Comparing and Optimizing the DARPA System F6 Program Value-Centric Design Methodologies

M. Gregory O'Neill*, Howard Yue†, Sreeja Nag‡, Paul Grogan§, and Olivier L. de Weck¶
Massachusetts Institute of Technology, Cambridge, MA, 02139

The Defense Advanced Research Projects Agency (DARPA) System F6 (Future, Fast, Flexible, Fractionated, Free-Flying) Program has the long-term objective of demonstrating that, in certain mission contexts, fractionated spacecraft are a desirable alternative to monolithic spacecraft. Conceptually, fractionated spacecraft consist of physically independent, "free-flying" modules that collaborate on-orbit to achieve a certain level of system-wide performance or functionality. During Phase I of the System F6 Program, four major aerospace companies developed computer-based simulation models, referred to as value-centric design methodology (VCDM) tools, to quantitatively compare the risk-adjusted, net value of comparable monolithic and fractionated spacecraft. This research effort seeks to further learning from the four VCDM tools through two independent investigations, both focused on developing key methodological insights for future value-centric design tool development. The first investigation entails objectively comparing characteristics of the tools such as their respective modeling architectures and quantification of risk-adjusted, net value. And the second investigation entails the optimization of PIVOT, one of the four tools from Phase I, using several complementary multi-disciplinary optimization methodologies. The first major conclusion from this research is that consensus amongst the four VCDM tools from Phase I, per the most valuable fractionated spacecraft is untenable given substantial diversity in their respective modeling architectures, inherent assumptions, and ensuing interpretation of value. The second conclusion is that the solution stability of well-vetted multi-disciplinary optimization methodologies is significantly inhibited by nested uncertainty in a tool, often manifested in non-deterministic objective functions, as found in all of the Phase I VCDM tools.

I. Introduction

The Defense Advanced Research Projects Agency (DARPA) System F6 (Future, Fast, Flexible, Fractionated, Free-Flying) Program has the long-term objective of demonstrating that, in certain mission contexts, fractionated spacecraft are a desirable alternative to monolithic spacecraft.[1,2] Large-scale implementation of the work resulting from the DARPA System F6 Program could lead to a large shift in the design, development, deployment, and ensuing operation of spacecraft for commercial and military space missions. Conceptually, fractionated spacecraft consist of physically independent, "free-flying" modules that each collaborate on-orbit to collectively achieve a certain level of system-wide performance or functionality. Each module in a fractionated spacecraft is composed of various subsystems, and thus a fractionated spacecraft might consist of one module responsible for power generation & storage, another module responsible for communications, another module responsible for the payload, and so on.[3] Part of the System F6 Program Phase I statement of work dictated that four industry-led teams independently develop a value-centric design methodology (VCDM) tool to quantitatively assess the risk-adjusted, net value of fractionated spacecraft, relative to comparable monolithic spacecraft. The four VCDM tools are enumerated in Table 1 and independent discussions of each can be found in conference papers, specifically from the AIAA Space 2009 Conference

*Graduate Research Assistant, Aeronautics and Astronautics, Bldg. E38-550, and AIAA Student Member.
†Graduate Research Assistant, Aeronautics and Astronautics, Bldg. 33-409, and AIAA Student Member.
‡Graduate Research Assistant, Aeronautics and Astronautics, Bldg. 33-409.
§Graduate Research Assistant, Aeronautics and Astronautics, Bldg. 33-409.
¶Associate Professor of Aeronautics and Astronautics and Engineering Systems, Associate Head of Engineering Systems Division, Building E40- 261, and AIAA Associate Fellow.

and Exposition.[4–6] In addition to a proprietary version delivered to DARPA at the completion of Phase I, a version of each VCDM tool was openly distributed to the public via the DARPA Tactical Technology Office System F6 website.[7]

Phase I Team	Tool Name
Lockheed Martin Company (LM)	SVM Tool (System Value Modeling Tool)
Northrop Grumman Corporation (NG)	SVMTool (Space Architecture Design Tool with System Value Modeling Tool)
Orbital Sciences Corporation (OSC)	PIVOT (Pleiades Innovative VCDM Optimization Tool)
Boeing Company (BC)	RAFTIMATE (Risk Adjusted, Flexible, Time Integrated, Free-Flying, Multi-Attribute Tradespace Exploration)

Table 1. Phase I VCDM Tools

In terms of reflection on the Phase I VCDM tools, to date there has only been a brief conceptual discussion of the tools, which, from a methodological standpoint does not offer much in terms of prescriptions for future VCDM tool development.[8] To further evaluate the outcomes of the Phase I value modeling effort, subsequently expanding the current learning from, and contributions of, these appreciable efforts, there is a need for an holistic, qualitative and quantitative investigation of the four Phase I VCDM tools. Therefore, this research first pursues a comparative study that facilitates a qualitative and quantitative investigation and ensuing comparison of the four *publicly available* Phase I VCDM tools with respect to key tool characteristics such as their respective modeling architectures and quantification of risk-adjusted, net value. The main contribution of this facet of the research is a succinct set of best practices for future value-centric design tool development.

Further prescriptions aimed at guiding future value-centric tool development are then formulated through this research's second investigation, applying several complementary multi-disciplinary objective optimization (MDOO) methodologies to PIVOT (OSC). PIVOT (OSC) was selected because Orbital Sciences Corporation was the only industry team from Phase I of the System F6 Program selected for participation in Phase II of the program. The motivation for optimizing PIVOT (OSC) is three-fold: first, a working example of optimizing value-centric design tools is provided; second, lessons learned from the optimization are provided, generalizable to any tool having a non-deterministic objective function; and third, the optimization informs the contributions from the first facet of this research, namely, through the provision of additional best practices for the (future) development of value-centric design tools relying on endogenous or exogenous optimization routines.

II. VCDM Tool Comparative Study

The primary goal of the VCDM tool comparative study is to provide a quantitative, "head-to-head" investigation and comparison of the VCDM tools developed in Phase I of the System F6 Program. The results from this study are separated into three parts. The first part focuses on comparing the characteristics of each VCDM tool such as inputs, outputs, and computational architecture. The second part focuses on discussing each VCDM tool, thereby enumerating differences in each tool's respective spacecraft valuation technique(s) and recommendations resulting from the cases considered in the comparative study, described hereafter. And the third part summarizes the prescriptive insights gained from this facet of the research effort, which will aid in the future development of value-centric design tools. Note, the ensuing discussion regarding the VCDM tool comparison is an abridged version of other documentation of this comparative research effort.[9]

A. Comparative Study Cases

An initial set of eight cases was considered for objectively comparing the VCDM tools respective spacecraft assessment capability; these cases are presented in Table 2. The cases represent a range of missions and fractionated architectures that would be desirable for a VCDM tool to be capable of assessing and valuing. The main drivers in the cases include the type of mission payload, spacecraft architecture, orbital parameters (altitude and inclination), and mission lifetime. Another set of drivers are so called enabling technologies of fractionated spacecraft and include inter-spacecraft communication, wireless power transfer (WPT), and relative navigation and guidance,[3] however, these drivers were not considered for the first set of cases. This decision was made because of substantial differences in the VCDM tools discovered through assessing the first eight comparative study cases, in particular with respect to input specification and modeling of enabling technologies. Thus, cases assessed other than those shown in Table 2 ultimately did not contribute unique insights beyond those formulated using these initial eight cases.

