An Iterative Subsystem-Generated Approach to Populating a Satellite Constellation Tradespace

Andrew A. Rader* and Franz T. Newland†
COM DEV International, Cambridge, ON, N1R7H6, Canada

and

Adam M. Ross‡
SEAri, Massachusetts Institute of Technology, Cambridge, MA, 02139

Tradespace exploration provides a structured, concept neutral means of considering a large number of design alternatives while avoiding a premature focus on point designs. However, the process can be difficult to implement in the design of complex space systems when considering practical partitioning and allocation of design tasks. Complicating the effort further is the existence of many subsystem interactions that must be considered in order to assemble a system level tradespace. This paper presents a process used to populate and analyze the tradespace for an example constellation of Earth observation satellites, by analyzing individual subsystem level trades in parallel and capturing system level interactions at key nodal checkpoints. The process is shown to allow for practical development of tradespace knowledge, leveraging subsystem-level design expertise, as well as identifying novel design alternatives at the intersection of subsystem proposed alternatives.

I. Introduction

THE objective of tradespace exploration is to examine the tradeoffs and impacts of design choices on cost and performance throughout the lifecycle of a project. The result is an improved understanding of the relative impacts of decisions on expected utility (a measure of perceived benefit to a stakeholder) and cost. Not only does such an approach identify design options that are high in value, but also it provides a systematic approach for comparing options based on their ability to satisfy stakeholder preferences. This in turn leads to a re-examination of stakeholder requirements with respect to the schedule, risk, and financial cost impacts they contribute. For example, option A might achieve a requirement 90% of the time at a reasonable cost, while option B achieves it 100% of the time but at more than double the cost. The choice between options A and B can only be made based on stakeholder preferences – what value is placed on each of the requirements?

Although the tradespace contains many uncertainties and assumptions during early stage design, tradespace exploration is a process that provides insight by illustrating the value impacts of design choices and revealing interesting design points for more detailed analysis. Furthermore, the nature of the uncertainties can be quantitatively estimated using Monte Carlo Simulation or other methods. Moreover, tradespace exploration is a process rather than a point analysis, where the inherent assumptions are constantly refined as the problem becomes better understood.

* Spacecraft Systems Engineer, Mission Design Group, 60 Struck Crt., and AIAA Member.
† Manager, Mission Engineering, Mission Design Group, 60 Struck Crt., and AIAA Senior Member.
‡ Research Scientist, Engineering Systems Division, E38-574, and AIAA Senior Member

Populating a tradespace for a complex space system involves quantifying the performance of competing design alternatives, each having design trade-offs across multiple subsystems. This process is complicated by the fact that the subsystems are interactive: it is not straightforward to span the tradespace by simply selecting design variables at only the system or subsystem level. Prior published satellite system tradespace exploration case studies[1] have dealt with this complexity through recursive computer-based models as well as human-in-the-loop concurrent engineering design sessions; however these approaches may be unable to take advantage of domain expert knowledge, which can provide targeted improvements to the fidelity of the analysis. This paper presents an iterative approach to populating the tradespace for a satellite constellation performing an Earth observation mission, where subsystem level optimizations were performed in parallel by domain experts using their own tools, with interactions between subsystems at the system level assessed and propagated at key nodal checkpoints. This bottom-up approach allows for an independent partitioning of the design space by expertise (e.g., orbit, ground, launch, payload, etc.) in order to assemble a system level tradespace that captures individual subsystem level trades.

An Earth observation satellite constellation mission is a particularly illustrative example because the mission involves tradeoffs in performance and cost that are difficult to quantify individually. For an observation mission with global coverage and requirements for high observation times, frequent revisits, and fast downlink, all with high reliability for multiple points on Earth, it is not clear how these attributes should be traded off versus each other or versus costs. Since these attributes are time varying a number of important questions arose regarding how to value attribute tradeoffs. These questions include: how should peak performance be valued versus mean, median, or worst case performance? How should global coverage be valued versus coverage over a particular area of interest? How much value is derived from additional ground stations as compared with additional satellites? In the tradespace exploration, multiple architectures are considered, including nanosatellites with low duty cycles and strict power limitations, microsatellites, and hosted payloads on larger satellites. Preliminary studies also examined non-satellite options such as airplanes and UAVs.

Tradespace exploration provides a mathematical construct by which the subjective utility of competing design options can be assessed on a common basis (e.g., stakeholder utility vs. lifecycle cost). Several features are highlighted by the example tradespace shown in Figure 1:

- Increasing *value* is comprised of both increased utility (increasing along the y-axis) and lower cost (decreasing along the x-axis).
- The Pareto Front represents designs that have the highest utility *for a given cost*, or the lowest cost for a given utility.
- Designs falling closer to the Pareto Front are higher in value. Designs falling father below the Pareto Front are dominated by higher value designs, yielding less utility for the same cost.
- If the objective is simply to meet a requirement (and there is no desire to exceed it), a location near the intersection of the Pareto Front and the user requirement level yields the most efficient implementation.
- Since multi-attribute utility (MAU) contains contributions from all attributes, there may be cases where a design that performs worse for one attribute is preferred based on its overall utility across all attributes.

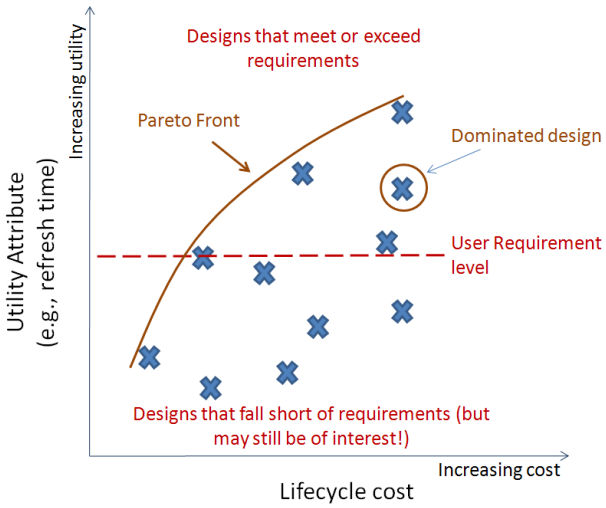

Figure 1 Example Tradespace (each x indicates a unique point design)

II. Defining the Tradespace

A tradespace represents a set of all potentially valuable design alternatives that could be developed in response to a particular design problem[2]. The key factors that are represented in the tradespace include vectors of *design variables*, parameters representing tradable aspects of design alternatives (i.e., the physical composition of alternate

potential designs), and their estimated performance *attributes* (i.e., their performance with respect to stakeholder preference), often incorporating significant uncertainty at the conceptual stage (Figure 2).

