASE

The proposed method was tested on a battery electric hauler, an automotive vehicle designed to transfer mined material from one place to another. As the mining industry propels towards autonomy and electrification, a lot of uncertainty exists concerning the value of such haulers over time. On the one hand, established approaches to system architecting are being challenged by novel technology, but on the other hand, autonomy confers the opportunity to optimize to the next level as human-oriented uncertainties are reduced. Besides, with the increasing servitization of these industries, these industries lease a pool of machines for specific periods to achieve functional requirements. As soon as the requirement is changed, the pool of machines is changed to best suit the purpose. A persistent question exists whether these machines could have been modified on-site to fulfill the new requirements. A mining site has an extensively long operational life, and in use- or result-oriented business models, a hauler can be commissioned to several operational sites, where it undergoes many contextual changes.

This work considers five such operational scenarios as a demonstrative case. These operational scenarios were modeled in a Discrete-event Simulation environment to extract relevant contextual variables within an operational scenario. For instance, there is an extensive list of contextual changes in a mining operational scenario, such as vehicle path properties, ambient temperature, locations of infrastructure such as crushers, charging stations, etc., infrastructure capabilities, maintenance competence, pre-existing machines such as excavators, wheel loaders, etc., stockpile properties, and so on. For this study, only vehicle path change is considered. With two possible path alternatives for each operational scenario, the so-called Era comprises ten context changes. Path properties have a notable effect on hauler performance. Aspects such as topography, speed limits, halt, etc. influence energy consumption, and the design challenge is understanding which hauler configuration sustains value under such external

https://doi.org/10.1017/pds.2023.323 Published online by Cambridge University Press

changes, i.e., the one that is value-robust. Figure 3 exemplifies different types of path elevation profiles, normalized such that the lowest elevation is zero despite being below sea level. Referring to Figure 3, types 1 and 2 are predominantly hilly, but one key difference is that type 1 is of regenerative kind (material is extracted at an elevation), whereas type 2 is of degenerative kind (material is transferred to an elevation). Type 3 is predominantly flat, while type 4 is a mix of type 1 and 2.

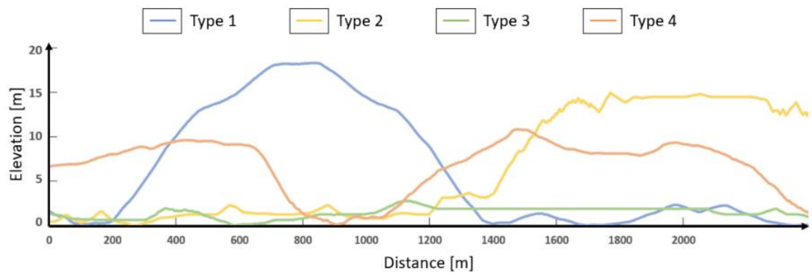

Figure 3: Path elevation profile for a fraction in different operational scenarios

To design a value-robust hauler, a combination of design variables catering to configuration and control policies was chosen. Those include the payload capacity, motor types, gearbox types, and battery capacity from a configuration perspective and a sequence of accelerating/braking torque and gear selection from a control perspective. A fit-to-purpose vehicle dynamics model was developed to simulate a two-axle vehicle in forward and backward motion with reasonable accuracy. Essentially, the vehicle must overcome inertial, gradient, rolling, and aero resistance during its operation, and based on these resistances, the force supplied by the motor results in acceleration or deceleration of the vehicle. In this work, a "quasistatic" (Guzzella and Sciarretta, 2007) approach was utilized, where the path is discretized into small instances, and the vehicle state and path properties are assumed to be constant in a particular instance. This discretization enables solving the problem numerically via DP but demands a finer resolution for better accuracy.
To simulate the hauler's operation, deterministic DP (Bertsekas, 2019) was implemented to find the optimal velocity for the hauler under the given context. In DP, as the vehicle transitions to the next state, a cost function can be used to minimize the instantaneous step cost and, in this case, a balance was struck between energy consumption and time required for completing a specified distance. Such a balance is necessary as only energy minimization would mean that the vehicle never accelerates, and only time minimization would mean that the vehicle never decelerates. Hence, a time penalty factor was introduced as a control variable that decides the balance between energy and time. The sequence of accelerating/braking torque and gear selection is, thus, a function of the time penalty factor. Finally, numerous constraints can be applicable at a configuration level to ensure a proper fit of sub-systems and at a control level to ensure proper vehicle performance on its path, as listed in (Ghandriz et al., 2020). Relevant constraints were elicited and applied for this simulation. Two objectives were defined, utility and cost. The utility function was decided to be the time required to complete the transportation of a fixed amount of ore, while the cost function comprises the fixed cost of the asset along with the operational costs that are simplified based on inputs from company partners. Exemplary Tradespace for the hauler in three reference contexts has been shown in Figure 4 by utilizing Module A.