Case #	Mission	Architecture	Altitude (km)	Inclination (deg)	Lifetime (yr)
1	LEO Remote Sensing	Monolithic	500	90	5
2	GEO Telecommunications	Monolithic	35,786	0	5
3	LEO Remote Sensing	2-Module Fractionated	500	90	5
4	GEO Telecommunications	2-Module Fractionated	35,786	0	5
5	LEO Remote Sensing	Monolithic	500	90	10
6	GEO Telecommunications	Monolithic	35,786	0	10
7	LEO Remote Sensing	2-Module Fractionated	500	90	10
8	GEO Telecommunications	2-Module Fractionated	35,786	0	10

Table 2. Comparative Study Case Descriptions

B. Tool Characteristics

The four VCDM tools were characterized and compared through four VCDM diagnostic categories: (1) model structure and implementation, (2) input specification, (3) inclusion of enabling technologies, and (4) outputs and value proposition quantification.

1. Model Structure and Implementation

In the VCDM tools, the "front-end" or user interface is the portion of the tool that users interact with to define, execute, and assess monolithic and fractionated spacecraft architectures. Of the four tools, SVMTool (NG) is the only one that uses a graphical user interface (GUI), which is in the form of an Excel "dashboard" that directly controls an extensive Matlab simulation. SVM Tool (LM) uses command prompts within Matlab as a basic user interface, while PIVOT (OSC) and RAFTIMATE (BC) rely on user-generated scripts. All four tools use Matlab as a "back-end" to perform the simulations. There was no standardized data format or structure amongst the four tools for storing and manipulating data; Table 3 summarizes these findings. In Table 3, the primary data format describes how information is internally represented within the tools; structures contain several multi-typed variables formatted as a custom abstract type whereas arrays contain a list of single-typed variables.

As shown in Table 3, there are nuances each tools respective spacecraft lifecycle simulation method, however, mathematically speaking, they are all discrete event simulations coupled with a Monte Carlo Analysis (MCA). In terms of simulation method, from the perspective of each tool: SVM Tool (LM) uses a Markov Chain to represent system states between development and end-of-life; SVMTool (NG) propagates lifecycle uncertainty from a sampling of continuous (in time) random variables; PIVOT (OSC) calculates the bus subsystem reliability for each month of the mission lifetime from FIT[a] values and then randomly assigns failures on a month-by-month basis; and RAFTIMATE (BC) sequentially uses the next occurring event to update the system state (from a set of possible states) until the end of the simulation is achieved. All four tools use Monte Carlo sampling methods to capture uncertainty in their respective objective functions.

[a]FIT is a unit of measurement denoting failures per billion device hours

Tool	Front-end / User Interface	Back-end	Database / Library	Primary Data Format	Simulation Method
SVM Tool (LM)	Matlab .m-files with prompts	Matlab .m-files	Matlab .mat files and Excel	Structures and arrays	Markov chain, Monte Carlo
SVMTool (NG)	Excel dashboard	Matlab .m-files	Excel	Arrays	Event simulation, Monte Carlo
PIVOT (OSC)	Matlab .m-files	Matlab .m-files	Matlab .mat files	Structures and arrays	Discrete event simulation, Monte Carlo
RAFTIMATE (BC)	Matlab .m-files	Matlab .m-files	Matlab .mat files and Excel	Structures and arrays	Discrete event simulation, Monte Carlo

Table 3. Model Structure and Implementation Overview

2. Input Specification

The input specification for the tools is segregated into two categories: database and architecture design. Database inputs are used to create a library of available payloads, spacecraft buses, and launch vehicles, these are often held relatively constant for a set of spacecraft assessments keeping the available hardware identical for all spacecraft considered. In all of the tools, the database inputs incorporate payloads and buses, most of them being commercial off-the-shelf (COTS) payloads and buses. Conversely, architecture design inputs specify parameters such as orbital parameters, spacecraft characteristics, and the employment of enabling technologies in a fractionated spacecraft such as wireless power transfer; these inputs often change for each spacecraft assessment.

Of the four tools, SVM Tool (LM) has the smallest number of database inputs (26) and RAFTIMATE (BC) has the largest (1,433). And of the four tools, SVM Tool (LM) has the smallest number of architecture design inputs (28) and SVMTool (NG) has the largest (197). Given the generally large number of inputs required for each tool, for any of the tools, appreciable effort and learning is required to comprehensively define the inputs required for a given spacecraft assessment.

3. Enabling Technologies

Table 4 summarizes potential enabling technologies for fractionated spacecraft, which include wireless power transmission (WPT), inter-module communication, and (autonomous) relative navigation. The terminology, enabling technologies, is used because these technologies are purported to enhance the value proposition for fractionated spacecraft.[3] SVM Tool (LM) was the only tool to include all relevant enabling technologies, however, all tools allowed for the use of the most technically mature enabling technology, inter-module communication. And aside from SVM Tool (LM), no other tool allowed for the use of wireless power transfer (WPT). Lastly, it is worth noting that PIVOT (OSC) is the only tool barring the user from disabling the use of enabling technologies.

Tool	Wireless Power Transfer (WPT)	Inter-module Communication	Relative Navigation	Technologies can be Enabled/Disabled
SVM Tool (LM)	Yes	Yes	Yes	Yes
SVMTool (NG)	No	Yes	No	Yes
PIVOT (OSC)	No	Yes	No	No
RAFTIMATE (BC)	No	Yes (Indirect)	No	Yes

Table 4. Enabling Technology Modeling Capability Overview

4. *Outputs and Value Quantification*

Despite the aforementioned differences amongst the four VCDM tools, they all took a similar simulation architectural approach in quantifying a given spacecraft's value proposition; this is depicted schematically in Figure 1.

All tools define the mission context in terms of the type of mission and orbital characteristics, and they have a monolithic and fractionated spacecraft architecture "synthesizer" that defines the architecture to be assessed for a given simulation. In some tools, the architecture synthesizer is manual, whereas in other tools such as SVM Tool (LM), a combinatorial architecture synthesizer is used to auto-create a large set of fractionated architectures based on payload and spacecraft module assignments. Additionally, all tools have models capable of generating the required physics-based characteristics of a given spacecraft based on its respective architecture (F6 Design Definition). And lastly, all tools contain some type of lifecycle simulation to assess a given architecture, as needed to quantify its respective value proposition.

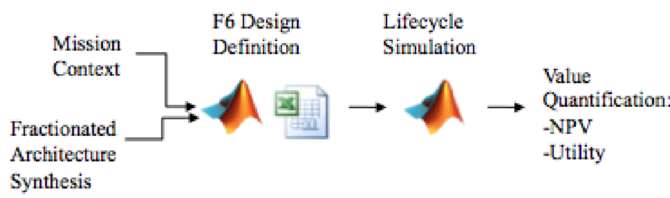

Figure 1. Common High-Level VCDM Modeling Approach of All Four VCDM Tool

As shown in Table 5, two tools, SVM Tool (LM) and PIVOT (OSC), quantify value in terms of net present value (NPV). SVMTool (NG) and RAFTIMATE (BC) output utility and benefit, respectively, versus cost; however, RAFTIMATE (BC) also includes a "monetizer" to convert a given benefit into monetary value. It is important to recognize that the value quantification and assumption differences amongst the four tools suggests that direct comparisons of value propositions between tools is largely inappropriate, but if pursued, such comparisons should be done so with caution.

