

Article

Quantitative Set-Based Design to Inform Design Teams

Eric Specking ^{*}, Nicholas Shallcross , Gregory S. Parnell and Edward Pohl

Department of Industrial Engineering, University of Arkansas, Fayetteville, NC 72701, USA; njshallc@uark.edu (N.S.); gparnell@uark.edu (G.S.P.); epohl@uark.edu (E.P.)

* Correspondence: especki@uark.edu; Tel.: +1-479-871-9997

Featured Application: Uses Model-Based Systems Engineering, Set-Based Design, and decision analysis to perform a tradespace exploration on an UAV illustrative case study to show the potential benefits of set-based design.

Abstract: System designers, analysts, and engineers use various techniques to develop complex systems. A traditional design approach, point-based design (PBD), uses system decomposition and modeling, simulation, optimization, and analysis to find and compare discrete design alternatives. Set-based design (SBD) is a concurrent engineering technique that compares a large number of design alternatives grouped into sets. The existing SBD literature discusses the qualitative team-based characteristics of SBD, but lacks insights into how to quantitatively perform SBD in a team environment. This paper proposes a qualitative SBD conceptual framework for system design, proposes a team-based, quantitative SBD approach for early system design and analysis, and uses an unmanned aerial vehicle case study with an integrated model-based engineering framework to demonstrate the potential benefits of SBD. We found that quantitative SBD tradespace exploration can identify potential designs, assess design feasibility, inform system requirement analysis, and evaluate feasible designs. Additionally, SBD helps designers and analysts assess design decisions by providing an understanding of how each design decision affects the feasible design space. We conclude that SBD provides a more holistic tradespace exploration process since it provides an integrated examination of system requirements and design decisions.

Keywords: decision analysis; tradespace exploration; set-based design; team-based methods; systems engineering; model-based engineering



Citation: Specking, E.; Shallcross, N.; Parnell, G.S.; Pohl, E. Quantitative Set-Based Design to Inform Design Teams. *Appl. Sci.* **2021**, *11*, 1239. <https://doi.org/10.3390/app11031239>

Academic Editor: Dov Dori
Received: 18 December 2020
Accepted: 22 January 2021
Published: 29 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The design of complex engineered systems requires detailed analyses performed by a large number of experts over a specific period. A traditional design approach, point-based design (PBD), uses system decomposition, modeling, simulation, optimization, and analysis to find and compare discrete design alternatives. PBD is a well-researched area [1–11]. An alternative to PBD is set-based design (SBD), which explores a large number of design alternatives grouped into sets and uses uncertainty resolution to select the most promising sets. SBD research dates back to the early 1990s [12,13].

Recent advances in SBD research have increased the adoption of SBD by several organizations. For example, in 2008 the then Commander of the Naval Sea Systems Command sent a memo entitled “Ship Design and Analysis Tool Goals”, which required the use of SBD and the use of new tools for trade-off analysis [14]. In 2018, the United States Air Force in their Capability Development Guidance mandated that “Development planners should use a set-based design modeling, simulation, and analysis approach, which supports the thorough exploration of alternative solutions while maintaining maximum design trade space”.

The purpose of this paper is to show how tradespace exploration in early system design using an integrated model-based engineering (MBE), SBD, and trade-off analysis

method can help develop requirements and identify high-performing design alternatives that have an affordable cost. We propose a qualitative SBD conceptual framework and quantitative SBD approach to inform decision-making by design teams. The quantitative SBD approach fills a gap in the literature on how to implement quantitative SBD. We use an unmanned aerial vehicle (UAV) case study with an integrated model-based engineering framework to demonstrate the benefits of SBD. Specifically, we demonstrate how SBD helps to (1) analyze requirements to inform requirement developers and (2) assess design decisions using design sets to better inform design teams when selecting design options.

We organize the rest of this paper in the following manner. Section 2, entitled Set-Based Design, provides insight into SBD as a design process and an overview of the recent SBD literature. Additionally, it contains the proposed quantitative SBD implementation methodology, which informs design teams. We then describe the UAV case study in Section 3 (Unmanned Aerial Vehicle Case Study). Section 4 (Insights from Quantitative Set-Based Design) uses this case study to describe and demonstrate how to use SBD to inform requirement analysis and select design options. Finally, Section 5 (Summary and Future Work) concludes with a summary and discussion of future research.

2. Set-Based Design

2.1. Point-Based versus Set-Based Design

The design of complex systems is challenging and very time-consuming. Traditionally, designers and engineers use point-based design (PBD) methods that decompose the system into subsystems or components and use subsystem design expertise to begin preliminary design. In this process, a team of designers concurrently use techniques from their field of expertise to perform analyses. Additionally, subsystems and systems engineers use modeling, simulation, and optimization tools to find good solutions throughout the PBD decomposition process. Optimizing PBD decomposition to solve engineering design problems is not a new research field [1–11]. These methods use system design variables or incorporate value to reflect stakeholder needs for decision making [15]. No matter the optimization method, the end result may be a few good solutions or several “optimal” solutions to investigate in further analyses. Of course, real-world complex systems often have a non-linear design space, which makes it difficult to find the true efficient frontier. These methods find “good” solutions but not necessarily solutions on the design space’s actual Pareto Frontier. This occurs throughout the system design process until the decision-maker selects a single PBD solution for further development.

Set-based design (SBD) builds upon the best practices of PBD. One key difference is that SBD considers a large number of alternatives grouped into sets and reduces the number of these sets by increasing the detail and analyses to determine feasibility. Wade et al. [16] illustrate this difference between PBD and SBD to provide a motivation to use SBD. The illustration, seen in Figure 1, shows how SBD has a greater potential to find sets of solutions on the Pareto Frontier based upon its use of a larger number of alternatives.

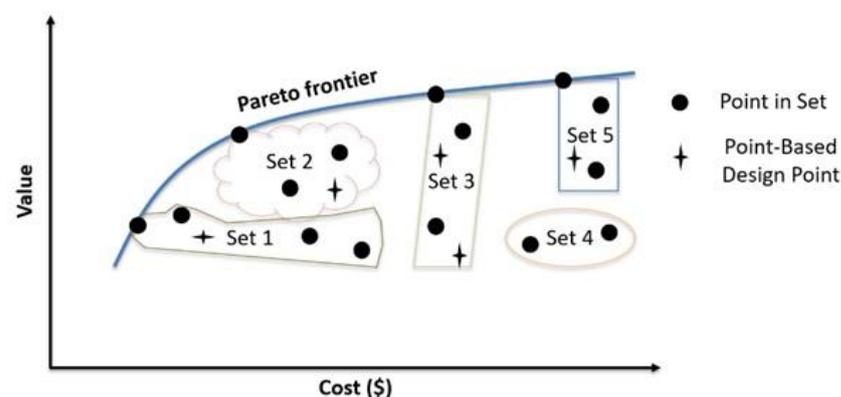


Figure 1. PBD (point-based design) and SBD (set-based design) Tradespace Comparison [16].

In 1993, Ward and Seering [13] published two SBD articles, which discussed a method to find the optimal design of a mechanical system by using sets of specifications. In 1995, Ward et al. [17] used the phrase “set-based concurrent engineering” to describe Toyota’s design approach, which included delaying decisions, communicating “ambiguously,” and producing a large number of alternatives. Ward et al. [17] found from analyzing Toyota’s processes that SBD increases communication, collaborators’ trust, and parallelism, while reducing the number of meetings and enabling an improved search for a “globally optimal design”. They further provided a five-step process to perform SBD: (1) define sets of alternatives at the system level; (2) define sets of alternatives at the subsystem level; (3) look at subsystems to find parallels to categorize the sets; (4) converge slowly to a single solution by using step 3 to determine subsystem specifications, and (5) maintain all decisions once they are made [17]. Later, Singer et al. [18] provided three SBD tenets: (1) “considers large number of designs”, (2) “allows specialist to consider a design from their own perspective and use the intersection between individual sets to optimize a design”, and (3) “establish feasibility before commitment”. The third tenet uses a slow set-narrowing process that includes increasing detail, a commitment to the selected set, and an uncertainty management process that uses process gates as elements to “establish feasibility”. These steps and characteristics are consistent with the work of other researchers [18–35]. However, Specking et al. [36] performed a SBD literature search that demonstrated a lack of quantitative techniques in the previous SBD literature. Additionally, they introduced the need to use decision analysis to capture value and the concepts of set drivers and set modifiers to add greater meaning to the design set definition. Set drivers “are the fundamental design decisions that define the platform characteristics that enable current and future missions”, while set modifiers “are design decisions that are ‘added on’ to the platform and can be modified to adapt to new missions and scenarios” [16].

A major gap in the literature is the lack of specific techniques to perform SBD in a team environment. Many papers discuss using teams to develop sets, but none of them describe a methodology for how the teams should collaborate to perform SBD. For example, Diaz [37] uses a team of stakeholders and morphological matrices to develop concept sets. This provides conceptual insight into a qualitative set creation method, but it does not provide techniques for those teams to analyze the sets to converge to set selection. One of the most common illustrations is by Ward et al. [17], which depicts Toyota’s parallel set-narrowing process. This diagram simply illustrates that all teams should work in parallel and converge to a single set.