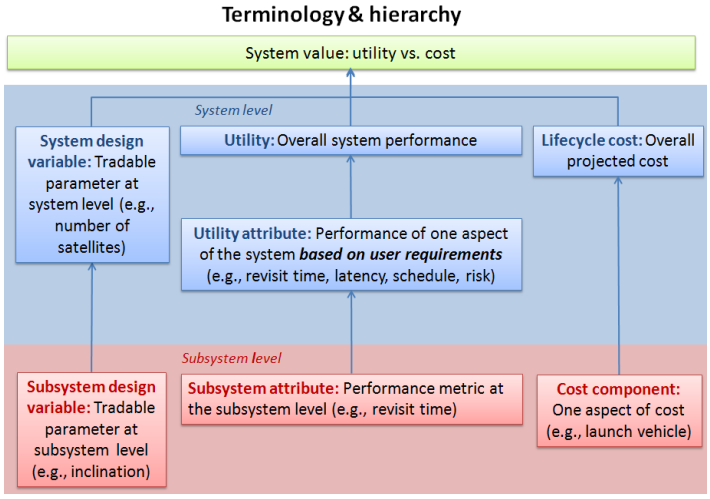

Figure 2 Tradespace Exploration Hierarchy & Terminology

Defining *design variables* involves an iterative brainstorming process between system and subsystem levels, and should reflect many potential architectures that might conceivably meet the mission's objectives. For example, the initial constellation tradespace included a range of options, including nanosatellite platforms, hosted payloads, controlled and uncontrolled constellations, as well as non-satellite options. Since tradespaces are constructed using concept neutral criteria (perceived benefits and costs), they allow for a comparison of vastly different concepts on the same basis[3].

Attributes are quantitative representations of stakeholder preference: each captures the magnitude of one aspect of utility. Defining attributes flows from stakeholder needs identification and preference elicitation[4,5]. Attributes are bounded by a range from the least acceptable performance (beyond which the system is useless to the stakeholder) to the most desirable performance (beyond which there is no extra benefit). A set of attributes should be complete, non-redundant, operational, decomposable, minimal, and perceived independently. Satisfaction for performance at different levels of attributes can be assessed using single attribute utility functions that typically vary from 0 (minimally acceptable) to 1 (most desirable)[6].

III. Populating the Tradespace by Iteration

Most space design work is performed by specialist groups at the subsystem level. A bottom-up iterative approach allows subsystem designers to focus on their area of expertise and optimize performance for a given set of constraints, while making initial assumptions regarding the role of other subsystems (Figure 3).

The first step in this approach is the definition of utility-driving attributes at the system level (Figure 3, top left). The goal is to satisfy the key decision makers with preferences on performance and cost (whether that is just the user, or it is the user and the customer). For the constellation example, since the sole stakeholder was deemed to be the end user, utility attributes were directly obtained from a decomposition of user requirements into quantifiable metrics (one for each major quantifiable requirement, e.g., data refresh rate, space to ground latency, accumulated time on target from all satellites for a defined time period).

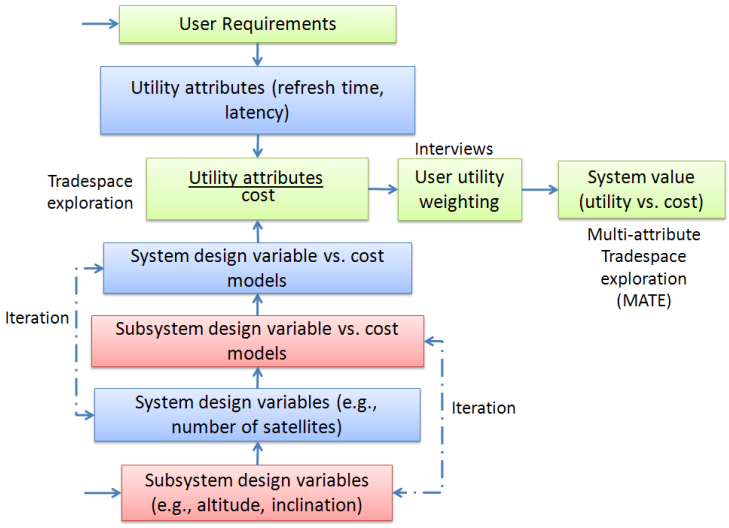

Figure 3 Tradespace Exploration Flow

Once the utility attributes are defined, the second step is to populate the tradespace using design variables (Figure 3, bottom left). The process of selecting design variables to consider is complicated by myriad interactions. Typically, system level design variables are proposed using expert

knowledge through inspection of the utility attributes (i.e. identifying specifiable design factors that drive performance). This proposal process relies upon expert engineering knowledge on how system level design decisions flow down to the subsystems. For complex designs, assumptions at the system level may preclude deeper insights that would be available through exploration of lower level (i.e. subsystem level) design variables.

Design variables at the system level (e.g., orbit) can be decomposed into *subsystem design variables* specific to one or more subsystems (e.g., 'orbit' can be decomposed into 'plane', 'inclination', 'spacing, etc.). Coupling between design variables in the tradespace is achieved using iterations, during which each subsystem is analyzed in relative isolation, but at the completion of which, the next iteration is seeded by analyses from all subsystems during the last iteration.

For the satellite constellation example presented here, the design team was composed of five expert teams, each focused on a different subsystem: orbit, payload, bus, ground, and launch. In order to increase efficiency, subsystem teams need not account for interaction with other subsystems during each iteration; rather, they seek to optimize their subsystem for specific sets of conditions (e.g., minimize revisit time for a given number of spacecraft, or minimize the number of spacecraft to meet a particular revisit time) while holding all other design variables constant. Each of these point optimizations results in a *boundary design* at the system level tradespace. For example, the orbit subsystem boundary designs were defined to minimize revisit time (targeted to meet the user requirements) with either controlled or uncontrolled constellations while using either (i) a minimum number of satellites, or (ii) a minimum number of orbital planes. Within these specifications, the orbital group was left free to optimize their orbital parameters (inclinations, planes, phasing, etc.).

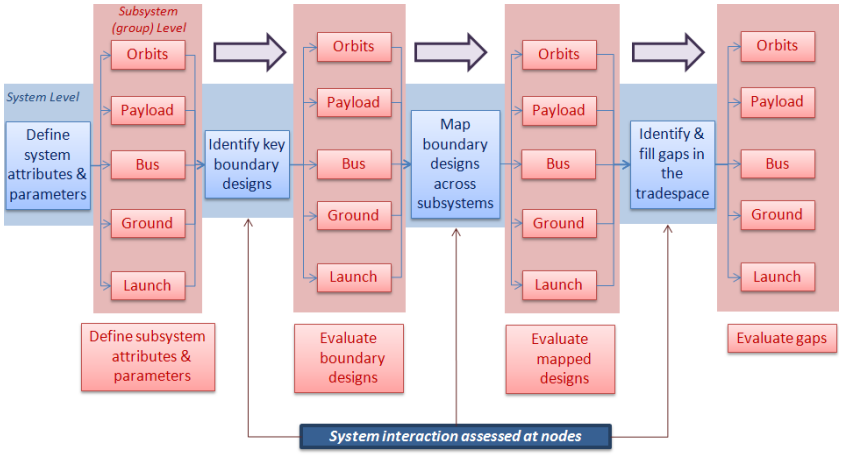

Figure 4 Populating the tradespace by iteration

Note that even if two subsystem teams optimize with the same goal, they may not necessarily produce the same (or even compatible) designs. Additionally, although each subsystem group defines design variables independently, in practice this could lead to infeasible, impractical, or impossible systems because choices made for one subsystem can impose constraints and costs across the other subsystems. For example, orbital variables cannot be defined in isolation of launch, and ground station performance (e.g., downlink time and data rate) cannot be defined without consideration of payload and orbit. Thus, a limited amount of communication between expert groups is required to ensure that the proposed boundary designs are realistic. For example, the orbit design variable was decomposed into the subsystem design variables altitude, orbit planes(s), inclination(s), eccentricities, and special types of orbits (sun-synchronous, dusk-dawn, etc.).