Tool	Value Quantification
SVM Tool (LM)	NPV ($), value at risk-gain (VARG) curves
SVMTool (NG)	Aggregate utility [0, 1] vs. lifecycle cost ($)
PIVOT (OSC)	NPV ($)
RAFTIMATE (BC)	Present benefit [0, 2] vs. present cost ($), present value ($) via monetizing benefit

Table 5. Outputs and Value Quantification Overview

C. Individual Tool Discussion

Each respective VCDM tool is succinctly discussed hereafter, predominantly based on the authors respective experiences in obtaining results for the eight aforementioned cases (see Section 2.1). A more extensive discussion of the tools can be found in other documentation of this research and the tools.[4–6,9]

1. *SVM Tool (Lockheed Martin)*

The Lockheed Martin Company SVM Tool (LM) is implemented entirely in Matlab, utilizing command line prompts to guide the user through model execution. After defining the user input variables in a design vector, SVM Tool (LM) proceeds to evaluate the net present value of the specified monolithic or fractionated architecture using a Markov Chain and Monte Carlo simulation. SVM Tool (LM) utilizes a payload and spacecraft bus database, stored in Excel, so that commercial off-the-shelf (COTS) payloads and buses can be used in a given spacecraft architecture. SVM Tool (LM) can only model remote sensing mission payloads.

Case Assessment Summary Two of the eight cases shown in Table 2, cases 1 and 3, were successfully analyzed using SVM Tool (LM). From the outputs of SVM Tool (LM) for these two cases, it suggests that for remote sensing missions, the monolithic architecture yields more value than a comparable fractionated architecture, though it is noted in the tool outputs that the "Basic Fractionated" architecture was analyzed out of several possible candidate fractionated architectures, given different launch vehicle combinations, anticipatory development options, etc. The expected NPVs for the monolithic and fractionated spacecraft are predicted to be 5.17 and 4.68 FY08$B, respectively, implying that the monolith has a higher expected value by 0.49 FY08$B or 10%. This result, along with a summary of the quantitative results from all tools, is shown in Table 6.

2. *SVMTool (Northrop Grumman)*

SVMTool (NG) uses a macro-coded object "dashboard" in Excel as a graphical user interface (GUI) that controls a back-end analysis performed in Matlab. Buttons on the dashboard guide the user through the assessment process for a single architecture or set of architectures, each button subsequently executing specific Matlab functions. The Excel spreadsheet tabs alongside the dashboard also double as a data storage medium for database inputs and outputs from the tool, the latter being promptly populated by calling certain Matlab functions from the dashboard. Subsequently, the SVMTool (NG) requires both programming in Visual Basic (Excel) and Matlab, making it unique from the other tools. SVMTool (NG) can only model remote sensing mission payloads.

Case Assessment Summary SVMTool (NG) was successful in running four of the eight cases (1, 3, 5, and 7) in the comparative study, corresponding to those cases that are remote sensing missions. Based on the results from these four cases, as shown in Table 6, one conclusion is that fractionated spacecraft consistently provide more utility (benefit) than comparable monolithic spacecraft, but for a larger lifecycle cost. The additional utility yielded by the most value-competitive fractionated architecture, as compared to a monolith, is projected to cost about 75 FY08$M more in lifecycle cost. As such, there are no instances of fractionated spacecraft being both more beneficial and less expensive than a comparable monolithic spacecraft, for remote sensing missions.

3. *PIVOT (Orbital Sciences)*

PIVOT (OSC) is coded in Matlab and the physical characteristics of the point designs (candidate spacecraft architectures) are determined from third party software and then stored and "front-loaded" into PIVOT (OSC). Thereafter, PIVOT (OSC) performs valuation simulations for each architectures input to the tool. The publicly released version of PIVOT (OSC) uses expected NPV (ENPV) as its value-related objective function. PIVOT (OSC) has the capability to model both remote sensing and telecommunications mission payloads. In PIVOT (OSC), up to three mission payloads can be used simultaneously in a spacecraft.

Case Assessment Summary PIVOT (OSC) was able to run six of the eight cases in the comparative study, cases 1, 3, 4, 5, 7, and 8. However, it is unknown if internal consistency and feasibility was maintained in the tool and ensuing results for the case assessments because certain variables are declared in multiple places in the PIVOT (OSC) source code; hence, changes in some key variables propagated through the assessments in a manner unbeknownst to the authors and not as global variables as they were intended. From the results of the PIVOT (OSC) case assessments enumerated in Table 6, the two-module fractionated spacecraft performing the remote sensing mission has the highest ENPV of the six successfully run cases. It is interesting to note that all six cases were predicted to have a negative ENPV, though as mentioned earlier, the accuracy of the results cannot be corroborated.

4. *RAFTIMATE (Boeing Company)*

The user interface for RAFTIMATE (BC) controls three Matlab scripts: run trade, generate cluster, and simulate cluster, which collectively generate and execute a simulation for a given spacecraft. Two separate databases, a module definition database and a launch vehicle database, feed into the cluster generation process. The core of RAFTIMATE (BC) is a set of Matlab functions that are accessed both in the generation and simulation phases. During each Monte Carlo simulation run, an event history logging the event descriptions, cash flows, and scoring (benefit) contributions is generated and stored. After all runs are complete, the aggregate value metrics (present benefit and cost) are computed and formatted in an Excel spreadsheet, and the present benefit can be converted into monetary units if desired via a monetizer function. RAFTIMATE (BC) can only model remote sensing mission payloads.

#	Mission	SVM Tool (LM) (ENPV in FY08\$B)	SVMTool (NG) (MAU unitless, LCC in FY08\$M)	PIVOT (OSC) (ENPV in FY08\$M)	RAFTIMATE (BC) (MPC in FY08\$M, MPB unitless)
1	Mono., LEO, 5 yr	ENPV = 5.17	MAU = 0.596 LCC = 436	ENPV = −185	MPC = 432 MPB = 1.55
2	Mono., GEO, 5 yr	{tc, alt}	{tc}	{mono}	{tc}
3	Frac., LEO, 5 yr	ENPV = 4.68	MAU = 0.713 LCC = 504	ENPV = −19.0	MPC = 232 MPB = 1.21
4	Frac., GEO, 5 yr	{tc, alt}	{tc}	ENPV = −160	{tc}
5	Mono., LEO, 10 yr	{lt}	MAU = 0.835 LCC = 740	ENPV = −186	MPC = 447 MPB = 1.81
6	Mono., GEO, 10 yr	{tc, alt, lt}	{tc}	{mono}	{tc}
7	Frac., LEO, 10 yr	{lt}	MAU = 0.912 LCC = 892	ENPV = −20.9	MPC = 264 MPB = 1.69
8	Frac., GEO, 10 yr	{tc, alt, lt}	{tc}	ENPV = −160	{tc}