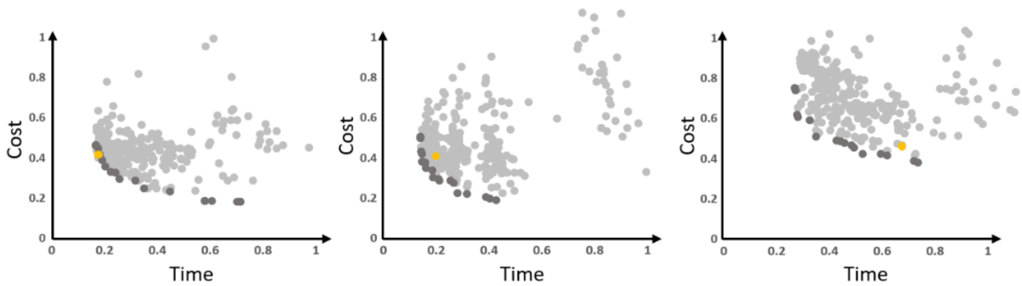

Figure 4: Exemplary Tradespace for three reference contexts as per Module A.

Consider a case when the business requirement is to achieve a function (i.e., transport a certain amount of material) within a specific time. Thus, the user inputs the objective time threshold as explained in the steps for Module B along with Pareto offset and clustering values. If the user chooses just one

https://doi.org/10.1017/pds.2023.323 Published online by Cambridge University Press

"best" design for achieving that function, it might perform superiorly in the given context but might underperform as soon as the context changes (see the yellow point in Figure 4). If the user chooses one "best" design in each context and then combines them to represent a platform of many designs as a changeable solution, then the design variable bounds can be extensively large, incurring unnecessary investment costs. The input of Pareto offset value and clustering value confine the Tradespace, yet allow for the inclusion of more design points, eventually reducing the risk of eliminating a dominated design point that could deliver a better value overall in an Era. Figure 5 illustrates part of the steps in Module B for a single context. Essentially, the Tradespaces are fed to Module B in iterations, where Pareto Front from each Tradespace is extracted, and the design space is reduced by offsetting it. Then, the band of design points is clustered (five in this case) based on the user input. Finally, a suitable cluster is selected based on the centroid value that is the closest to the time threshold input.

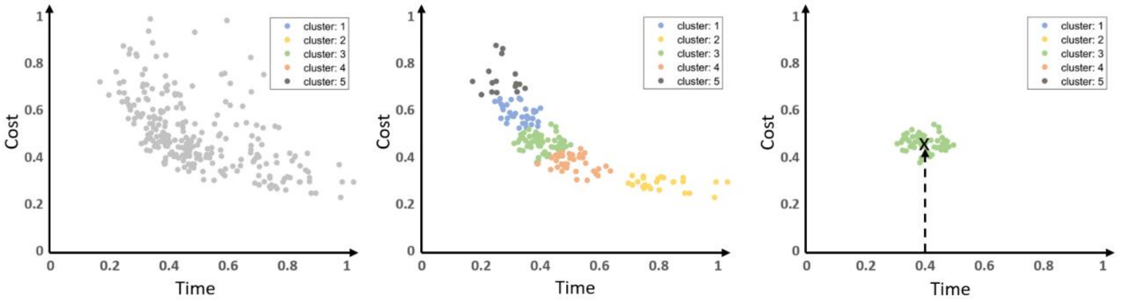

Figure 5: Illustration of Pareto offset, clustering, and selection as explained in Module B

By combining all the clusters and eliminating all the repetitive design points, a set of unique design points is achieved that fulfills the desired time threshold. By re-routing these through step 4 in Figure 1, the value for each design for all given contexts can be calculated. Figure 6 shows exemplary results of executing Module B. Two bar graphs, A & B, have been plotted (left), showing which design variable values are crucial for different objective thresholds (simulating the impact of change in requirements). Intuitively, as the user input for the time threshold increases, it can be inferred that the payload capacity must increase from both the bars plots.