The third step is to evaluate interactions at the system level: i.e., how do the design variables at the subsystem level combine into system level design variables, and how do these in turn influence the utility attributes? For example for the satellite constellation, the data refresh rate attribute was influenced by design variables selected by several design teams (revisit time - orbit, processing time - payload, ground station location - ground). This subsystem interaction is assessed and propagated at the system level at three key nodal checkpoints (Figure 4).

Once the impacts of subsystem design variables on the performance of the system are understood for the first iteration, the fourth step is to initialize the next iteration by mapping subsystem-generated designs amongst the other subsystems. This process introduces the potential for a full factorial expansion of all boundary designs (e.g., 5 orbital sub-designs, 3 ground sub-designs, and 2 launch sub-designs could result in a maximum of 5×3×2=30 designs). However, there is also the potential for significant overlap (e.g., certain ground station configurations only make sense for certain orbital configurations), so somewhat less than a full factorial expansion typically results.

Additionally, it is sometimes possible to eliminate infeasible or low value designs by inspection at this stage instead of carrying them forward in the analysis. This mapping process is illustrated in Figure 5.

Finally, as a fifth step, there is an opportunity to add missing "gap" designs based on an initial exploration of the tradespace. This can be used to fill in areas of the tradespace that are incomplete or were missed. Insight gained by examining interactions at the system level can aid in this process by suggesting the inclusion of interesting designs that were previously neglected. This is typical when two or more subsystems share subsystem level elements of system level design variables, attributes, or costs. For example, the launch analysis might suggest that significant cost saving could be achieved by implementing particular orbital configurations that permit launch sharing (either with other satellites in the same constellation or with other planned missions).

A. Utility Attributes

For each of the subsystem level analyses, including the first boundary designs, the subsystem teams must determine their relevant subsystem level attributes, design variables, and cost drivers, and how these impact the system level tradespace. This involves tracing subsystem design variables to subsystem attributes and thence to system level utility attributes. Since this process is performed at the subsystem level by expert teams, subsystem engineers can use higher fidelity models to evaluate the impact of their subsystem at the system level (unlike previous studies where subsystem-system interactions have been decomposed from the system level[1]). Many system level attributes, design variables, and cost drivers are multidimensional parameters composed of individual subsystem level components. For example, "latency" (a system level attribute describing how long it takes data to get to the user), is composed of at least:

1. Time from data collection to downlink
2. Time taken to downlink the data set
3. Time to taken distribute to data from ground station to the user
4. Time to process data

Moreover, in this example the time required to downlink a set of data depends on several

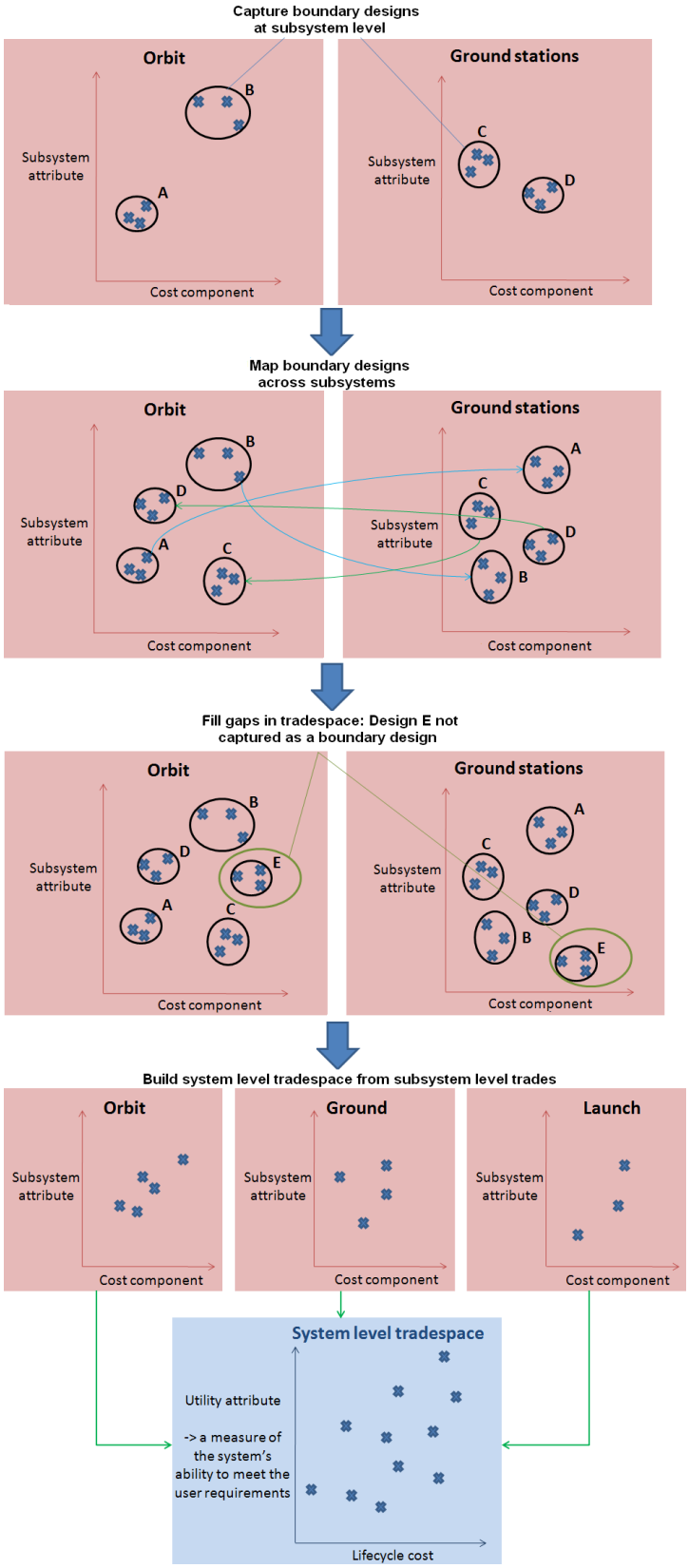

Figure 5 Assembling a system level tradespace

characteristics of the spacecraft and ground station subsystems, such as frequency band, gain, noise, and ground station location and coverage. Similarly, the system level attribute "refresh time" – the time from when one data set is available to the user until the next becomes available - is composed of the subsystem level attribute revisit time (orbit) in addition to latency contributions.

It is not always straightforward to define a metric that quantifies a particular attribute either at the system or subsystem level. For example, revisit time varies not only by satellite in the constellation, but by orbital parameters, orbital epoch, location of interest on the Earth, and possibly by other factors such as weather (intervening clouds), planned outages (maneuvers), and spacecraft power budget (duty cycle). Thus, in defining a metric to quantify an attribute, highest priorities should be given to a) utility to the user, and b) the ability of the metric to be measured. It is possible to define an attribute by weighting two or more related metrics. For example, for the satellite constellation study, 'typical low end' and 'worst case' performance were both considered to be equally relevant – thus refresh time was composed of equal contributions of both the 90^{th} percentile low end case and the worst case (assuming nominal conditions for both).

Since an attribute is a reflection of the utility of the system to a stakeholder, it is possible to quantify aspects of performance related to risk and schedule, or define any other attribute metric that suits the particular problem (including cost as an attribute would be redundant, since it is explicitly represented in the tradespace). For this example, since coverage for a partially deployed constellation (or one with a temporary or permanent loss of a satellite) was highly relevant, "redundancy" – defined as performance metric with the loss of one spacecraft - was included as a system level attribute. However, studies have shown that having too many attributes (e.g., more than 5-7) per stakeholder makes it difficult to accurately elicit stakeholder preferences for each attribute when all attributes are considered simultaneously[5]. Thus, it is probably a good idea to limit the number of system level attributes considered per stakeholder to fewer than seven, and ideally no more than five. For the satellite constellation example, five system level attributes were considered (see Table 1).