tc - inability to model a telecommunications payload

mono - inability to model a single-payload monolithic spacecraft

alt - inability to enforce an altitude requirement

lt - inability to set a mission lifetime

Table 6. Results for Comparative Study Cases

Case Assessment Summary Due to the limited availability of module definitions with the publicly available version of RAFTIMATE (BC), only the supplied demo mission could be assessed, which is hard-coded in RAFTIMATE (BC). Given the demo mission was a low-Earth orbit sensing mission, it most closely resembles cases 1, 3, 5, and 7. The value of spacecraft, as quantified using RAFTIMATE (BC), consists of the two dimensions of benefit score and present lifecycle cost. For the demo mission, the benefit scoring function is the minimum of the number of transmitters (payload of IM modules) and number of sensing payloads (payloads of PM1 and PM2 modules) in the spacecraft. Here, IM and PM designate information and payload sensing modules respectively. As a result, the present benefit score is bounded between zero and two and this present benefit can be "monetized" using a simple, but unsubstantiated transfer function.

The results from running cases 1, 3, 5, and 7 using RAFTIMATE (BC) are interesting in their illustration of opposite phenomena from the results obtained from the other tools. In comparing the five-year missions (cases 1 and 3), the monolith (case 1), provides a higher benefit (by approximately 0.3) but at a higher cost (by about 200 FY08\$M) than the fractionated spacecraft. In comparing the ten-year missions (cases 5 and 7), the monolith provides a less benefit and at a higher cost than the fractionated spacecraft. Also, it appears that the ten-year missions exhibit a similar average cost and benefit to the five-year missions, but have a lower benefit variance and higher cost variance (though this is not captured by the central tendency metrics shown in Table 6).

5. *Summary of Quantitative Results for Comparative Study Cases*

Table 6 provides a summary of the quantitative results obtained from each of the four tools in their respective assessment of the eight cases given in Table 2. In Table 6: MAU is multi-attribute utility (-), LCC is life-cycle cost (FY08$M), MPC is mean present cost (FY08$M), and MPB is mean present benefit (-). The reason that a particular case could not be run for a given tool is noted in brackets in Table 6. The symbols tc, mono, alt, and lt represent an inability to model a telecommunications payload, model a single-payload monolithic spacecraft, enforce an altitude requirement, and set the mission lifetime, respectively. As summarily evident in Table 6, no tool successfully analyzed all eight comparative study cases.

For the remote sensing mission, SVM Tool (LM) suggests that the monolith (case 1) is most valuable, whereas PIVOT (OSC) suggests that the fractionated spacecraft in case 3 is the most valuable. Conversely, for this mission, SVMTool (NG) and RAFTIMATE (BC) present a tradeoff between benefit and cost where depending on the stakeholder's preference structure, either case 1, 3, 5, or 7 is the most valuable architecture for the remote sensing mission. And for the telecommunications mission, SVM Tool (LM), SVMTool (NG), and RAFTIMATE (BC) offer no quantitative opinion as to the most valuable architecture, however PIVOT (OSC) suggests that the fractionated spacecraft architecture embodied by cases 6 and 8 are equally valuable but this conclusion is made without knowing the value of a comparable monolith for this mission.

Immediately from the Table 6 one recognizes the difficulty in comparing case value propositions amongst the four VCDM tools, as they each have their own unique method for quantifying value. Further difficulties in tool comparison arise when one considers the inherent assumptions and alternative value statistical metrics such as standard deviation, or a similar measure of variability such as percentiles and skewness. Although not discussed here, these parameters introduce additional dimensions of comparison, exacerbating the already onerous task of ascertaining the most valuable architecture for a given spacecraft mission using the four tools. Therefore, Table 6 ultimately substantiates that when considering comparable monolithic and fractionated spacecraft, from each mission perspective, there is no consensus amongst the tools with respect to most valuable architecture.

D. Conclusion: Comparative Study

The Phase I VCDM tool comparative study yielded several important insights with regard to the characteristics of the publicly available VCDM tools, as delivered at the end of Phase I. These insights form the most succinct list of lessons learned by the individuals contributing to this research effort.

1. *Insights: VCDM Tool Characteristics*

1. *Tool "readiness" of publicly released versions of the Phase 1 VCDM Tools varied* - RAFTIMATE (BC) and PIVOT (OSC) are demo-oriented and thereby not capable of running custom scenarios such that architectures and missions other than those hard-coded in the tool can be assessed. In contrast, the SVM Tool (LM) and SVMTool (NG) tools are readily capable to handle such custom scenarios, in their respective mission domain.

2. *Sufficient technical documentation does not accompany any of the tools* - None of the VCDM tools provided to the public are accompanied by documentation sufficient for understanding the capability of the tool such that the user attains a level of confidence in configuring and operating the tool per their specific objectives.

3. *User Interfaces* - RAFTIMATE (BC) and PIVOT (OSC) had a command-based user "interface," which inhibited easy access to, and configuration of, the tool inputs and resulting outputs. SVM Tool (LM) tool provided slightly easier interaction with the tool inputs and outputs via command line prompts. Conversely, SVMTool (NG) tool provided the most user-friendly GUI via a Microsoft Excel platform, which allowed for ready use of the tool as well as access to all the inputs and outputs of the tool.

4. *Database and User Inputs* - All of the tools relied on populating databases for use in designing and sizing a spacecraft, which contained (COTS) buses, payloads, etc. Concurrently, all tools also had

architectural design inputs defining architecture- and mission-related parameters that needed to be specified for every single spacecraft assessment. Similarities in the number of database and user inputs is not as important as the realization that for all of the tools there is a steep learning curve for understanding and using the input structure. Additionally, amongst the tools, the inputs varied significantly to the point where inputs simply could not be readily compared between any two tools. This made it impossible to keep a given spacecraft or mission assessment with the four tools objective on the basis of a set of common inputs. Furthermore, comparison of the tools is exacerbated by vast differences in the interpretation and modeling of enabling technologies. Regardless, all tools were common in their requirement of an appreciable amount of time (15-30 minutes) to fully specify all inputs in the tool.

5. *Enabling Technologies* - SVM Tool (LM) had by far had the most extensive enabling technology assessment capability, modeling all three types of technologies: inter-module communication, relative navigation, and wireless power transmission. Moreover, SVM Tool (LM) allowed for the use of multiple approaches for achieving wireless power transmission in fractionated spacecraft, a notable value proposition exploration capability. Conversely, RAFTIMATE (BC) and SVMTool (NG) only modeled inter-module communication and unlike these two tools, PIVOT (OSC) modeled inter-module communications but the option to use this technology could not be disabled.

6. *Value Quantification* - The four VCDM tools' quantification of value was indicative of the vastly differing perceptions of value by the respective developers of the tools. SVM Tool (LM) and PIVOT (OSC) perceived and quantified spacecraft value as net present value, or discounted cash flow (revenue less cost). Conversely, RAFTIMATE (BC) and SVMTool (NG) perceived and quantified spacecraft value as the ordered pair of expected utility and lifecycle cost. However, RAFTIMATE (BC) did allow for the utility of a spacecraft to be transformed into monetary value via an unsubstantiated transfer function.