2.2. Set-Based Design Conceptual Framework

Figure 2 provides a conceptual framework to illustrate the use of SBD system design and system analysis techniques throughout the system design lifecycle. The process starts with determining the system needs/requirements. These requirements drive all design stages. The framework flows from these requirements to the exploratory, concept, and development phases through time. Each phase uses design and analysis techniques that provide additional information about the system. The key is that design and analysis can occur concurrently. Subsystem design teams develop models that system analysts use to assess system performance and provide insights back to subsystem designers about the system feasibility and performance. These insights allow subsystem designers to improve their design concepts. Another key feature is the framework’s ability to update requirements throughout the design process. Additional information about feasibility, performance, and cost from models and simulations informs requirement refinement. This information is available due to the large set of designs being considered, which helps to explore the entire design space and show how each requirement affects the feasible design space. As design uncertainties are resolved by models, simulations, and prototypes, the initial numbers of sets slowly converge to a single point solution at the end of the process. The production phase uses this solution. This paper focuses on tradespace exploration with

SBD and trade-off analysis during early design stages (i.e., pre-milestone A or exploratory and concept life cycle stages).

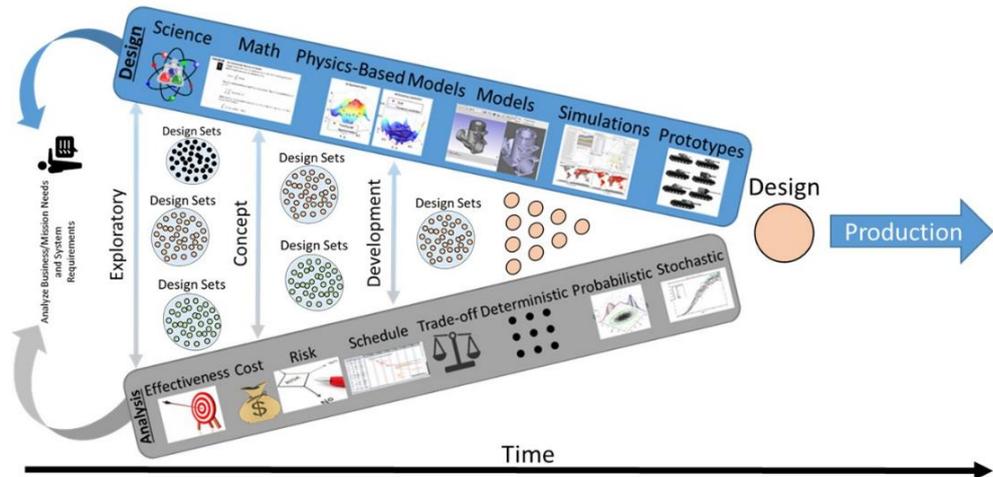


Figure 2. SBD Conceptual Framework for System Design.

2.3. Quantitative Set-Based Design with System Design Teams

Quantitative SBD implementation methodologies are an emerging area of research. Rapp et al. [38] provided an SBD scheme for product development that “minimizes the impacts, by proactively considering the possibility of changes in the external factors and the implication of mid-course design changes.” Specking et al. [39] introduced and demonstrated an SBD tradespace exploration implementation process for early system design, as shown in Figure 3. They also demonstrated that SBD requires the use of MBE and an integrated framework in an early design to adequately explore the tradespace.

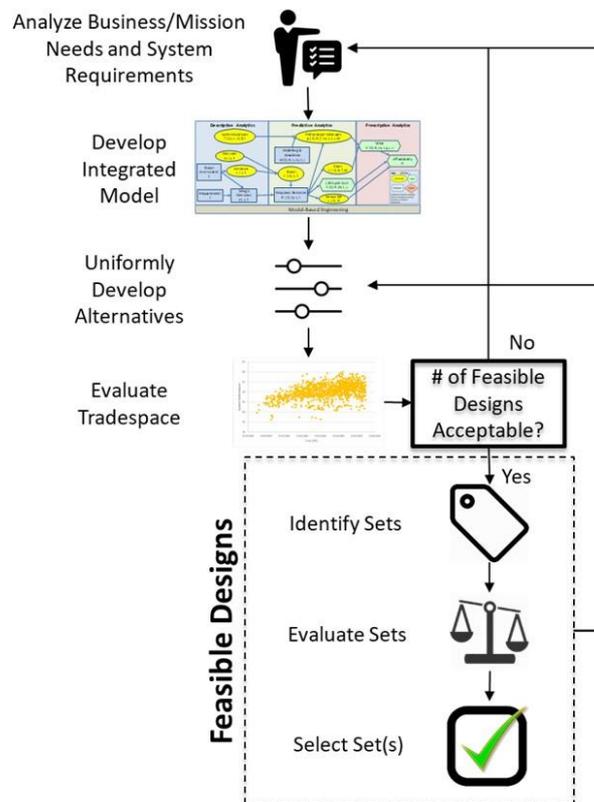


Figure 3. System Analysts SBD Tradespace Exploration Process (Modified from Ref. [39]).

Figure 3 starts by determining the mission needs and system requirements. It then requires the creation of an integrated model that connects these requirements to design decisions, and ultimately, to an affordability analysis. Instead of using optimization, the SBD process uses Monte Carlo simulation to uniformly develop the alternatives and evaluate the alternatives using desired response variables, such as system performance or cost, as illustrated in Figure 4 as an affordability analysis (system performance versus system cost). The integrated model assesses the feasibility of each alternative based on the requirements. If the number of feasible designs is unacceptable, the mission needs and system requirements are reevaluated, and the model is updated. The feasible points in the tradespace can be grouped into sets once the number of feasible designs is found to be acceptable by the decision authority. SBD's use of sets enables additional analyses, such as the one illustrated in Figure 5. These sets are evaluated to provide insights into the tradespace and system requirements. Analysts use dominance to eliminate sets. For example, Figure 5 demonstrates that engine type P dominates engine type E. Analysts then present the remaining sets of alternatives to the decision authority to carry forward to the next design phase.



Figure 4. SBD Feasible Tradespace Created by Monte Carlo Simulation.

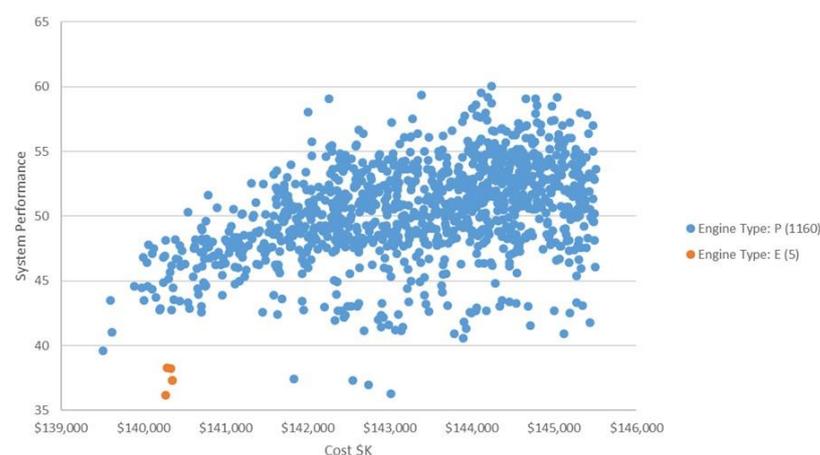


Figure 5. Tradespace Classified by Engine Type as Sets.

The process in Figure 3 provides an “analyst” perspective or the technical aspect of SBD, but it does not describe how to use that analysis to encourage interactions between subsystem design teams, i.e., the social aspect of SBD, to inform design decisions. The qualitative SBD literature uses a parallel approach, which requires interaction with subsystem design teams. This research uses the early design set-based design tradespace exploration

process found in Figure 3 by using quantitative analysis to provide design teams with the ability to assess the impact of their proposed subsystem design on the system.

Figure 6 depicts our proposed quantitative SBD implementation methodology with the roles and responsibilities described in Table 1. Many organizations might not have all of these roles or may have other names for them. The engineering manager or systems engineer may be expected to fill many of these roles. The implementation of this method requires technical and social skills. Implementers must be able to perform the required analysis, while managing the overall process and communicating with the right stakeholders at the right time in the most effective manner.