Table 1. Utility attributes and contributing subsystems for the satellite constellation example

Utility attribute	Contributing subsystems
Refresh time globally	• Orbit (time to revisit anywhere on the globe) • Launch (only as an orbit driver) • Ground (time to revisit a ground station after collecting data & downlink/distribute data)
Refresh time over particular area of interest	• Orbit (time to revisit the selected area) • Launch (only as an orbit driver) • Ground (time to revisit a ground station after collecting data & downlink/distribute data)
Time on target (accumulated for a particular point of interest on Earth and contributed by all spacecraft for a defined time period)	• Orbit (accumulated observation time) • Launch (only as an orbit driver) • Payload (footprint, duty cycle)
Latency	• Orbit (time to downlink after collecting data) • Launch (only as an orbit driver) • Payload (data volume and format, downlink rate) • Ground (time to downlink, distribute, & process data)
Redundancy	• Bus (reliability/availability) • Orbit (refresh time with loss of spacecraft) • Ground (refresh time with loss of ground station)

B. Design Variables and their Associated Cost Drivers

Design variables defined at the system level must be decomposed and allocated for consideration at the subsystem level; not all design variables are relevant to all subsystems, and some are heavily influenced by multiple subsystems. For example, launch feasibility determines what orbital configurations are possible. Most design variables have associated cost drivers; some of these can be determined at the subsystem level, while others can only be quantified at the system level. Table 2 shows an example of design variables for the satellite constellation study at subsystem and system level and their associated cost drivers.

Table 2. Example design variables and their associated cost drivers

System design variables	Subsystem design variables	Associated cost drivers
Spacecraft/bus	• Number of spacecraft • Nanosat/microsat/smallsat/hosted payload on larger satellite • Controlled/uncontrolled	• Design & build • Launch • Operations
Orbital characteristics	• Orbit planes • Eccentricities • Separation • Special cases (e.g., sun-synchronous)	• Launch • Propulsion & delta-v
Launch vehicle	• Number of satellites per launch • Primary vs. secondary payload • Replacement availability	• Launch vehicles • Launch operations
Payload	• Selection of payload for mission	• Equipment design & build cost • Operations
Downlink	• Frequency • Bandwidth • Geographic availability	• Equipment design & build cost • Regulatory issues
Redundancy, reliability, & replacement strategy	• On-orbit (hot or cold) spares vs. replacement • Spacecraft reliability (expected lifespan)	• Initial cost vs. replacement cost
Ground station	• Number and placement of stations • Build vs. buy • Stationary vs. mobile	• Equipment design & build cost • Start-up vs. operational costs
Data processing & handling	• Storage • Security	• Start-up costs • Operational costs

A set of design variables, once recomposed and assessed at the system level, form a design. Table 3 shows an example of designs considered for the satellite constellation study. These represent feasible pairings of subsystem design variables to compose full point designs. As shown in Table 3, nanosatellite, microsatellite, and hosted payloads on larger satellites were considered. Satellites with and without propulsion, and in polar and mixed inclination orbits were considered (all satellites were in LEO). Two ground station configurations were considered, "basic" (with 4 ground stations), and "upgraded" (with six ground stations). Each of these choices had impacts on the system and pairings with other subsystem design variables. For example:

- Orbital designs requiring controlled spacecraft were combined with the bus option that included propulsion.
- Designs that included only polar orbit used only polar ground station configurations, while designs that included low inclination orbits were combined with equatorial ground stations.
- A hosted payload would have less choice of orbital parameters because these would be dictated by the primary mission; thus, it was assumed for the "hosted payload" design that the actual orbital parameters would be reflective of "typical" historical satellites.

Table 3. Example designs

Design number	Bus	Propulsion?	Orbit type	Ground station configuration
1	Microsatellites	Yes	Mixed polar	Basic
2	Microsatellites	No	Mixed polar	Basic
3	Microsatellites	No	Mixed including equatorial	Upgraded
4	Nanosatellites	No	Mixed polar	Basic
5	Hosted payloads on large satellites	Yes	Mixed polar in historical locations	Basic
6	Microsatellites	Yes	Mixed polar	Upgraded

IV. Exploring the Tradespace by Attribute

Once we have populated the tradespace and have determined how the design variables relate to attributes and costs, we can explore the individual attribute tradespaces. In order to account for uncertainties in the relationship between the design variables, costs, and attributes, the analysis for the satellite constellation study uses a 5-point Monte Carlo, and seeks to examine the performance of each of the designs as an 'ellipse of uncertainty' rather than a single point. To this end, for any given design variable, the associated attribute and cost was selected from a normal distribution with a defined mean and standard deviation instead of defining a fixed relationship. Other types of distributions or relationships could be used.

In tradespaces, the Pareto Front includes design alternatives that have the highest benefit for a given cost, or the lowest cost for a given benefit. Alternatives close to the Pareto Front are highest in value. Alternatives below and to the right of the Pareto Front are dominated by higher value alternatives, and thus yield lower expected value. However, the "best" design still depends on how much a user is willing to pay for a given utility: i.e., the choice between lower cost and lower performance or higher cost and higher performance comes down to user preference.

Figure 6 shows the resulting tradespace for global refresh time. Design 5 (hosted payload, see Table 3) yields the worst global refresh time at the lowest cost. Designs 3 and 4 are close to the knee in the Pareto Front and offer amongst the best refresh time at a somewhat higher cost, although design 3 has more cost uncertainty and 4 has more performance uncertainty. Designs 1, 2, and 6 are dominated by more efficient designs. Design 2 ends up as the most expensive option even though its satellites are uncontrolled and theoretically less expensive, because more satellites (and launches) are required to ensure adequate global coverage (i.e., the analysis assumes the uncontrolled satellites are subject to launch errors and drift).

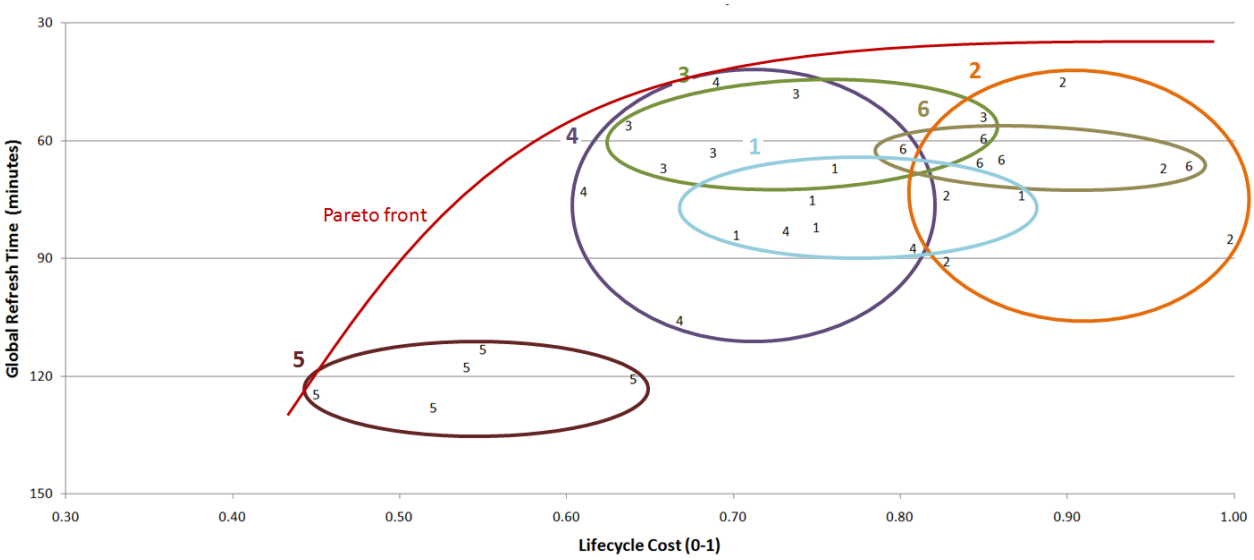

Figure 6 Example global refresh time vs. lifecycle cost (design numbers are defined in Table 2)