2. Insights: Case Assessments

In terms of the eight comparative study cases used to test the spacecraft assessment capability of the four VCDM tools, many of the objectives of the study were not achieved despite the authors best efforts. In particular, no single tool could even run all eight cases considered in the study. The subsequent outputs of the tools could not be readily compared, or compared at all, given severe disparity amongst each of tools respective quantification of spacecraft value propositions and implicit assumptions. Furthermore, the inputs could not be standardized across the tools, which exacerbated the inability to appropriately compare the outputs of the tools. And many other comparison incompatibilities existed amongst the tools due to a widely varying model fidelity within each of the tools and inclusion/exclusion of enabling technologies. Thus, prescriptions as to the most valuable spacecraft architecture could not be formulated by consensus using the results from the four VCDM tools. It is suspected that the variety value propositions and ensuing prescriptions as to the most valuable fractionated spacecraft was, in part, a consequence of DARPA not imposing adequate standards to guide the value modeling effort in Phase I of the System F6 Program such as use of enabling technologies and required trade space exploration capability. Thus without appropriate guidelines, the four independently developed VCDM tools, were very different, although perhaps the vast disparities amongst the tools was intentional to gain competing perspectives of spacecraft architecture value propositions.

3. Insights: Future VCDM Tool Development

1. *VCDM tool composition*

    (a) *Employ a hierarchical spacecraft architecture characterization* - A hierarchical architecture composition refers to uniformly decomposing spacecraft based on varying levels of system detail (e.g., spacecraft, module, subsystem, and component). Such decomposition allows for the most efficient specification and tracking of architectures from inputs, through simulation, to outputs. Matlab data structures provide a clean way for providing such hierarchical data representations.

    (b) *A bus and payload database is helpful for designing and sizing a given architecture based on real-world systems* – Employing a database wherein COTS (or custom) spacecraft buses and payloads can be stored and then used in a given architecture was found to be highly useful and practical.

The best manner to achieve this is using Excel to store the bus and payload databases and then have Matlab call the databases as needed given the current simulation and architecture.

(c) *Employ a centralized, non-redundant data input (structure)* - A centralized input structure is suggested for ease of access to, and configuration of, all tool inputs. This avoids having to specify inputs more than once in a tool, thus allowing for true global variables, and also enables easy tracking of input/output relationships.

(d) *Need a user interface to bridge the "gap" between input dissemination in a simulation and the desired input structure for user-specification* - A command-based user "interface" is not effective and therefore of little practicality to users of a tool having such an interface. Two options are suggested for incorporating a GUI into a tool: (1) Matlab GUI relying on Excel databases where the input/output structure is in Matlab; and (2) Excel GUI with embedded databases that controls Matlab where the input/output structure is in Excel. No tool incorporated the former and SVMTool (NG) incorporated the latter.

2. *Need for upfront definition of desired model functionality* - Defining the desired functionality of a VCDM tool is necessary to control its respective development and, in particular ensuring that the model fidelity is kept uniform relative to that level functionality. For example, is the tool a point design or trade space exploration tool. This activity is important regardless if one is prescribing the development of one or numerous tools to serve the same overarching purpose.

3. *Further incorporation of mission valuation capabilities* - To complement the current assessment of technical risk in the VCDM tools, it is desirable to continue to mature the tool's value-based risk management capability from a technical, programmatic, and operational perspective. As such, it is desirable for VCDM tools to continually mature to address risk management throughout the entirety of a program, from conception to end-of-life.

III. Multi-Disciplinary Objective Optimization of PIVOT

The intrinsic contribution from the previous comparative study of the four System F6 Program, Phase I VCDM tools is facilitating the future development of these, and other value-centric tools to aid in the selection of the most valuable architecture amongst a set of candidate architectures. To this end, the motivation for the multi-disciplinary and objective optimization (MDOO) of PIVOT also embodies this ideal, in particular because optimization routines, in some capacity, are often a crucial element of any system design tool requiring (extensive) trades, and the nuanced trades between monolithic and fractionated spacecraft value propositions are certainly no exception. Thus, the specific motivation for optimizing PIVOT (OSC) is threefold: first, a working example of optimizing value-centric design tools is provided; second, lessons learned from the optimization are provided, generalizable to any tool having a non-deterministic objective function; and third, the optimization informs the contributions from the first facet of this research, namely, through the provision of additional best practices for the (future) development of value-centric design tools relying on endogenous or exogenous optimization routines.

The optimization uses three complementary MDOO algorithms: genetic algorithm, simulated annealing, and full-factorial search. The version of PIVOT used for this particular MDOO is PIVOT, Build 1; note, this version of PIVOT is not the one made publicly available and is instead one of the earliest versions of an improved PIVOT from OSC per the value modeling effort in Phase II of the System F6 Program. PIVOT, Build 1 is a point design tool, not a trade space exploration tool, intended to model a specific demonstration (demo) mission for DARPA. This demo mission considers one fractionated spacecraft with four modules launched on one vehicle and, in terms of enabling technologies for fractionated spacecraft, only inter-module communication is employed. Given these architectural and mission constraints, the application of MDOO to PIVOT, Build 1 seeks to explore nuances in the mission for this particular demo mission and fractionated spacecraft since broader architectural trades cannot be explored using the tool.

The MDOO of PIVOT begins with a convergence study to assess the magnitude of variability in the objective functions. For a given number of MCA trials, the mean cost and revenue were calculated through numerous simulations and use to produce a cumulative distribution function (CDF) of the deviation of mean cost and revenue from the average values over all simulations, which is taken as the true value. Figure 2 shows that the mean cost can deviate from the true value by approximately $\pm 2\%$ at 100 MCA trials, while at 10,000 MCA trials, the deviation decreases to ±0.1%. The tradeoff, however, is in the time it takes for a single function evaluation and its inherent accuracy, which is shown in brackets in the legend, namely, 20 seconds for 100 MCA trials and increasing to 3 hours for 10,000 trials. Similar observations can be made for the mean revenue.