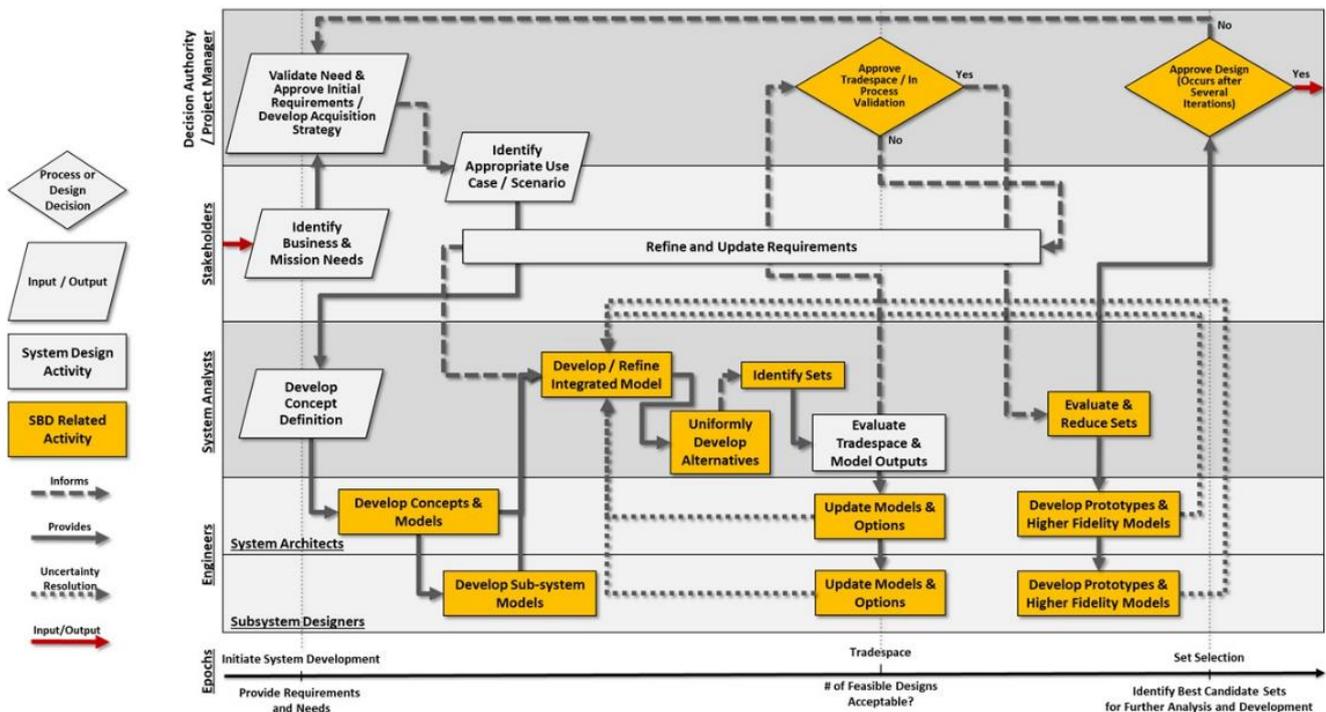


Figure 6. Quantitative Set-Based Design Process Swim Lane.

Table 1. Roles and Responsibilities of Team-Based Set-Based Design Process.

Role	Responsibilities
Decision Authority / Project Manager	Make decisions, answer for the successes or failures of the project, and communicate with stakeholders
Stakeholders	Provide insight into project needs and requirements
System Analysts	Work with system designers and engineers to gather relevant models to develop and analyze the integrated model to provide information to the decision authority
System Architects	Develop overall system architecture, concepts, and models
Subsystem Designers	Develop subsystem designs and models

As with any system design, the process begins with mission analysis. This opportunity or capability shortfall results in a need for a new system design. Following need validation and use case development, systems engineers define design concepts and request design models from the various system architects and subsystem design teams. Analysts then construct an integrated system model from these subsystem and system models, as well as other analytical models and simulations, which are consolidated in an MBE environment.

Systems analysts use this integrated model to uniformly generate alternatives based upon the desired decision variables in the order of hundreds of thousands of alternatives by using Monte Carlo simulation.

Based on the generated alternatives and the integrated models, the systems analysts identify and define the set-drivers which define design sets in the cost vs. value tradespace. As new information is provided, the designers and analysts refine the decision options and the models. All feasible sets should be retained in the model in case uncertainty resolution determines that the previous most promising sets are infeasible, which allows analysts to return to these sets. This process is continued until the system architects, subsystem designers and systems analysts agree with the tradespace, which is sent to the decision authorities to obtain approval prior to design selection. If the tradespace is not approved, the decision authority informs the stakeholders of issues, such as requirements issues, who then may update the requirements. System analysts then repeat the integrated model refinement through tradespace evaluation processes until the tradespace is resubmitted to the decision authority for approval. Once approved, system analysts evaluate and identify the most promising sets for uncertainty resolution. At this point, the system architects and subsystem designers provide higher fidelity models and develop prototypes to resolve uncertainty and inform set selection. Resolving uncertainty is key for any SBD process. These higher fidelity models and prototypes help increase the detail of the system as development continues. This aligns with Singer et al.'s [18] elements to "establish feasibility". Any changes require an update to the integrated model to repeat the integrated model refinement through set evaluation/selection processes. Ultimately, this process converges to a few points, which are presented to the decision authority in a tradespace. The decision authority selects a final design to move to production.

A key aspect of this process is the ability to efficiently revise and integrate updated design information from designers, results from models and simulations, and requirements from stakeholders, without incurring significant added delays or costs in the design process. Our process accomplishes this through the use of integrated MBE methodologies, maintaining numerous potential design alternatives, and delaying design decisions until uncertainty is adequately resolved. The result of this quantitative SBD process is the development of a Pareto optimal set of alternatives, presented to the decision authority for final selection and approval.

3. Unmanned Aerial Vehicle Case Study

3.1. Overview

We use the UAV case study developed by Small [40] to demonstrate our SBD techniques. This notional case study used an integrated trade-off analytics framework, shown in Figure 7. The case study uses seven design decisions (engine type, wingspan, operating altitude, electro-optical (EO) sensor width choice, EO field of view (FOV), infrared (IR) sensor width choice, and IR FOV) that propagate through eleven performance measures and several parametric physics models to calculate value and life-cycle cost.

The UAV case study uses value curves for each performance measure to scale from a minimum acceptable performance level to an ideal performance level. Parametric physics models score the alternative on its corresponding value curve for each alternative. The UAV model classifies an alternative as infeasible if it does not meet the required performance level for any performance measure. If an alternative meets the required performance level for all performance measures, the model automatically calculates an aggregated total value for the alternative by using an additive value model [41].

The UAV model uses the design decisions to automatically calculate each alternative's life-cycle cost in parallel with value. The seven design decisions are inputs to the calculation of hardware development costs, operation and maintenance costs, support costs, and indirect support costs.

The model presents all results in a cost versus value tradespace, such as the one seen in Figure 4. This helps decision makers converge upon high-performing design alternatives

with an affordable cost.

The UAV case study uses the method illustrated in Figure 3. Using Monte Carlo simulation to uniformly explore the tradespace, we create 100,000 unique alternatives by uniformly selecting an option for each design decision. For example, if you only consider two engine types, approximately 50% of the 100,000 alternatives will have one engine type, while the other 50% will have the other engine type. All alternatives propagate through the integrated model to determine their feasibility and associated value and cost.

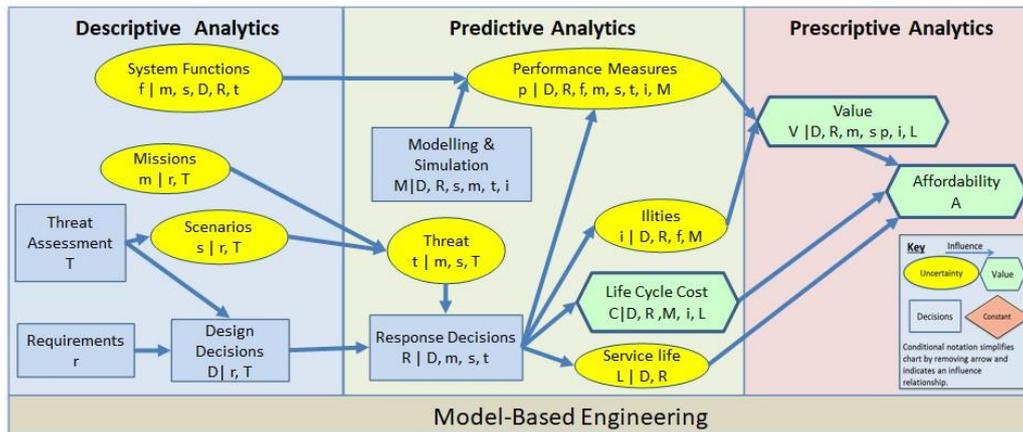


Figure 7. Integrated Trade-off Analysis Framework [40].

3.2. SBD Method Verification

Specking et al. [42] verified the UAV's tradespace exploration process. They hypothesized that optimization could be used to verify the SBD tradespace exploration process by determining if SBD found the model's Pareto Frontier. They used a genetic algorithm coded in a custom Excel-based visual basic macro to find Pareto points. A genetic algorithm was used since the UAV model contained a highly nonlinear (nonconvex) tradespace. The UAV SBD implementation found 189 design points that dominated the points found by the genetic algorithm.

4. Insights from Quantitative Set-Based Design

Singer et al.'s [18] three tenets of SBD, described in Section 2.1, are the features that separate SBD from PBD. These tenets, when used with our proposed implementation methodology, enable several benefits for system designers, systems engineers, the decision authority, and other stakeholders. These benefits include informing requirement analysis and assessing design decisions through sets and/or using a team of experts.