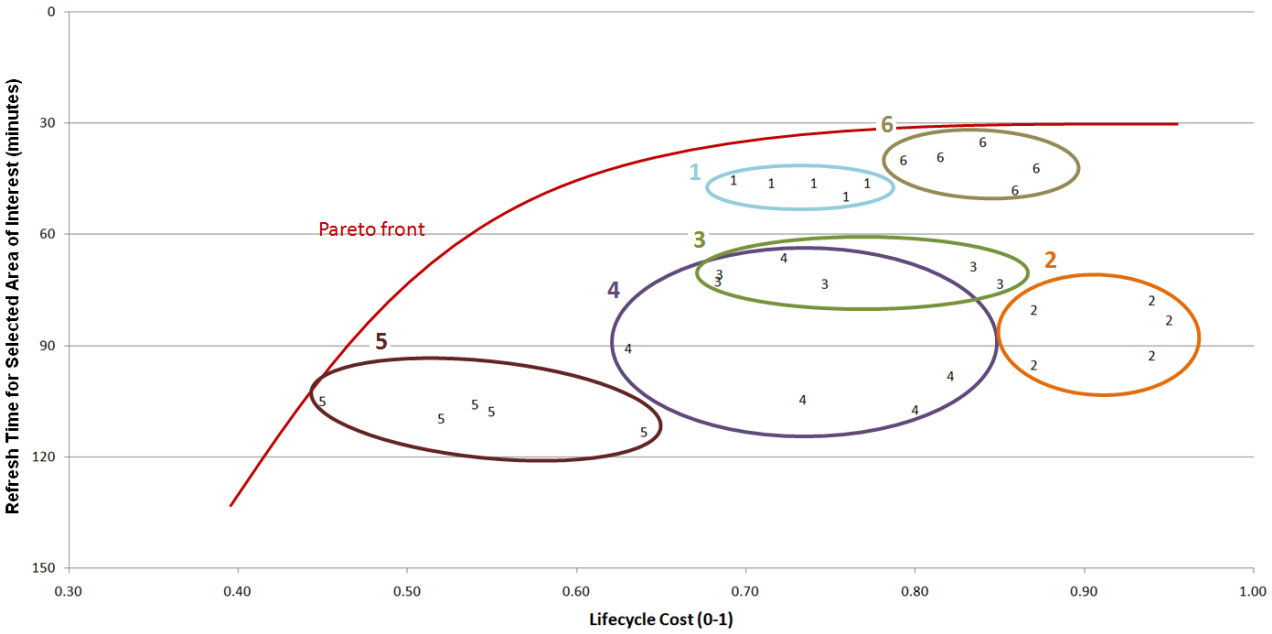

Figure 7 Example refresh time for a selected area of interest vs. lifecycle cost

Figure 7 shows the tradespace for refresh time over the selected area of interest. Again, design 5 offers low end performance at a lower cost. However, for refresh time for the selected area, designs 1 and 6 dominate the other designs at the high end of performance. The difference in performance between global refresh time and refresh time for the selected area is primarily a result of orbit selection: since the selected area of interest was at a Northern latitude, optimum refresh time is achieved with more polar satellites – equatorial low Earth orbit satellites fail to improve refresh time over that area. However, relying exclusively on polar satellite yields a higher overall refresh time compared with employing a mix of polar, mid-inclination, and equatorial satellites.

Figure 8 shows the resulting tradespace for time on target. Since time on target is measured for a single point on

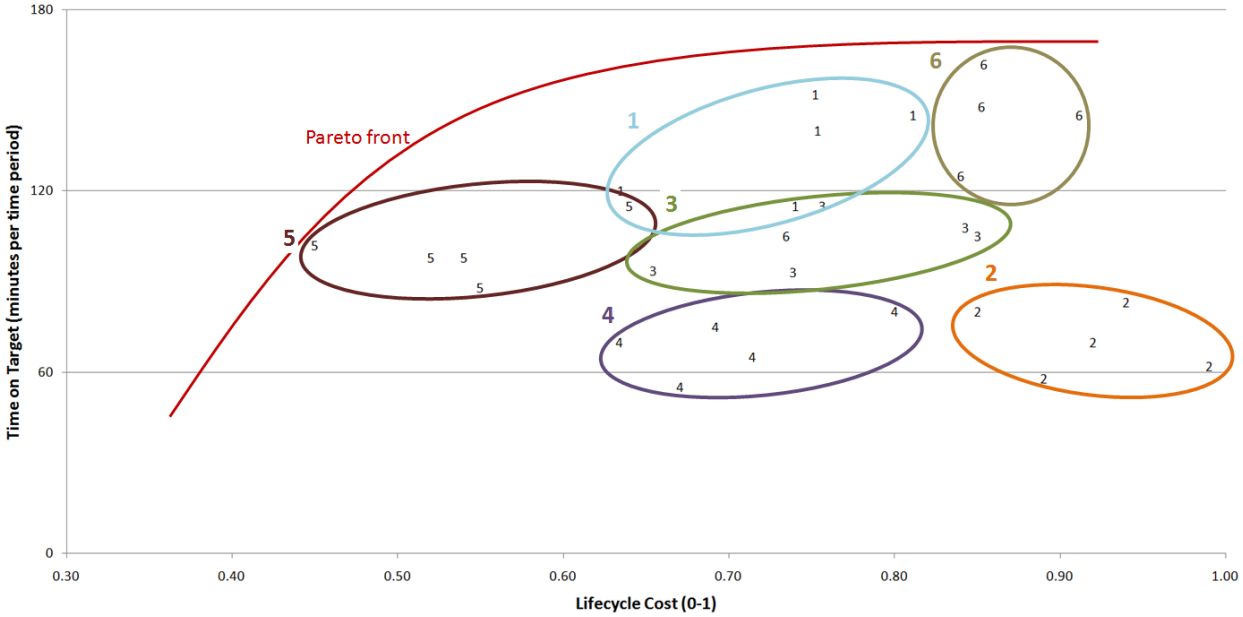

Figure 8 Example accumulated time on target vs. lifecycle cost

Earth selected at a Northern latitude, the distribution of designs in the tradespace for time on target resembles the tradespace shown in Figure 7.

Figure 9 shows the tradespace for latency. Since latency is heavily dependent on ground station location, design 6 with additional ground stations shows the best performance (albeit at a higher cost). Note also that although design 3 also has additional ground stations, it does not show a marked decrease in latency over the other designs because the extra ground stations are required merely to access the equatorial satellites and fail to have much overall effect in decreasing latency.