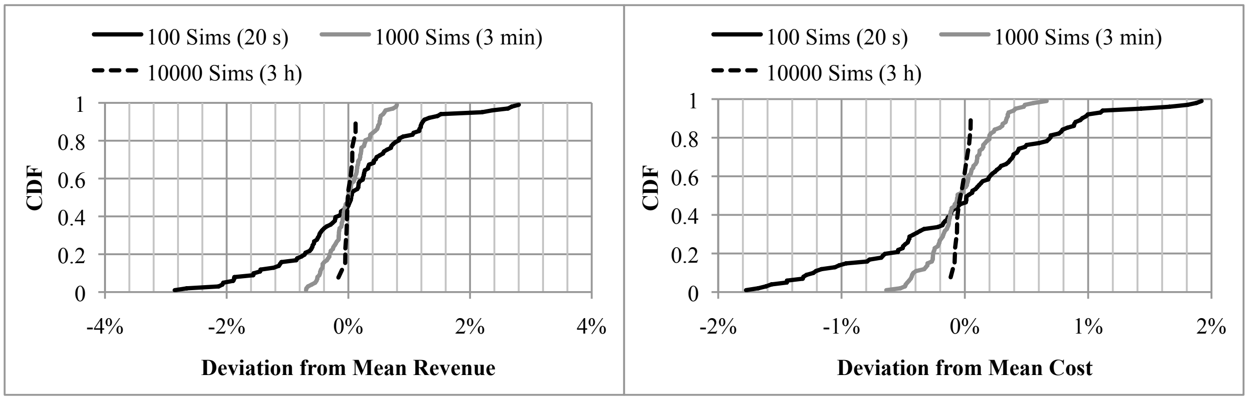

Figure 2. Convergence Study Results

A number of constraints were imposed on the authors by the PIVOT developers, ultimately inhibiting the ability to verify the accuracy, or rectify the inaccuracy of PIVOT. The first notable constraint was that the authors were not allowed to alter the PIVOT code in any manner, primarily for reasons of process (code development) control. The remaining constraints result from improper (i.e., insufficient) documentation of assumptions inherent in PIVOT and also an inability to explore inputs and outputs beyond those desired by the developers of PIVOT. Furthermore, the publicly released version of PIVOT is "scrubbed" of any proprietary data, relying largely on publicly available, generalized relationships, and thus the user is left to make their own assumptions about the accuracy of the results generated by PIVOT.

A. Problem Formulation

The application to MDOO to PIVOT is multi-disciplinary because PIVOT must model numerous spacecraft subsystems, including: communication and data handling, attitude and control, power systems, mechanics, data processing and storage, and harnesses and flight software. Furthermore, the MDOO is multi-objective given that two, competing objective functions, revenue and cost, are an active trade relative to the utopia point namely, zero cost and infinite revenue. Hence, the optimization of PIVOT can be characterized as a MDOO problem.

The MDOO problem formulation for optimizing PIVOT is mathematically represented in Equation (1) where $\boldsymbol{J}$ is the objective function vector, $\boldsymbol{x}$ is the design vector, $\boldsymbol{g}$ is the inequality constraint vector, and $\boldsymbol{h}$ is the equality constraint vector.

$$\begin{aligned} \min \quad & \boldsymbol{J}(\boldsymbol{x},\boldsymbol{p}) \\ s.t. \quad & \boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}) \leq \boldsymbol{0} \\ & \boldsymbol{h}(\boldsymbol{x},\boldsymbol{p}) = \boldsymbol{0} \\ & x_{i,LB} \leq x_i \leq x_{i,UB} \qquad (i = 1, ..., n) \\ where \quad & \boldsymbol{J} = \begin{bmatrix} J_1(\boldsymbol{x}) & ... & J_z(\boldsymbol{x}) \end{bmatrix}^T \\ & \boldsymbol{x} = \begin{bmatrix} x_1 & ... & x_n \end{bmatrix}^T \\ & \boldsymbol{g} = \begin{bmatrix} g_1(\boldsymbol{x}) & ... & g_{m_1}(\boldsymbol{x}) \end{bmatrix}^T \\ & \boldsymbol{h} = \begin{bmatrix} h_1(\boldsymbol{x}) & ... & h_{m_2}(\boldsymbol{x}) \end{bmatrix}^T \end{aligned} \tag{1}$$

As mentioned previously, the optimization was conducted with two, competing objectives: minimization of the program total discounted cost (cost) and maximization of the program total discounted revenue (revenue). Since PIVOT is a stochastic model that simulates the lifecycle of a fractionated spacecraft by subjecting it to random uncertainties such as launch failures, the objective function values are averaged over numerous MCA trials to yield the mean program cost and mean program revenue $\begin{bmatrix} J_1(\boldsymbol{x}) & J_2(\boldsymbol{x}) \end{bmatrix}$.

Identification of the key variables for this particular MDOO problem was achieved through a design of experiments. This exercise demonstrated 'orbit altitude' and 'module lifetime' were dominant with respect to changes in the objective functions and hence these were selected as the two design variables $\begin{bmatrix} x_1 & x_2 \end{bmatrix}$. Module lifetime is the expected design life of each module in a spacecraft; this is independent of the mission (i.e., intended operational) lifetime of the spacecraft. Since the PIVOT demo mission is constrained to Low Earth Orbit (LEO), the altitude was constrained from 300-800 km. Module lifetime was bounded between 2 years and 10 years because PIVOT has a hard-coded design horizon (maximum) of 10 years.

The two objectives are mutually competing because as mission lifetime increases, cost does as well, but so does the opportunity to generate more revenue. The utopia region is therefore one of high revenue and low cost and it was initially expected that a positive correlation between cost and revenue would exist due to the physics-based, cost, and revenue model in PIVOT.

B. Optimization Algorithm and Results

1. Algorithm Selection

Willcox and de Weck provide a rough method for optimization algorithm selection.[10] Given that this optimization problem contains a nonlinear objective function with implicit model constraints and a continuous, real design vector, sequential quadratic programming (SQP) was chosen as an appropriate gradient-based optimization algorithm. On the other hand, Willcox and de Weck state that economic metrics such as NPV, cost, and revenue tend to have sharp "cliffs" (i.e., steep gradients) when visualized in the objective space which could present problems for gradient-based methods. Consequently, these objective spaces often contain many local optima, which inhibit the effectiveness of gradient-based optimizers. Therefore, two heuristic methods, simulated annealing (SA) and genetic algorithm (GA), were chosen as the preferred methods to perform the optimization and a third method, full-factorial search, was used to verify the accuracy of the heuristic optimizations.

2. Simulated Annealing (SA) Algorithm

The first heuristic optimization algorithm used for the MDOO problem given in Equation (1) was simulated annealing. The multi-objective function for the SA optimization was an equally weighted summation of discounted revenue and cost, thereby yielding net present value (NPV).

TUNING The simulated annealing control (input) parameters were iteratively adapted to finesse the candidate optimal solutions leading to the eventual attainment of the optimal solution. Specifically, for PIVOT and this particular optimization problem, an exponential cooling schedule was used with the initial temperature set to be sufficiently high to ensure an initial "melted" state of the objective function. For the optimization, a temperature step change of 0.2 was found to be effective. The number of rearrangements attempted to reach equilibrium at a given temperature was 20.

A custom perturbation function was developed. The function calculates the initial design vector, $\boldsymbol{x}$, via Equation (2).

$$\boldsymbol{x} = (\boldsymbol{x}_u - \boldsymbol{x}_l) * \mathbf{rand} + \boldsymbol{x}_l \tag{2}$$

In Equation (2), $\boldsymbol{x_u}$ and $\boldsymbol{x_l}$ are the vectors bounding $\boldsymbol{x}$ from above and below, and **rand** is a randomly generated number between 0 and 1. It was found that this perturbation function worked well for this MDOO problem.