4.1. Informing Requirement Analysis

Requirement analysis is an important part of systems engineering. Sometimes, the requirements result in no feasible alternatives. This is why it is important for analysts to have an understanding of how requirements affect the tradespace. SBD with an integrated model using MBE techniques enables analysts to assess the impact of a requirement's change on the feasible tradespace in near real-time. Parnell et al. [43] provide an example. This section expands upon that example.

Figure 8 shows these feasible alternatives in yellow and provides a starting point to help analyze requirements to inform requirement developers. The alternatives in orange display all of the explored 100,000 designs. The blue points represent the alternatives that are feasible based solely upon the UAV case study's physics-based parametric models (27,750). When requirements are added, the UAV model has only 2526 feasible design alternatives. The number of feasible solutions increased to 4366 (black) from the UAV model's original number of feasible solutions when the requirements were relaxed. Constraining the requirements reduced the feasible space to 924 designs (red). Table 2 contains

constrained, original, and relaxed requirements for the UAV case study.

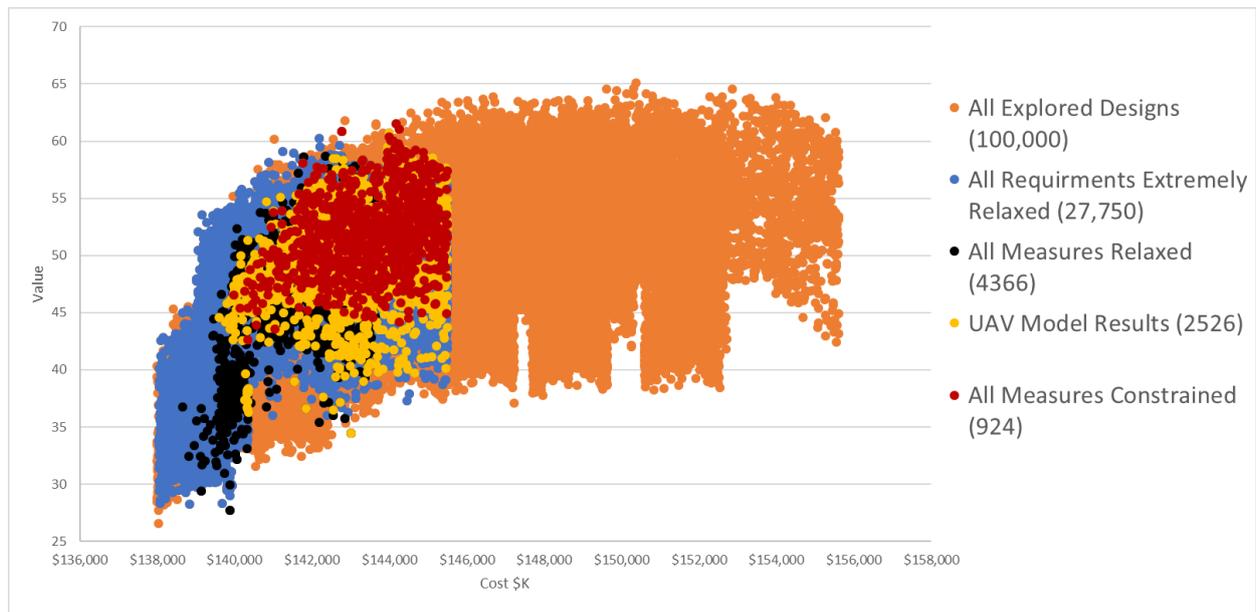


Figure 8. Effects of Requirements on the UAV’s Feasible Tradespace [43].

Table 2. UAV Case Study with Relaxed, Original, and Constrained Requirements.

Performance Measure	Constrained	UAV Case Study	Relaxed
UAS weight (lbs)	40	50	60
Time required to fly 10 km (minutes)	10	15	20
Time required to scan a 5 km × 5 km box during the day (minutes)	180	200	220
Time required to scan a 5 km × 5 km box during the night (minutes)	180	200	220
Dwell time (minutes)	90	60	30
Perceived area of UAV at operating altitude (ft ²)	14	16	18
Difference between operating altitude and attack helicopter operating altitude of 1000 m	250	0	0
Detect human activity in daylight	0.7	0.6	0.5
Detect vehicular activity in daylight	0.7	0.6	0.5
Detect human activity at night	0.7	0.6	0.5
Detect vehicular activity at night	0.7	0.6	0.5

We also examine the number of feasible designs produced when each requirement is changed one by one. For this analysis, we only change the requirement of interest, while maintaining all other requirements at their original UAV case study values. Using an integrated model with MBE makes this an easy analysis to perform. The model updates in near-real time when we change each requirement. We then can record the number of feasible designs. Figure 9 shows the results of a one-by-one analysis for the UAV case study that used the constrained, original, and relaxed values in Table 2. We display the results by using a tornado graph, which places the requirements with the greatest change in feasible designs at the top and the requirements with the fewest changes or no change in feasible design at the bottom. Figure 9 shows that the “Detect Human Activity at Night” and “Detect Human Activity in Daylight” requirements have the greatest effect on the feasible tradespace, while “Detect Vehicular Activity at Night”, “Detect Vehicular Activity

in Daylight”, and “UAS weight” did not affect the feasible tradespace at all, given the relaxed, original, and constrained values.

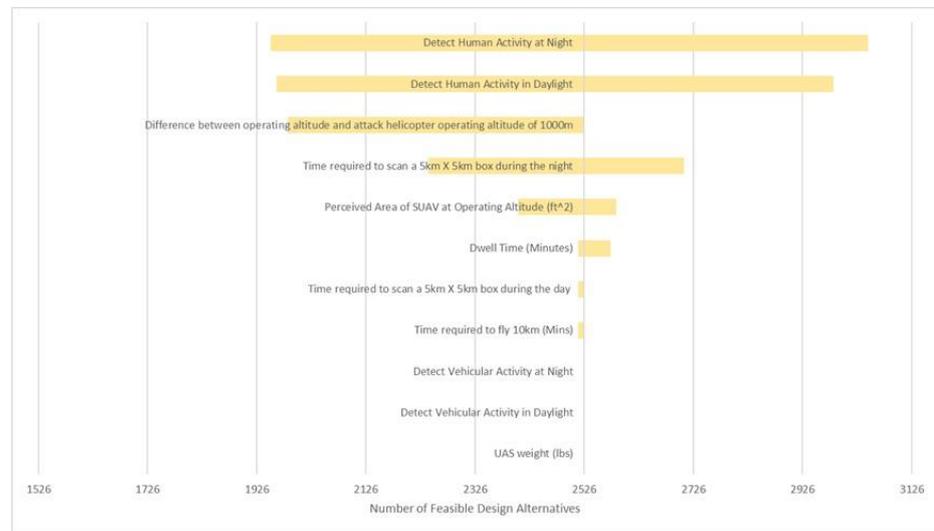


Figure 9. UAV Case Study Results of One-By-One Requirement Analysis [43].

It is important to note that analysts should not focus only on the number of feasible designs. Instead, they should evaluate the model’s entire output. Knowing the change in the number of feasible designs provides useful information, but we find additional insights by analyzing the entire set of results. These results can provide insights into the quality of the feasible designs produced by the requirement changes. We may find that the requirement changes produce Pareto points not found in the original model results. For example, Figure 10 shows the impact on the feasible tradespace when we analyze the results for the most sensitive requirement, “Detect Human Activity at Night”. It is evident from this figure that this requirement affects the number and quality of feasible designs. This is seen by the additional relaxed (black) and constrained (red) points on the Pareto Frontier when compared to the original UAV results (yellow). In fact, the Pareto Frontier is almost created by the new feasible designs.



Figure 10. Effect on Feasible Tradespace by Changing Most Sensitive UAV Requirement.

4.2. Assessing Design Decisions Using Sets

Using a refined UAV model with more realistic requirements, we investigate the insights of using sets. This model produced 1165 feasible designs. One of the distinguishing features of SBD is the grouping of design alternatives into sets of alternatives to provide insights to designers.

Figures 11 and 12 illustrate four examples of using decision variables to classify design decisions as set drivers for the refined UAV case study. Figure 11A uses engine type as the set driver. This demonstrates that some decision options are better than others. The piston engine type, P, produces more feasible alternatives when compared to the electric engine type, E, but more importantly the piston engine’s alternatives dominate the electric engine’s alternatives by providing greater value at an equal or lower cost. Less than 1% (5 out of 50,000) of the electric engine alternatives created by the Monte Carlo simulation are feasible. It is clear from this analysis that the selected design should use the piston engine or consider adding a different or improved alternative engine.