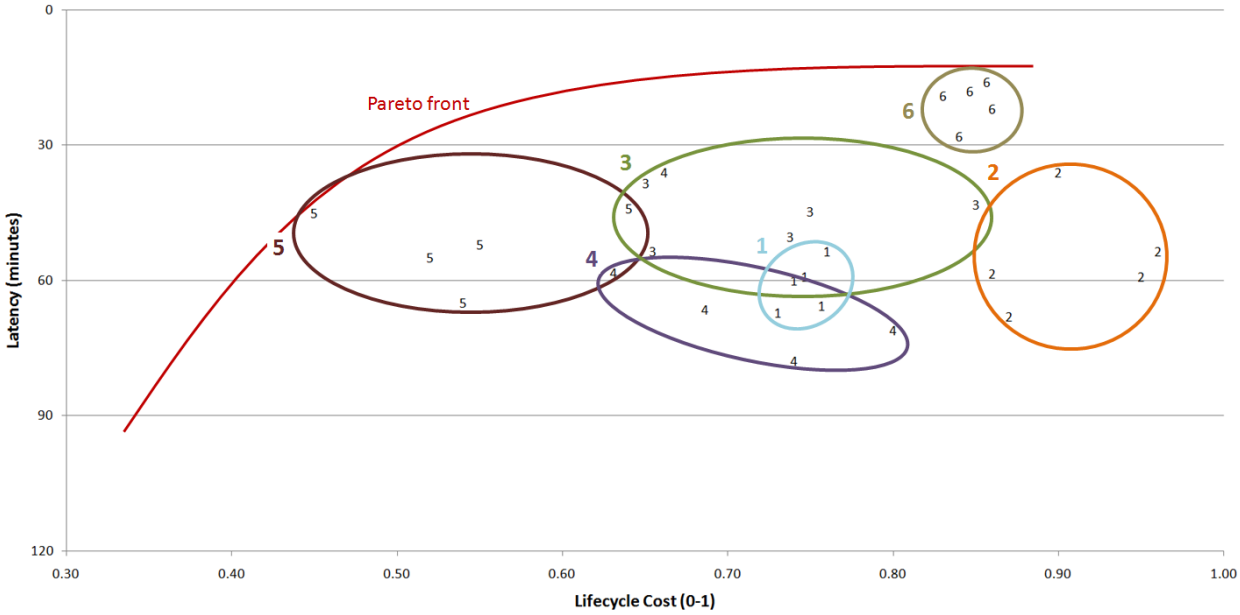

Figure 9 Example latency vs. lifecycle cost

V. Multi-Attribute Tradespace Exploration

An exploration of the tradespace allows a direct comparison of the cost-benefit tradeoffs offered by competing subsystem-generated designs, now combined to represent distinct system design alternatives. In order to compare design value across the five system attributes (global refresh time, refresh time for a selected area of interest, time on target, latency, and robustness), we employ the process of Multi-Attribute Tradespace Exploration (MATE)[4]. In MATE, preferences with respect to the attributes (themselves derived from the user requirements) are mapped into utility functions. As shown in Figure 6, multi-attribute utility combines single-attribute utility functions (e.g., refresh time, probability of detection) into a single metric that quantifies how a decision maker values different attributes relative to each other. Having an aggregate utility metric to reflect decision maker satisfaction helps to clarify the tradespace so that high value designs can be identified and examined in more detail.

A. Defining Utilities

The first step in this process is to define the utility associated with each individual attribute. Utility is an appraisal of the benefit under uncertainty of an attribute to the user. For example, performance that falls short of a user requirement could be designated 0, and performance that meets a requirement could be designated a 1. Note that a utility of 0 is defined as the minimally acceptable (but still acceptable) level of utility[7]. Thus, even more severe than a utility of 0, a requirement that is absolutely essential (i.e., a 'showstopper') could *exclude a design from further analysis* if it fails to be met (i.e., at a level lower than a threshold requirement[8]). However, in most situations there is an additional utility gained by not just meeting, but actually exceeding requirements. Similarly, falling slightly short of some requirements might be acceptable if it significantly reduces the cost.

Two questions should be considered regarding utility:

- How much benefit can be gained by exceeding requirements, and what is the distribution of this additional utility?
- What is the utility impact of failing to meet requirements, and how is this distributed?

Single attribute utility functions can be defined to represent perceived benefit under uncertainty for several types of attributes (Figure 11). A *binary attribute* implies that meeting the requirement provides the best possible performance with regards to that attribute (utility = 1), or failing to meet the requirement provides the minimial acceptable level of utility (but still acceptable) (utility =0).

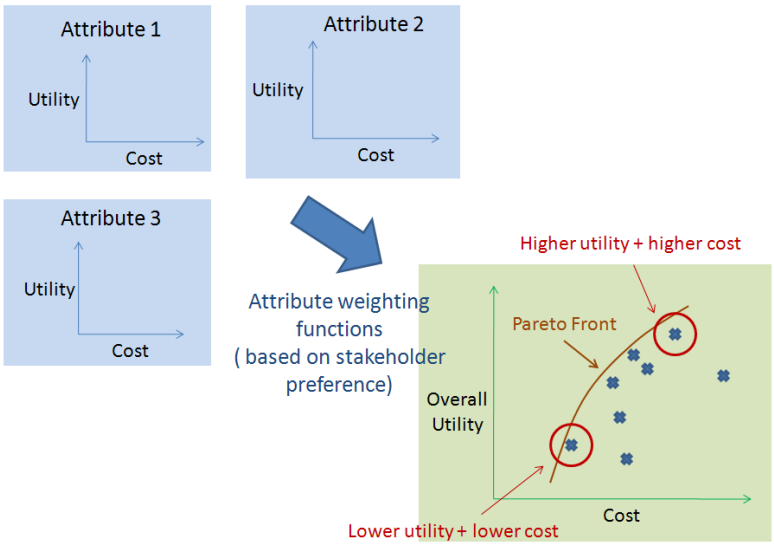

Figure 10 Utility weighting to determine multi-attribute utility (MAU)

Alternatively, a binary attribute could be defined such that meeting a requirement provides the best possible performance with regards to that attribute (utility = 1), or failing to meet the requirement eliminates the design from further analysis (utility undefined). Binary attributes defined in this way are constraints, since they must be satisfied for feasible designs. For the satellite constellation example, binary attributes aren't even considered at the tradespace level, because designs that failed to meet a minimum performance level were eliminated by the design teams before making it to the system level.