RESULTS Given the aforementioned SA tuning, an optimal solution was found at an altitude of 577 km and a module lifetime of 2.56 years that yielded the following results:

- NPV = -46.0 FY08\$M
- Cost = 66.3 FY08\$M

- Revenue = 20.3 FY08$M

Pareto Front Estimation The SA algorithm was deemed unsuitable for use in a Pareto Front estimation. In particular, this is because successfully executing a SA optimization relies heavily on a user-in-the-loop. Consequently, based on the authors' experience, it takes 10+ iterations using SA to "tweak" the SA parameters (such as the heating schedule, stop criterion, etc.) so that a global optimum is found. Thus, the SA cannot be readily used in automated Pareto Front estimation algorithms such as Normal Boundary Intersection (NBI),[10] which rely on autonomous execution of the SA algorithm. To this end, an attempt to embed SA in a NBI algorithm was made but the Pareto Front points determined by the algorithm were all easily improved upon by single, iterative executions of SA, thus confirming the avoidance of SA to approximate the Pareto Front using automated Pareto methods.

3. *Genetic Algorithm (GA)*

Tuning For the GA, a population of 50 was used in conjunction with a maximum generation limit of 20 and 100 MCA trials in order to limit the calculation time. Because the objective space contains many local optima, a relatively high mutation rate of 0.1 was used to ensure the optimizer would not be trapped in a local optimum.

Results Given the aforementioned GA tuning, an optimal solution was found at an altitude of 590 km and a module lifetime of 2 years that yielded the following results:

- NPV = -45.3 FY08$M
- Cost = 65.4 FY08$M
- Revenue = 20.1 FY08$M

The two optima found using SA and GA agree given the range of respective uncertainty bounds for revenue and cost at 100 MCA trials, as demonstrated in the convergence study summarized in Figure 2.

Pareto Front Estimation The NBI method with an embedded GA was used to estimate the Pareto front. However, due to the stochastic nature of the objective function evaluation, it was difficult for the modified NBI algorithm to determine the anchor points and search along the normal-utopia line. Subsequently, the NBI-GA algorithm executed for 36 hours without converging and was consequently discarded in favor of a full-factorial Pareto front estimation method.

4. *Full-Factorial Enumeration Algorithm*

Given the aforementioned difficulties associated with estimating the Pareto Front using SA or GA, a full-factorial enumeration of the solution space was considered a viable option. The full-factorial option was also pursued because it allows for verification of the true global optimum for this particular MDOO problem. The caveat to this approach for finding the optimum of the objective functions is that it is very resource intensive, for an equivalent amount of MCA trials as used for the heuristic methods. The full-factorial enumeration required approximately one week whereas the heuristic methods took a few hours. To this end, the results from the full-factorial search will be demonstrated to verify the accuracy of the heuristic optimization methods for this problem. This observation suggests that heuristic methods are the best optimization algorithms for this particular problem. Nonetheless, the full-factorial enumeration provides some benefits for this particular problem, namely, it is the only method for approximating the Pareto Front and it also refutes the validity of gradient-based optimization methods for this optimization problem.

Tuning The tuning for the full-factorial enumeration includes setting the bounds and associated step-sizes (i.e., fidelity) for the full-factorial enumeration as well as the number of MCA trials to be used. For orbit altitude and module lifetime, the bounds and associated step-sizes are as follows: orbit altitude between 350 and 650 km with a 5 km step size and module lifetime between 2 and 10 years with a 0.1 year step size, thus leading to the creation (sampling) of 4,941 solution points in the space. For consistency, a total of 100 MCA trials were used at each of the sampling points. (Note, that the full-factorial solution space, assumes a slightly smaller altitude range than is given in the MDOO problem formulation by Equation (1) because of a logic statement in PIVOT discovered after executing the SA and GA algorithms that restricts revenue from being generated outside of the orbit altitude range of 400-600 km for LEO missions. The reason for this is not documented by the tool developers.)

Results Given the aforementioned full-factorial enumeration tuning, an optimal solution was found at an altitude of 600 km and a module lifetime of 2 years that yielded the following results:

- NPV = -43.9 FY08$M
- Cost = 64.7 FY08$M
- Revenue = 20.8 FY08$M

Thus the SA and GA optimum is statistically identical to the full-factorial optimum, allowing for uncertainty in revenue and cost values for 100 MCA trials, as demonstrated in the convergence study.

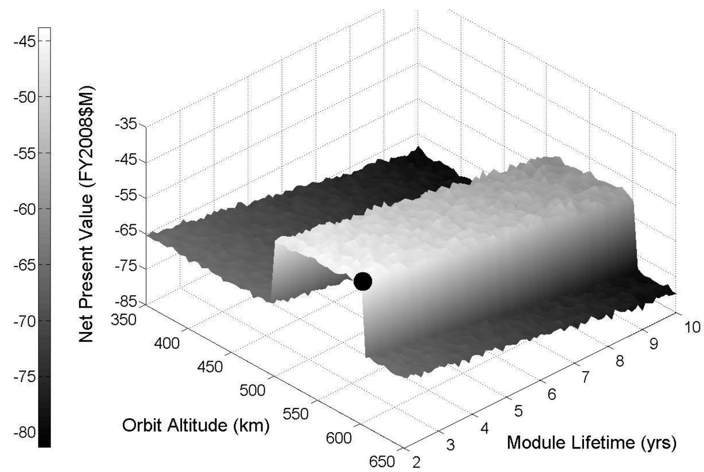

Figure 3. Solution Space via Full-Factorial Enumeration

Pareto Front Estimation For tractability, the solution space for this optimization problem is shown in Figure 3 combines the cost and revenue (equally weighted) into one function, NPV. The single highlighted point in the solution space represents the optimal solution. A Pareto filter was coded to filter out the weakly and strongly dominated designs in the solution space, thus leaving only those solutions that are non-dominated (i.e., the Pareto Set). The estimated Pareto Front is shown in Figure 4 where the joined diamonds represent the Pareto Front and the circles represent the weakly or fully dominated solutions. It is evident from the Pareto Front in Figure 4 that a few candidate architectures dominate the entire solution space represented in Figure 3. Furthermore, the Pareto Front demonstrates that a less expensive spacecraft can readily achieve nearly as high a revenue as much more expensive spacecraft, relatively speaking, thus making the case for the two objective functions, cost and revenue, to actually be far less competing than originally expected.

C. Conclusion: Optimization

The optimization problem considered in this project led to several observed lessons. However, given that PIVOT is still under development and has drastically changed from the version of the tool used for the optimization, physics-based prescriptions as to the best design based on the aforementioned MDOO exercise are not formulated. Instead the lessons learned, summarized hereafter, are generalizable and thereby broadly prescriptive because of their focus on insights ascertained from the optimization exercise, independent of a specific modeling tool.

The first of these lessons is that it became apparent that optimizing an objective function that is stochastic in nature is inherently more difficult than one that is deterministic. Conceptually, one can imagine randomness causing the objective space to morph during optimization, effectively altering the landscape while the optimizer tries to determine the location of the highest peak or deepest valley. Operationally, this manifested itself as the inability for the optimizer to converge, bouncing between different solutions that have perpetually changing objective function values. In the context of value-centric design methodology (VCDM) tools, this presents some challenges for determining the optimum spacecraft architecture, as spacecraft characteristics such as flexibility, robustness, and value often require the use of random processes to properly quantify their influence. Some solutions exist to mitigate this challenge. In rare cases, if the stochasticity can be completely described by a small subset of analytical distributions, it is possible to perform a statistical spacecraft assessment without resorting to sampling of random numbers. This would allow a tool to contain probabilistic elements while the objective function would output a consistent value for a given set of inputs. However, as noted, this situation does not generally occur in practice. An alternative solution is "seeding" random number generators, if applicable. Essentially, this involves pre-defining a set of random number generators, which do not change and are used in the objective function evaluation, permitting reproducibility of the objective function value. This forces the objective space to remain constant, allowing the optimizer to work with what it perceives as a deterministic objective function.