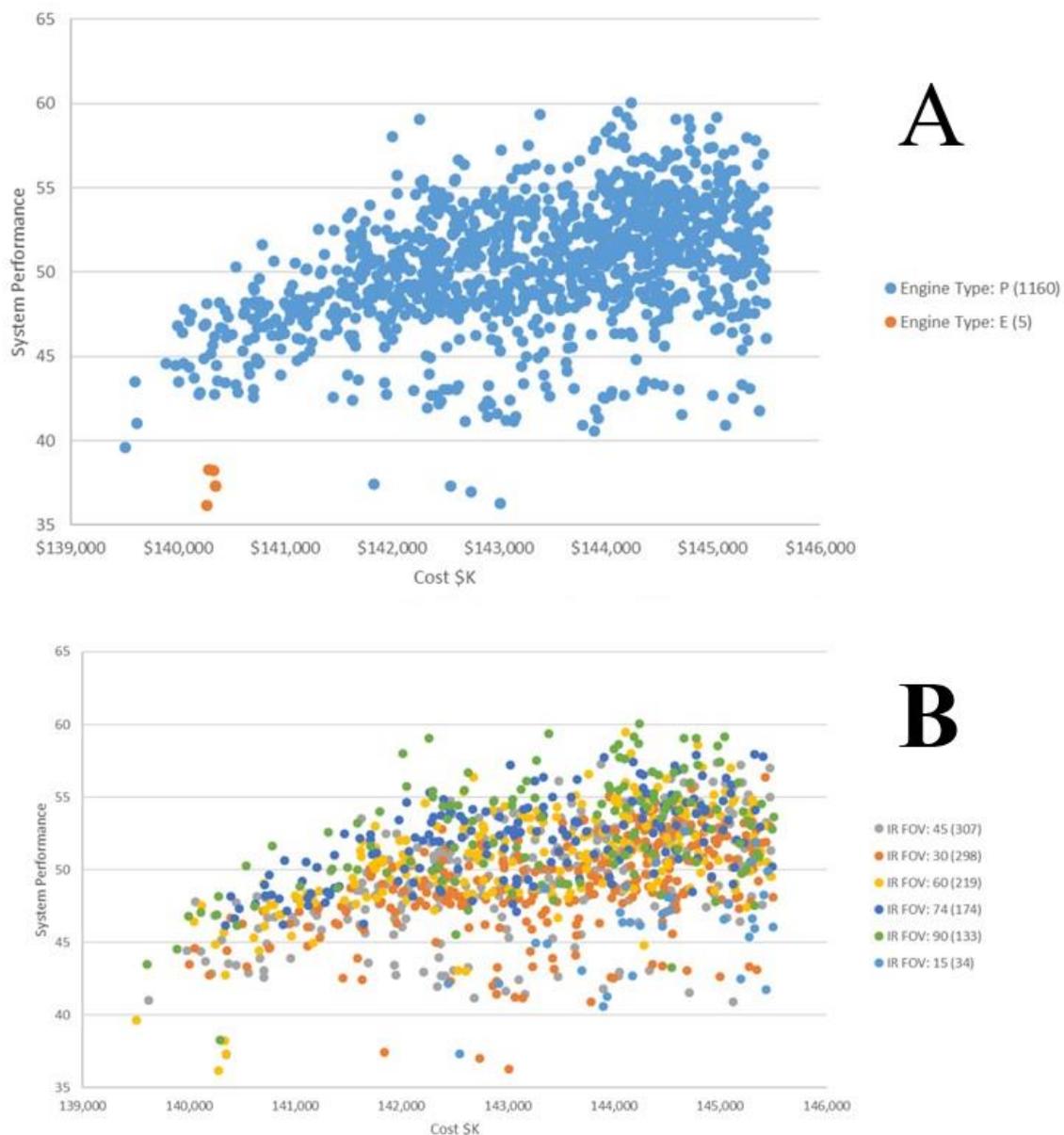


Figure 11. Tradespace Classified by (A) Engine Type (top) and (B) IR Sensor Field of View (Bottom).

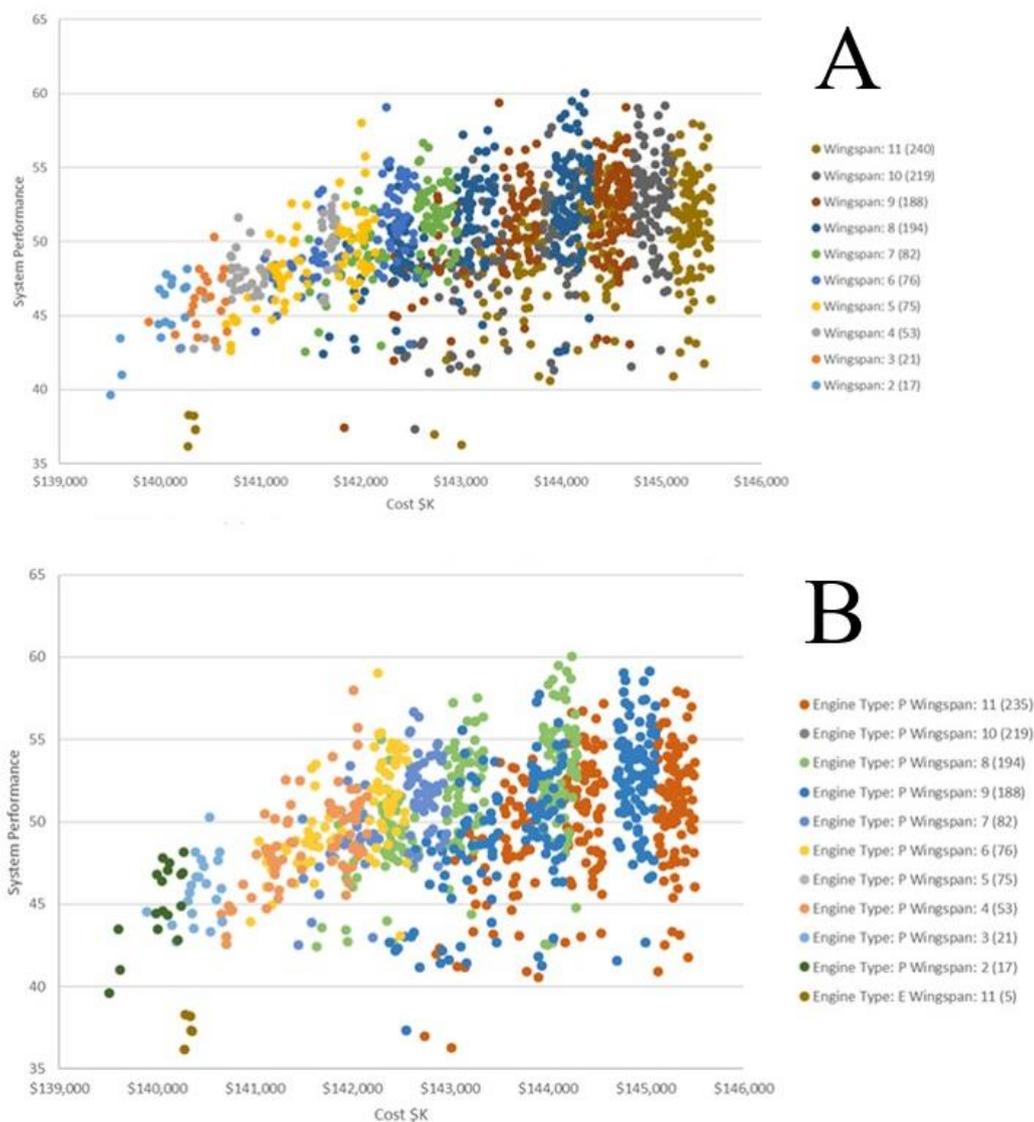


Figure 12. Tradespace Classified by (A) Wingspan (top) and (B) Engine Type and Wingspan (Bottom).

Figure 11B classifies the feasible tradespace with the IR FOV as the set driver. This provides a significant contrast to Figure 11A, since all of the sets overlap. It is evident that IR FOV 15 is dominated by other options, which means that it could be removed as an option, but the remaining options' values and cost ranges span the tradespace. This means that IR FOV is probably a set modifier.

Figure 12A classifies the feasible tradespace by wingspan. In this illustration, the labels for all design options for wingspan are grouped by rounding down to the nearest whole number measured in feet. The illustration is interesting since the sets are column-like groupings of points. These sets have some overlap. This means that wingspan may be a set driver or a set modifier.

Figure 12B classifies the feasible tradespace using wingspan and engine type as set drivers. When reviewing the tradespace by wingspan, it is not easy to determine which sets to focus our analyses on. However, if we classify the tradespace by wingspan and engine type, we see that we can disregard all of the feasible electric engines that use a wingspan within the 11 feet range. The records of all feasible sets should be retained in case uncertainty resolution changes the assessment of the most promising sets. This allows designers to consider focusing and performing additional analysis on the most promising sets for uncertainty resolution.

4.3. Providing Design Decisions Insights to Design Teams

SBD requires greater initial analytical effort than traditional PBD methods. The added effort is justified by the potential to more fully explore the tradespace and develop better system designs. Performing quantitative SBD provides the opportunity to better inform system and sub-system design teams and help them recommend better designs.