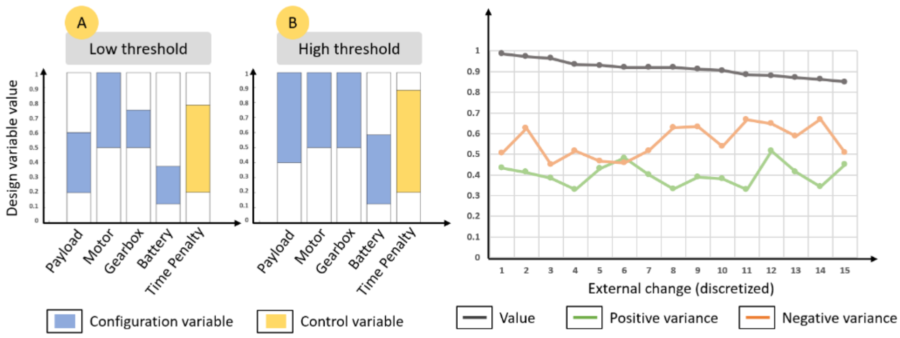

Figure 6: Exemplary design bounds (left) and cumulative value and its variance (right)

Three graphs are plotted as shown in Figure 6 (right); one of them resembles the value of a design point as a normalized qualitative sum in an Era, while the other two are normalized positive and negative value variances. To support changeability quantification, one of the design variables can be fixed to a single value, and the effect on value can be visualized. Typically, any constraint on the design variable variability will reduce the value. From a design point of view, a design that delivers superior value in one context but significantly struggles to deliver value in another is not a great choice due to the associated risks (Erkoyuncu et al., 2011). A value-robust design point would be the one that has min. negative value variance, max. positive value variance, while maintaining a relatively higher value. The graph of value arranged in descending order enables a direct comparison of value variances that can be an avenue for decision-making concerning changeability. For instance, design point 6 has a lesser negative variance, higher positive variance, and maintains a relatively higher value, despite not being the "best" design point. Such designs can be good discussion points for further development. These graphs update based on the changed design variable values and objective thresholds, eventually supporting decisions concerning changeability.

https://doi.org/10.1017/pds.2023.323 Published online by Cambridge University Press

6 CONCLUDING REMARKS

This paper proposes a method for supporting changeability quantification in PSS in two modules. The first module enables the population of the Tradespace by exploiting the LHS method, and the other module synthesizes the Tradespace by exploiting a clustering algorithm. The proposed method can be positioned as addressing the perceptual complexity in design decision-making (Rhodes and Ross, 2010). Also, the method was practically implemented on a battery electric hauler with two objectives: utility and costs. For an effective change quantification, the design variables need to be characterized in terms of the cost of investment and change (Ross et al., 2008; Rehn et al., 2019). At this juncture, the proposed method raises awareness of which design variables are sensitive to value sustainment. It further guides design decisions on changeability by allowing the engineering team to eliminate design variations and visualizing the impact of value and its variance. Also, concerning value variance, the risks must be within acceptable limits (Erkoyuncu et al., 2011). However, weighing the risks in terms of impact and likelihood has not been treated in the proposed method. Characterizing the design variables and characterizing the risks shall be seen as the next step for enhancing the proposed method.