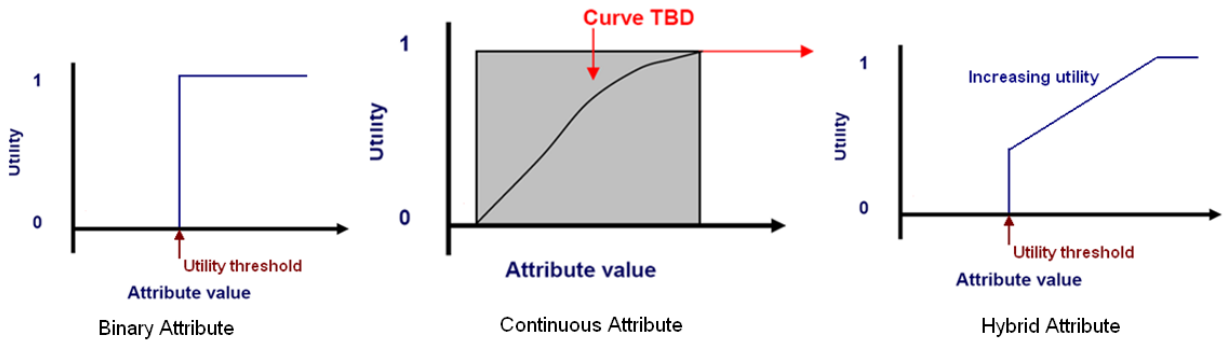

Figure 11 Example attribute types and their associated utility functions

A *continuous attribute* has utility increasing with the attribute value between some bounds where utility is defined as 0 and 1. Alternatively, in the case of a negative relationship, more utility could be provided by a decreasing attribute, as is the case in the satellite constellation example with refresh time and latency ("more is better" vs. "less is better"). Most attributes fall into this category because increasing performance generally provides more utility. However, there may be some point at which improved performance has no effect. Conversely, there generally is some point at which the utility can be considered zero (or considered a constraint that eliminates some designs from further analysis). A *hybrid attribute* is some combination of binary and continuous (i.e., piecewise continuous, discrete), or is customized to solve a particular problem. Note that the utility functions do not have to be linear.

For utility functions, the satellite constellation example assumes (see Figure 12):

- All attributes are continuous.
- Meeting a requirement on average achieves a utility of 0.5.
- Doubling or exceeding double a requirement on average is assigned a utility of 1.
- Falling short of a requirement by half on average is assigned a utility of 0.
- Designs falling short of a requirement by more than half on average are excluded from the analysis.
- Utilities between 0 and 1 are interpolated on a linear scale, giving a utility value of between 0 and 1.

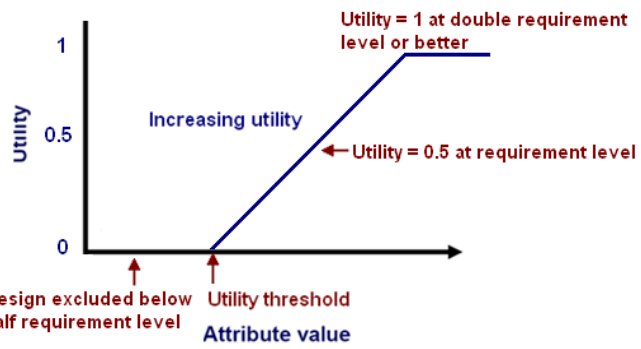

Figure 12 Utility function used for satellite constellation example

Thus a single attribute utility of 0.5 can be interpreted as meeting the requirements "on average". A utility higher than 0.5 performs better than the requirements "on average", and a utility less than 0.5 fails to meet the requirements "on average". However, this does not mean that a utility of higher than 0.5 necessarily achieves all the requirements using this definition of utility.

B. Combining Utilities

The next step is to define utility weightings for each utility attribute so that all utilities can be combined to find a multi-attribute utility (MAU). The insight gained from the process permits a re-analysis at the individual attribute level so that no information is lost in the process, while maintaining manageability through a smaller number of decision metrics. If there are no cross-term benefits for the attributes (i.e., each attribute contributes to utility independently), then using a simple weighted sum is a reasonable method for aggregating single attribute utilities[5,7], and this assumption was applied here:

$$U\ \ x\ \ = \sum_{i=1}^{N} k_i U_i\ \ X_i \tag{1}$$

Where U(x) is the multi-attribute utility (MAU), k_i is a user weighting for the ith attribute, and $U_i(X_i)$ is the utility of the ith attribute. Note that k_i could be a function rather than a scalar, and that it makes no difference if attributes are coupled (i.e., global refresh is correlated with refresh at the selected area of interest), as long as their utilities can be decoupled (i.e., there is *perceived* independence of the attributes).

For the satellite constellation example, utility weightings were determined from a conjoint analysis. Conjoint analysis presents the user with a survey of alternatives with varying attributes and allows them to specify which alternative they prefer in each case. The five utility attributes were broken into example designs offering 'low', 'medium', and 'high levels' levels of performance. The conditions for the attributes were each given a nominal rank contribution of "1" (low), "2" medium, or "3" high, and each design had a nominal total rank of 10.

For example, a design having:

Attribute 1. low performance global refresh (1),
Attribute 2. low performance refresh for the selected area of interest (1),
Attribute 3. high performance time on target (3),
Attribute 4. high performance latency (3), and
Attribute 5. medium performance robustness (2)

would have a nominal rank of 1+1+3+3+2=10. In the conjoint analysis process, the user is presented with two designs having the same nominal rank (10 in this case) and asked to choose which they prefer (A or B). This yields a true rank for each of the three designs. By comparing the true rank and nominal rank, we can infer weightings for the individual attributes. A breakdown of weightings on the utility attributes for this example is shown in Figure 13.

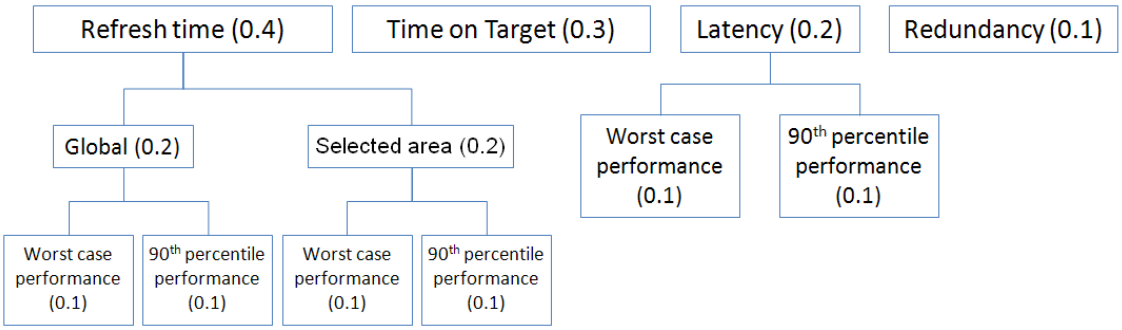

Figure 13 Example attribute weights from conjoint analysis

C. Example Multi-Attribute Tradespaces

Armed with the utility functions and weightings defined above, a multi-attribute tradespace can be constructed with aggregate benefit (as multi-attribute utility) and aggregate cost (as normalized lifecycle cost) (Figure 14). This tradespace retains the 5-point Monte Carlo distributions generated in assembling the individual attribute tradespaces.