To illustrate this, consider the SBD implementation method found in Figure 6, and the refined UAV model that produced 1165 feasible designs from 100,000 design alternatives with an illustrative team. After developing an integrated model and uniformly developing alternatives, system analysts can consider each individual design decision. They then evaluate the tradespace and model output in order to present the results to the relevant system architects, system engineers, and subsystem designers.

Figure 13 shows an example tradespace for the refined UAV case study, which uses the EO sensor width choice as the set driver. The analyst presents this tradespace to the corresponding concept architects and EO sensor designers. From the tradespace review, the engineers decide to remove the EO sensor width choices 7, 8, and 9 since these EO sensors contain zero feasible designs. Additionally, they decide to remove EO width choice 6 since its 13 feasible designs provide lower values at a higher cost compared with other EO sensor width choices.

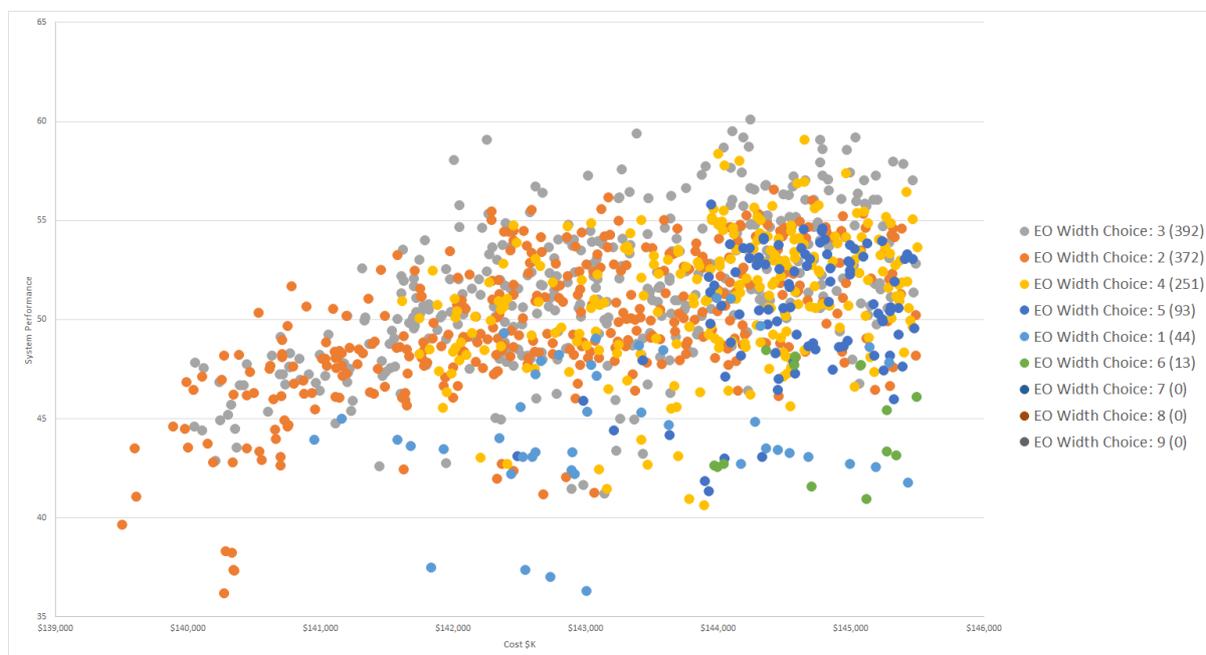


Figure 13. Original UAV Tradespace with Electro-Optical Sensor Width Choice as Set Driver.

The system analyst performs this analysis with each design decision and has a similar conversation with the appropriate engineering teams. The analyst uses dominance and infeasibility to provide insights to these teams. The outcomes of these conversations remove the electric engine type, operating altitudes 600 to 1000, and infrared (IR) sensor width choices 6, 7, 8, and 9, in addition to the EO sensor width options.

The system analyst then updates the integrated model by changing the decision options appropriately and uniformly redeveloping the alternatives. For example, this process removes the 50,000 designs that use the electric engine and then creates 100,000 designs that use the piston engine. The result of this process produces a new revised tradespace, seen in Figure 14. System architects, systems engineers, and subsystem designers have similar conversations as before. This time the EO sensor design team removes EO sensor width choices 1 and 5 since EO sensor width choices 2, 3, and 4 provide equal to greater value at an equal or lower cost. In addition to these EO sensor width options, the other

engineering teams remove IR sensor width choices 1 and 5, EO sensor field of view 15, and IR sensor field of view 15.

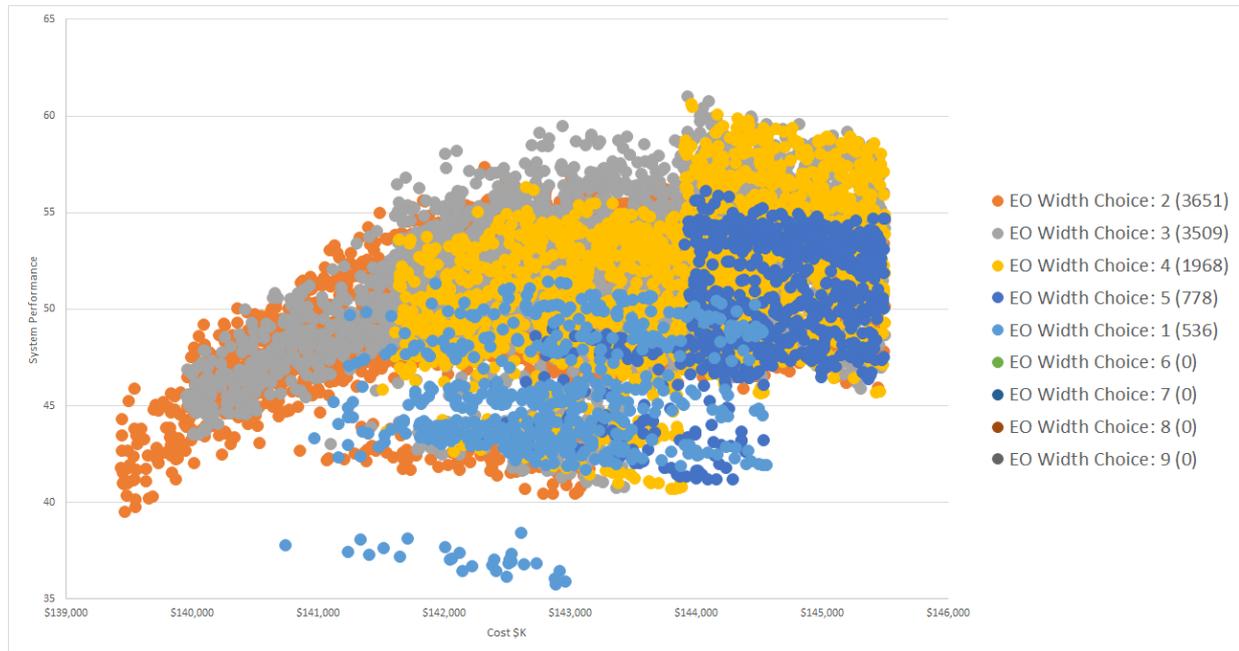


Figure 14. First Revised UAV Tradespace with Electro-Optical Sensor Width Choice as Set Driver.

The system analyst repeats this process of discussing, removing, and recreating alternatives until all teams agree with the final tradespace to propose to the decision authority. Project managers and system analysts will need to balance this uncertainty resolution technique with time.

We found the example UAV final tradespace after repeating this process three times, shown in Figure 15 and summarized in Table 3. Table 4 summarizes the overall impact on the tradespace by this SBD refinement process. This method of tradespace refinement increased the number of feasible designs by 3626% (1165 to 43,414). This is due to the removal of infeasible or dominated decision options, which allowed the exploration of other combinations of feasible decision options. This creates denser sets, which provides higher confidence for those decision options due to the increase in information.

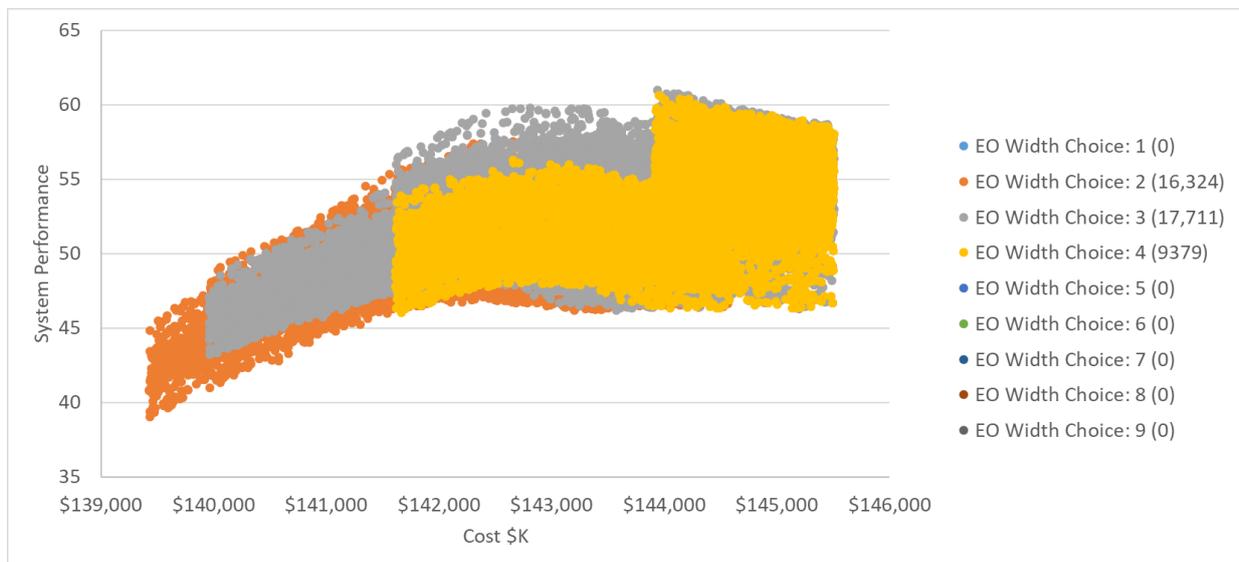


Figure 15. Final UAV Tradespace with Electro-Optical Sensor Width Choice as Set Driver.