The proposed method is best suited for discrete variables, and thus it is recommended that continuous design variables be discretized. One of the key advantages of doing so is a reduction in the number of possible designs, especially the redundant ones that show marginal benefit over the adjacent ones. Besides, since the proposed method is best suited for early design decision-making, such a minuscule resolution is seldom a requirement. On contextual changes, however, the proposed method was demonstrated for a context change in only one dimension, i.e., the path properties. There can be enormous contextual changes that a system deals with along the lifecycle (Mekdeci et al., 2012; Rhodes and Ross, 2010). Simultaneous simulation of configuration and control variables can lead to high computational complexity due to DP. However, it is argued that such simultaneous simulation is necessary for understanding which configuration changes can be readily addressed by merely a control change that is often cost-effective. Certainly, value space or policy space approximation by techniques such as approximate-DP/reinforcement learning (Bertsekas, 2019) can be a valuable enhancement to the proposed method. On computational complexity, the LHS method was exploited to populate the Tradespace, and thus, despite referring to the "best" design as Pareto, those designs may not (truly) represent a Pareto Front. Populating the Tradespace with LHS can be tricky initially, as choosing the number of design points for the given application is left to the user. Techniques such as sequential sampling (Yondo et al., 2018) can certainly assist in circumventing this issue.

ACKNOWLEDGMENTS

The work was performed partially in the frame of the TRUST-SOS project funded by the Swedish Innovation Agency (VINNOVA) through the FFI Fossil-free mobile work machine initiative. The work has also been performed in the frame of the Strategic Innovation Program Swedish Mining Innovation, concurrently funded by the Swedish Innovation Agency (VINNOVA), the Swedish Research Council for Sustainable Development (FORMAS), and the Swedish Energy Agency (Energimyndigheten).

REFERENCES

Avison, D.E., Lau, F., Myers, M.D. and Nielsen, P.A. (1999), “Action research”, *Communications of the ACM*, Vol. 42 No. 1, pp. 94–97. https://doi.org/10.1145/291469.291479

Bertoni, A. and Bertoni, M. (2019), “Modeling ‘ilities’ in early Product-Service Systems design”, *Procedia CIRP*, Vol. 83, pp. 230–235. https://doi.org/10.1016/j.procir.2019.03.091

Bertsekas, D. (2019), *Reinforcement Learning and Optimal Control*, Athena Scientific.

Blessing, L.T. and Chakrabarti, A. (2009), DRM: *A Design Research Methodology*, Springer.

Collopy, P.D. and Hollingsworth, P.M. (2011), “Value-Driven Design”, *Journal of Aircraft*, American Institute of Aeronautics and Astronautics, Vol. 48 No. 3, pp. 749–759. https://doi.org/10.2514/1.C000311

Crawley, E., Cameron, B. and Selva, D. (2016), *System Architecture: Strategy and Product Development for Complex Systems*, Pearson.

Das, A.K. and Pratihar, D.K. (2019), “A novel approach for neuro-fuzzy system-based multi-objective optimization to capture inherent fuzziness in engineering processes”, *Knowledge-Based Systems*, Vol. 175, pp. 1–11. https://doi.org/10.1016/j.knosys.2019.03.017

Dwyer, D.M. and Efatmaneshnik, M. (2020), “Changeability analysis for existing systems”, *Australian Journal of Multi-Disciplinary Engineering*, Vol. 16 No. 1, pp. 43–53. https://doi.org/10.1080/14488388.2020.1781345

Erkoyuncu, J.A., Roy, R., Shehab, E. and Cheruvu, K. (2011), “Understanding service uncertainties in industrial product–service system cost estimation”, *The International Journal of Advanced Manufacturing Technology*, Vol. 52 No. 9, pp. 1223–1238. https://doi.org/10.1007/s00170-010-2767-3

Ghandriz, T., Jacobson, B., Laine, L. and Hellgren, J. (2020), “Impact of automated driving systems on road freight transport and electrified propulsion of heavy vehicles”, *Transportation Research Part C: Emerging Technologies*, Vol. 115, p. 102610. https://doi.org/10.1016/j.trc.2020.102610

Guzzella, L. and Sciarretta, A. (2007), *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*, Springer Science & Business Media.

Heydari, B. and Herder, P. (2020), “Technical and Social Complexity”, in Maier, A., Oehmen, J. and Vermaas, P.E. (Eds.), *Handbook of Engineering Systems Design*, Springer International Publishing, Cham, pp. 1–30. https://doi.org/10.1007/978-3-030-46054-9_9-1

Kossiakoff, A. and Sweet, W.N. (2003), *Systems Engineering: Principles and Practices*, Wiley Online Library.