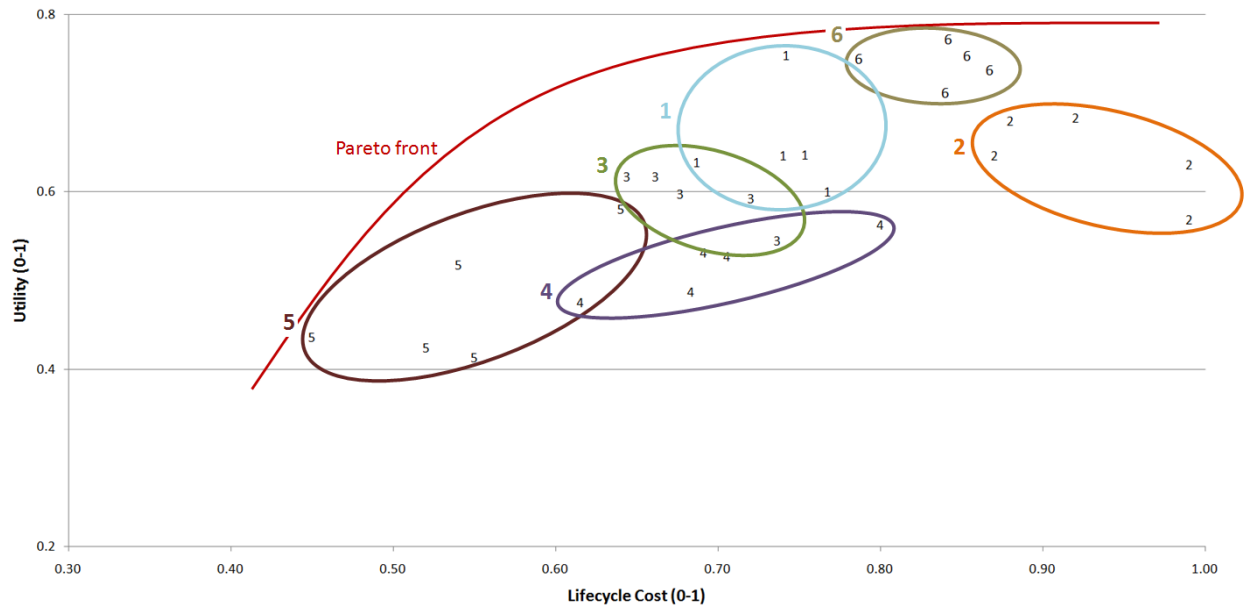

Figure 14 Example multi-attribute utility vs. lifecycle cost

As shown in Figure 14, designs 6, 5, 3, and 1 all offer relatively efficient performance vs. cost tradeoffs, while designs 2 and 4 are dominated by more efficient solutions. The actual best choice between designs 6, 5, 3, and 1 depends on entirely on user preference; i.e., is it worth paying more to get more? Another consideration is risk: for example, design 6 provides excellent performance with much less uncertainty than design 1.

The process of assembling and examining the multi-attribute tradespace can also reveal predefined biases and lead to a re-examination of user requirements and user preference in the form of utility attribute definitions, utility

functions, and utility weightings. For example, the fact that some designs, which perform well in most respects, were excluded from the tradespace because they failed to meet the minimally acceptable level for a particular requirement, might suggest that the minimally acceptable threshold for that requirement could be reduced. Additionally, the tradespace can be used to highlight areas for further analysis. For example, a user might decide to examine other hosted payload options to yield further cost savings at the cost of reduced performance or increased risk relative to design 5.

VI. Discussion

A system level tradespace is specific to both the mission and stakeholder preferences, and is subject to uncertainties and potential errors in both. Additionally, the tradespace is sensitive to assumptions in how design variables relate to attributes and cost. The resulting tradespace captures utilities that are completely dependent on:

- the attributes definitions (e.g., what specific point(s) on Earth are selected to measure time on target),
- the single attribute utility functions definitions (e.g., how does the level of a particular attribute relate to the utility it provides to the stakeholder?), and
- the utility function weightings (e.g., how important is refresh time compared with time on target?).

In fact, because a tradespace is problem-specific, the process of assembling the tradespace provides as much insight into the design problem as examining the assembled tradespace. Engaging the subsystem design teams early in the process capitalizes on expert knowledge and increases understanding of the mission early in the design process, saving time and reducing risk that would otherwise be carried to subsequent design steps. Furthermore, engaging the system users early in the process leads to a re-examination of user requirements and mission objectives. Additionally, involving the system users in the design process increases their confidence that the design team is examining potential solutions in a systematic and unbiased way.

Although each of the designs shown in Figure 14 represents a particular instance of a configuration (e.g., it would be possible to define slightly different orbital parameters or ground stations for each of the designs), early expert engagement means that most performance optimizations have already been performed at the subsystem level. For example, while it would be possible to define different configurations of hosted payloads for the mission, the particular configuration representing design 5 is likely to represent a high value option within the hosted payload tradespace because it has already been optimized by subsystem design experts.

Instead of generating a full system tradespace that might contain hundreds or thousands of potential designs as most previous tradespace exploration case studies have done[1,4], the iterative approach to assembling a tradespace presented here has allowed the application of expert knowledge to generate a filtered subset of the full tradespace containing only high value solutions. A potential drawback to this bottom-up approach is that there is a potential for missed solutions (not considered by the subsystem teams) that may offer even higher value than one of the designs selected for analysis by the subsystem design teams. However, this bottom-up iterative approach is particularly appropriate for industry problems, where it may be difficult to assemble full computational system models that capture problem-specific relationships between design variables, attributes, and costs.

VII. Conclusions

Tradespace exploration is a powerful tool to examine a large number of potential designs on a common basis, but it can be difficult to apply to an industry engineering problem, where expert teams have traditionally focused on the design of particular subsystems. This work presented an iterative bottom-up approach to populate the tradespace for a satellite constellation. This approach capitalizes on the ability of expert design groups to work autonomously to optimize to particular goals within their subsystem, while capturing interactions between subsystems at the system level. As such, it offers advantages over a top-down computational partitioning of the design space for design problems where it is difficult to assemble full computational system models. The approach also enables a targeted generation of valuable sets of designs (through subsystem optimizations and guided “gap” design identification), elucidating subsets of a full tradespace with hundreds or thousands of designs, most of which may provide little

value; it thus represents a compromise between full tradespace exploration and a point design centric approach. Additionally, early involvement of both design teams and stakeholders improves understanding of the design tradeoffs, user requirements, and mission objectives. The result is a tradespace and a process that takes advantage of expert knowledge and provides a systematic approach to evaluating subjective value perceptions of performance and cost tradeoffs on a common basis.

Acknowledgments

This work was carried out in the Mission Development Group at COM DEV International, Ltd.

References

[1] Ross, A.M., Hastings, D.E., Warmkessel, J.M., and Diller, N.P., "Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design," *AIAA Journal of Spacecraft and Rockets*, Jan/Feb 2004.

[2]Ross, A.M. and Hastings, D.E., "The Tradespace Exploration Paradigm," *INCOSE International Symposium 2005*, Rochester, NY, July 2005.

[3] Ross, A.M., McManus, H.L., Rhodes, D.H., and Hastings, D.E., "Revisiting the Tradespace Exploration Paradigm: Structuring the Exploration Process," *AIAA Space 2010*, Anaheim, CA, September 2010.

[4] Ross, A. M., Diller, N. P., Hastings, D. E., and Warmkessel, J. M. "Multi-Attribute Tradespace Exploration with Concurrent Design as a Front-End for Effective Space System Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, 2004, pp. 20-28.

[5] Keeney, R. L., and Raiffa, H., *Decisions with Multiple Objectives--Preferences and Value Tradeoffs*. 2nd ed. Cambridge: Cambridge University Press, 1993.

[6] Ross, A.M., McManus, H.L., Rhodes, D.H., and Hastings, D.E., "A Role for Interactive Tradespace Exploration in Multi-Stakeholder Negotiations," *AIAA Space 2010*, Anaheim, CA, September 2010.

[7]Thurston, D. L., "Real and Misconceived Limitations to Decision Based Design with Utility Analysis," J. of Mech. Des., Vol. 123, No. 2, 2001, pp176-182.

[8]*Defense Acquisition Guidebook*, Department of Defense, Washington, DC, 2010.