Table 3. Summary of Tradespace Refinement Results for All Design Teams.

Design Decision	Original	1st Revised	2nd Revised	Final
Wingspan	2–12	2–12	2–12	2–12
Engine Type	E, P	P	P	P
Operating Altitude	300–1000	300–599	300–599	300–599
EO Sensor Width Choice	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5	2, 3, 4	2, 3, 4
IR Sensor Width Choice	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5	2, 3, 4	2, 3, 4
EO Sensor FOV	15, 30, 45, 60, 75, 90	15, 30, 45, 60, 75, 90	30, 45, 60, 75, 90	45, 60, 75, 90
IR Sensor FOV	15, 30, 45, 60, 75, 90	15, 30, 45, 60, 75, 90	30, 45, 60, 75, 90	45, 60, 75, 90

Table 4. Impact of Tradespace Refinement.

Tradespace	# of Considered Alternatives	# of Feasible Alternatives	# of Pareto Points	% Feasible (of Sampled)
Original	100,000	1165	12	1.2%
1st Revised	100,000	10,442	19	10.4%
2nd Revised	100,000	32,799	19	33%
Final	100,000	43,414	18	43%

An additional interesting analysis involves overlapping the four tradespaces, seen in Figure 16. This shows that all of the UAV tradespaces produced a similar range in terms of cost and value. The most interesting finding is that the tradespace refinement process did not dramatically affect the Pareto Frontier. It is not safe to say that this demonstrates that the 1st refinement with 10,442 feasible points is enough. The conversations with the design teams should lead to more information and higher-fidelity models, which help with uncertainty resolution. This helps increase the likelihood that the chosen sets of designs for the next design phase remain feasible.

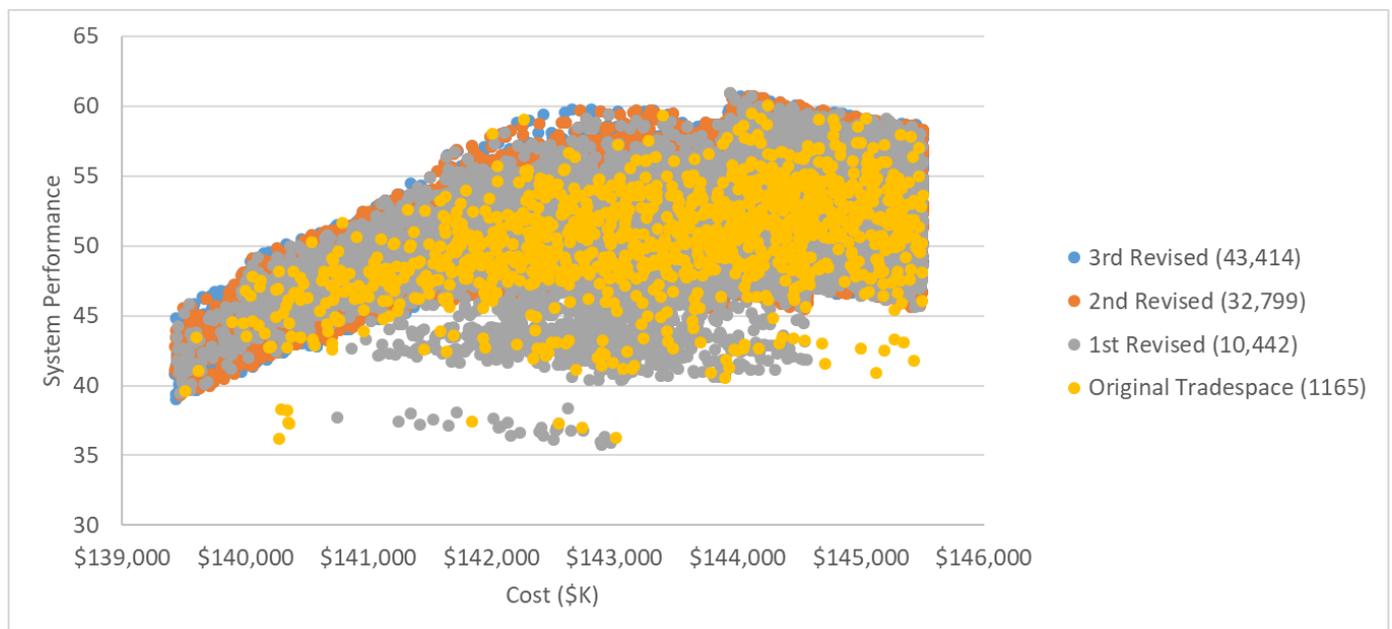


Figure 16. Tradespace Evaluation Comparison.

5. Summary and Future Work

Subsystem designers, system designers, systems engineers, systems analysts, and engineering managers seek high-performing design alternatives that have an affordable cost. Traditional PBD methods use system decomposition and optimization to find these design alternatives in the tradespace. Non-linear (i.e., non-convex) tradespaces require the use of heuristic optimization techniques, which find “good” design alternatives but are

not guaranteed to be globally optimum. SBD is an alternative to PBD and a concurrent engineering design technique that compares a large number of design alternatives. SBD classifies design alternatives into sets with common design decisions. Analysts explore the tradespace by evaluating the various sets to identify the most promising sets for uncertainty resolution or selection. Once one selects a set, all design alternatives in that set are available in the next design phase. Our proposed SBD conceptual framework for system design illustrates this process.

This paper provides a quantitative SBD approach for early system design and analysis to demonstrate how tradespace exploration and trade-off analysis helps analysts identify promising sets for uncertainty resolution, and eventually converge to high-performing design alternatives that have an affordable cost. We show how quantitative SBD can inform subsystem engineers, systems engineers, systems analysts, stakeholders, engineering managers, and decision authorities.

We use an Unmanned Aerial Vehicle (UAV) case study with an integrated MBE framework to illustrate how SBD helps to (1) analyze requirements to inform requirement developers and (2) assess design decisions through the use of design sets to inform design teams. SBD informs requirement analysis by analyzing each requirement's effect on the feasible tradespace by relaxing or constraining requirements. Additionally, SBD helps designers and analysts assess design decisions by providing an understanding of how each design decision affects the feasible tradespace. We conclude that SBD provides a more holistic tradespace exploration process since it provides an integrated examination of system requirements and design decisions. The exploration of a larger set of alternatives enables a process to explore the design space more completely than traditional methods. This integrated examination helps system analysts find high-performing design alternatives that have an affordable cost to present to the decision authority.

SBD research has advanced greatly over the last several years, but SBD still provides a great opportunity for additional research. SBD requires designers and engineers to provide models and simulations instead of designs during early system design. This requires a culture change to move from PBD to SBD. The models and simulations needed to implement SBD during early design require access to data and other information. The various levels of information lead to models with various levels of fidelity to resolve uncertainty. These levels of fidelity affect the runtime of the SBD analyses and could require an increase in computational power. More research is needed to expand our SBD implementation process to better understand the effects of various levels of model fidelity on the execution of the quantitative SBD process.

Author Contributions: Conceptualization, E.S.; formal analysis, E.S.; funding acquisition, G.S.P. and E.P.; investigation, E.S.; methodology, E.S. and N.S.; project administration, G.S.P. and E.P.; writing—original draft, E.S.; writing—review and editing, E.S., N.S., G.S.P. and E.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the United States Army Engineer Research and Development Center (ERDC) as part of an Engineering Resilient Systems (ERS) research project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Excel-based UAV model available upon request.

Acknowledgments: We appreciate Colin Small allowing us to use the UAV case study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lasdon, L.S. *Optimization Theory for Large Systems*; Courier Corporation: North Chelmsford, MA, USA, 1970.
2. Wismer, D.A. *Optimization Methods for Large-Scale Systems... with Applications*; McGraw-Hill Companies: New York, NY, USA, 1971.
3. Sobieszczanski-Sobieski, J.; Barthelemy, J.-F.M.; Giles, G.L. *Aerospace Engineering Design by Systematic Decomposition and Multilevel Optimization*; National Aeronautics and Space Administration, Langley Research Center: Hampton, VA, USA, 1984.