Kumar, K.M. and Reddy, A.R.M. (2017), “An efficient k-means clustering filtering algorithm using density based initial cluster centers”, *Information Sciences*, Vol. 418–419, pp. 286–301. https://doi.org/10.1016/j.ins.2017.07.036

Machchhar, R.J. and Bertoni, A. (2022), “Designing Value-Robust Product-Service Systems by Incorporating Changeability: A Reference Framework”, *Collaborative Networks in Digitalization and Society 5.0*, Springer International Publishing, Cham, pp. 623–630. https://doi.org/10.1007/978-3-031-14844-6_50

Martins, J.R.R.A. and Ning, A. (2021), *Engineering Design Optimization*, 1st ed., Cambridge University Press. https://doi.org/10.1017/9781108980647

McManus, H. and Hastings, D. (2005), “A framework for understanding uncertainty and its mitigation and exploitation in complex systems”, Vol. 15, presented at the INCOSE international symposium, Wiley Online Library, pp. 484–503.

Mekdeci, B., Ross, A.M., Rhodes, D.H. and Hastings, D.E. (2012), “A taxonomy of perturbations: Determining the ways that systems lose value”, 2012 *IEEE International Systems Conference SysCon* 2012, pp. 1–6. https://doi.org/10.1109/SysCon.2012.6189487

Midway, S.R. (2020), “Principles of Effective Data Visualization”, *Patterns*, Vol. 1 No. 9, p. 100141. https://doi.org/10.1016/j.patter.2020.100141

Mourtzis, D., Fotia, S., Boli, N. and Pittaro, P. (2018), “Product-service system (PSS) complexity metrics within mass customization and Industry 4.0 environment”, *The International Journal of Advanced Manufacturing Technology*, Vol. 97 No. 1, pp. 91–103. https://doi.org/10.1007/s00170-018-1903-3

Pirola, F., Boucher, X., Wiesner, S. and Pezzotta, G. (2020), “Digital technologies in product-service systems: a literature review and a research agenda”, *Computers in Industry*, Vol. 123, p. 103301. https://doi.org/10.1016/j.compind.2020.103301

Qiu, H., Xu, Y., Gao, L., Li, X. and Chi, L. (2016), “Multi-stage design space reduction and metamodeling optimization method based on self-organizing maps and fuzzy clustering”, *Expert Systems with Applications*, Vol. 46, pp. 180–195. https://doi.org/10.1016/j.eswa.2015.10.033

Rehn, C.F., Pettersen, S.S., Garcia, J.J., Brett, P.O., Erikstad, S.O., Asbjørnslett, B.E., Ross, A.M., et al. (2019), “Quantification of changeability level for engineering systems”, *Systems Engineering*, Vol. 22 No. 1, pp. 80–94. https://doi.org/10.1002/sys.21472

Rhodes, D.H. and Ross, A.M. (2010), “Five aspects of engineering complex systems emerging constructs and methods”, 2010 *IEEE International Systems Conference*, pp. 190–195. https://doi.org/10.1109/SYSTEMS.2010.5482431

Rondini, A., Bertoni, M. and Pezzotta, G. (2020), “At the origins of Product Service Systems: Supporting the concept assessment with the Engineering Value Assessment method”, *CIRP Journal of Manufacturing Science and Technology*, Vol. 29, pp. 157–175. https://doi.org/10.1016/j.cirpj.2018.08.002

Ross, A.M., Rhodes, D.H. and Hastings, D.E. (2008), “Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value”, *Systems Engineering*, John Wiley & Sons, Ltd, Vol. 11 No. 3, pp. 246–262. https://doi.org/10.1002/sys.20098

Specking, E., Parnell, G., Pohl, E. and Buchanan, R. (2019), “Evaluating a Set-Based Design Tradespace Exploration Process”, *Procedia Computer Science*, Vol. 153, pp. 185–192. https://doi.org/10.1016/j.procs.2019.05.069

Yondo, R., Andrés, E. and Valero, E. (2018), “A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses”, *Progress in Aerospace Sciences*, Vol. 96, pp. 23–61. https://doi.org/10.1016/j.paerosci.2017.11.003

Zio, E. and Bazzo, R. (2011), “A clustering procedure for reducing the number of representative solutions in the Pareto Front of multiobjective optimization problems”, *European Journal of Operational Research*, Vol. 210 No. 3, pp. 624–634. https://doi.org/10.1016/j.ejor.2010.10.021