4. Azarm, S.; Li, W.-C. Multi-level design optimization using global monotonicity analysis. *J. Mech. Transm. Autom. Des.* **1989**, *111*, 259–263. [[CrossRef](#)]
5. Haimes, Y.Y. *Hierarchical Multiobjective Analysis of Large-Scale Systems*; Hemisphere Pub.: New York, NY, USA, 1990.
6. Sobieszcanski-Sobieski, J. Sensitivity analysis and multidisciplinary optimization for aircraft design: Recent advances and results. In Proceedings of the 16th Congress International Council of the Aeronautical Sciences (ICAS), Jerusalem, Israel, 28 August–2 September 1988.
7. Sheridan, D.; Clark, D.; Jones, R.; Fein, J. The ASSET Program- A Current Navy Initiative. In Proceedings of the SNAME Spring Meeting, Los Angeles, CA, USA, 15 December 1984.
8. Cramer, E.J.; Frank, P.D.; Shubin, G.R.; Dennis, J.; Lewis, R. On alternative problem formulations for multidisciplinary design optimization. In Proceedings of the 4th Annual AIAA/Air Force/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH, USA, 21–23 September 1992.
9. Davis, W. A generalized decomposition procedure and its application to engineering design. *J. Mech. Des.* **1978**, *100*, 739–746. [[CrossRef](#)]
10. Johnson, R.; Benson, R. A basic two-stage decomposition strategy for design optimization. *J. Mech. Transm. Autom. Des.* **1984**, *106*, 380–386. [[CrossRef](#)]
11. Johnson, R.; Benson, R. A multistage decomposition strategy for design optimization. *J. Mech. Transm. Autom. Des.* **1984**, *106*, 387–393. [[CrossRef](#)]
12. Ward, A.C.; Seering, W.P. The Performance of a Mechanical Design Compiler. *Trans. Am. Soc. Mech. Eng. J. Mech. Des.* **1993**, *115*, 341–345. [[CrossRef](#)]
13. Ward, A.C.; Seering, W.P. Quantitative inference in a mechanical design compiler. *Trans. Am. Soc. Mech. Eng. J. Mech. Des.* **1993**, *115*, 29–35. [[CrossRef](#)]
14. Doerry, N. A Vision for Ship Design and Analysis Tools. In *Marine Technology*; SNAME: Alexandria, VA, USA, 2012.
15. Hootman, J.C.; Whitcomb, C. A military effectiveness analysis and decision making framework for naval ship design and acquisition. *Naval Eng. J.* **2005**, *117*, 43–61. [[CrossRef](#)]
16. Wade, Z.; Parnell, G.; Goerger, S.; Pohl, E.; Specking, E. Designing Engineered Resilient Systems Using Set-Based Design. In Proceedings of the 16th Annual Conference on Systems Engineering Research, Charlottesville, VA, USA, 8–9 May 2018.
17. Ward, A.; Liker, J.K.; Cristiano, J.J.; Sobek, D.K. The second Toyota paradox: How delaying decisions can make better cars faster. *Sloan Manag. Rev.* **1995**, *36*, 43.
18. Singer, D.J.; Doerry, N.; Buckley, M.E. What Is Set-Based Design? *Naval Eng. J.* **2009**, *121*, 31–43. [[CrossRef](#)]
19. Burrow, J.; Doerry, N.; Earnesty, M.; Was, J.; Myers, J.; Banko, J.; McConnell, J.; Pepper, J.; Tafolla, T. *Concept Exploration of the Amphibious Combat Vehicle*; SNAME Maritime Convention: Houston, TX, USA, 2011.
20. Finch, W.W.; Ward, A.C. A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty. In Proceedings of the 1997 ASME Design Engineering Technical Conferences, Sacramento, CA, USA, 14–17 September 1997. Paper No. DETC97/DTM-3886.
21. Ford, D.N.; Sobek, D.K. Adapting real options to new product development by modeling the second Toyota paradox. *IEEE Trans. Eng. Manag.* **2005**, *52*, 175–185. [[CrossRef](#)]
22. Ghosh, S.; Seering, W. Set-Based Thinking in the Engineering Design Community and Beyond. In Proceedings of the ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, New York, NY, USA, 17–20 August 2014; p. V007T07A040.
23. Kim, W. A Framework for Set-Based Manufacturing Analysis and Visual Feedback. Ph.D. Thesis, The Pennsylvania State University, Centre County, PA, USA, 2015.
24. Liker, J.K.; Sobek, D.K.; Ward, A.C.; Cristiano, J.J. Involving suppliers in product development in the United States and Japan: Evidence for set-based concurrent engineering. *IEEE Trans. Eng. Manag.* **1996**, *43*, 165–178. [[CrossRef](#)]
25. Madhavan, K.; Shahan, D.; Seepersad, C.C.; Hlavinka, D.A.; Benson, W. An industrial trial of a set-based approach to collaborative design. In Proceedings of the ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, New York, NY, USA, 3–6 August 2008; pp. 737–747.
26. Malak, R.J.; Aughenbaugh, J.M.; Paredis, C.J. Multi-attribute utility analysis in set-based conceptual design. *Comput. Aided Des.* **2008**, *41*, 214–227. [[CrossRef](#)]
27. McKenney, T.A.; Kemink, L.F.; Singer, D.J. Adapting to Changes in Design Requirements Using Set-Based Design. *Naval Eng. J.* **2011**, *123*, 66–77. [[CrossRef](#)]
28. McKenney, T.A.; Singer, D.J. Determining the influence of variables for functional design groups in the set-based design process. In Proceedings of the ASNE Day 2012: Proceedings of the American Society of Naval Engineers Day 2012, Alexandria, VA, USA, 10 February 2012.
29. Mebane, W.L.; Carlson, C.M.; Dowd, C.; Singer, D.J.; Buckley, M.E. Set-Based Design and the Ship to Shore Connector. *Naval Eng. J.* **2011**, *123*, 79–92. [[CrossRef](#)]
30. Nahm, Y.-E.; Ishikawa, H. A new 3D-CAD system for set-based parametric design. *Int. J. Adv. Manuf. Technol.* **2006**, *29*, 137–150.
31. Naval Sea Systems Command. *Ship Design Manager (SDM) and Systems Integration Manager (SIM) Manual*; Naval Sea Systems Command: Washington, DC, USA, 2012.
32. Panchal, J.H. A Framework for Simulation-Based Integrated Design of Multiscale Products and Design Processes. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2005.

33. Raudberget, D. The decision process in Set-based Concurrent Engineering-An industrial case study. In Proceedings of the DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia, 17–20 May 2010.
34. Sobek, D.K.; Ward, A.C.; Liker, J.K. Toyota's principles of set-based concurrent engineering. *Sloan Manag. Rev.* **1999**, *40*, 67.
35. Ward, A.; Durward Sobek, I.; John, J.C.; Jeffrey, K.L. Toyota, concurrent engineering, and set-based design. In *Engineered in Japan: Japanese Technology-Management Practices*; Oxford University Press: Oxford, UK, 1995; pp. 192–216.
36. Specking, E.; Whitcomb, C.; Parnell, G.; Goerger, S.; Pohl, E.; Kundeti, N. Literature Review: Exploring the Role of Set-Based Design in Trade-off Analytics. *Naval Eng. J.* **2018**, *130*, 51–62.
37. Diaz Dominguez, D. Enhancing the conceptual design process of automotive exterior systems. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.
38. Rapp, S.; Chinnam, R.; Doerry, N.; Murat, A.; Witus, G. Product development resilience through set-based design. *Syst. Eng.* **2018**, *21*, 490–500. [[CrossRef](#)]
39. Specking, E.; Parnell, G.; Pohl, E.; Buchanan, R. Early Design Space Exploration with Model-Based System Engineering and Set-Based Design. *Systems* **2018**, *6*, 45. [[CrossRef](#)]
40. Small, C. Demonstrating Set-Based Design Techniques—A UAV Case Study. Master's Thesis, University of Arkansas, Fayetteville, AR, USA, 2018. Available online: <http://scholarworks.uark.edu/etd/2699> (accessed on 6 June 2019).
41. Parnell, G.S.; Bresnic, T.A.; Tani, S.N.; Johnson, E.R. *Handbook of Decision Analysis*; Wiley: Hoboken, NJ, USA, 2013.
42. Specking, E.; Parnell, G.; Pohl, E.; Buchanan, R. Evaluating the Tradespace Exploration Process of an Early System Design. In Proceedings of the 17th Annual Conference on Systems Engineering Research, Washington, DC, USA, 3–4 April 2019.
43. Parnell, G.S.; Specking, E.; Goerger, S.; Cilli, M.; Pohl, E. Using Set-Based Design to Inform System Requirements and Evaluate Design Decisions. In Proceedings of the 29th Annual INCOSE International Symposium, Orlando, FL, USA, 20–25 July 2019.