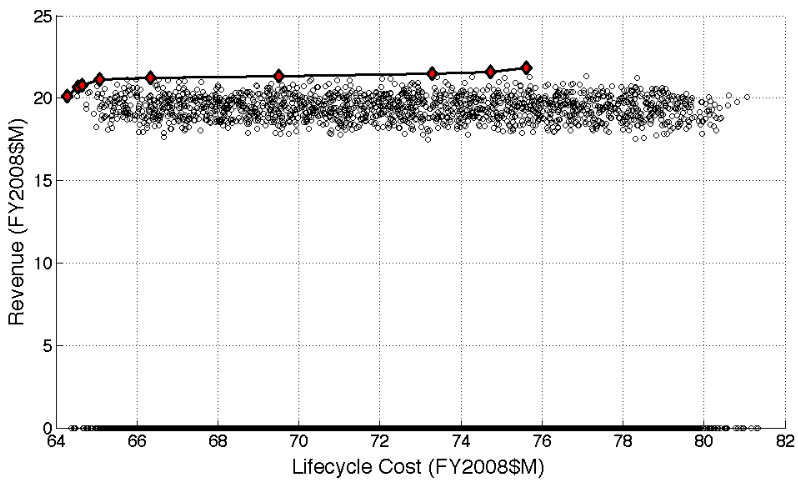

Figure 4. Pareto Font Estimation

Another major difficulty encountered during the optimization was the inability to alter the code for program/version control reasons. As such, execution of optimization algorithms was confined to layering code on and around the core PIVOT algorithm, adapting it to the optimization framework, and extracting the appropriate value metrics. During the course of the optimization project, certain challenges such as the aforementioned morphing objective space were identified but could not be resolved. The ability to modify a model during its respective optimization is vital to the success of the optimization and model development process.

Finally, as is common in corporate and industrial settings, challenges that arise from optimizing a code written by another party, which essentially behaves as a "black box," especially when lacking sufficient technical documentation, were experienced. If the original code developers are not available for consultation, one must spend a significant amount of time examining and internalizing the code to effectively tune the optimization algorithms and correctly interpret the results. It is therefore recommended that a posterior optimizations be performed in situations were full transparency of the tool is had and situations where this is not possible be avoided if possible in order to save time and reduce the possibility of misuse and misinterpretation of an optimization algorithm and its respective results as applied to a particular model. The latter two lessons learned are prescriptive for tool developers who outsource the optimization of their respective tool to a third party or perform the optimization a posterior to the majority of the model development effort at which point changes in the tool are cost prohibitive.

IV. Conclusion

One of the essential outcomes of Phase I of the Defense Advanced Research Projects Agency (DARPA) System F6 (Future, Fast, Flexible, Fractionated, Free-Flying) Program was the development and application of four, value-centric design methodology (VCDM) tools. The intent of these tools was to assess the risk-adjusted, net value of comparable monolithic and fractionated spacecraft. This research effort sought to further the contributions of the four VCDM tools as well as facilitate any future development of these, and other value-centric design tools through two independent investigations, leading to several notable contributions.

The first research investigation entailed an in-depth qualitative and quantitative comparative study of the tools with respect to the tool characteristics (e.g., model architecture, input structure) and recommendations made by each respective tool per a set of eight cases (spacecraft and missions) assessed. The results from this aspect of the research lead to two notable conclusions. First, given the same set of monolithic and fractionated spacecraft architectures and mission to be assessed, all four VCDM tools were not able to reach a consensus as to the most valuable spacecraft architecture. This conclusion was substantiated through a demonstration of significant disparity amongst the four tools in, among other things: modeling fidelity, technology inclusion/exclusion, mission consideration, input and output structures, and fundamentally differing perceptions and quantification of spacecraft value. The second conclusion to be drawn from the comparative study is that the VCDM tools made publicly available are not capable of performing broad (or detailed) trades amongst monolithic and fractionated spacecraft value propositions. This conclusion was substantiated by missions and spacecraft architectures being "hard-coded" in the VCDM tools, a lack of documentation accompanying each VCDM tool, and consequently a steep learning curve for using each tool (even if the user is knowledgeable of spacecraft design). Ultimately, the collection of four DARPA System F6 Program, Phase I VCDM tools, in the capacity they were investigated in this research, were not found to be useful as a value-driven consensus building tool for making decisions about spacecraft architectural and mission strategies to maximize "value."

The second investigation of the research entailed optimizing PIVOT, one of the four VCDM tools from the Phase I of the System F6 Program, using several complementary multi-disciplinary and objective optimization methodologies: genetic algorithm, simulated annealing, and full-factorial enumeration. The most important lesson learned from this facet of the research is that the solution stability of well-vetted multi-disciplinary optimization methodologies is significantly inhibited by nested uncertainty in a tool, often manifested in non-deterministic objective functions, as found in all of the Phase I VCDM tools. These stochastic objective functions exacerbate the difficulty in finding a global optimum as they effectively create a dynamic landscape in which a optimization algorithm must find the optimum solution. Therefore, special care must be taken in balancing the stability of objective function evaluations and the computational resources required to successfully execute an optimization algorithm. This research does offer some solutions to mitigate the issue of objective function instability while retaining nested uncertainty in a tool such as the use of seeding for random number generators used in a modeling tool.

Given the successful execution of the two research investigations, the first intellectual contribution of this research is contended to be an objective review and comparison of the System F6 Program, Phase I VCDM tools, an analysis yet to be done that yields many new insights and contributions from this aspect of Phase I. The second intellectual contribution is the provision of a working example of optimizing a risk-adjusted, net value spacecraft design tool, subsequently leading to prescriptive insights generalizable to the optimization of any tool relying on endogenous or exogenous optimization routines, as often evidenced by non-deterministic objective functions. While the contributions of this research effort are substantiated, further work is warranted to investigate other purported contributions of the System F6 Program value modeling and assessment effort.

References

[1]DARPA Awards Contracts for Fractionated Spacecraft Program (2008). URL `http://www.darpa.mil/news/2008/F6.pdf`.

[2]DARPA Awards Contract for Detailed Design of Fractionated Spacecraft Program (2009). URL `http://www.darpa.mil/news/2009/ F6_NewsRelease_December2009.pdf`.

[3]O'Neill, M. G. *Assessing the Impacts of Fractionation on Pointing-Intensive Spacecraft*. Master's thesis, Massachusetts Institute of Technology (2009).

[4]Dragos, M. *et al.* A Modular, High-Fidelity Tool to Model the Utility of Fractionated Space Systems. *American Institute of Aeronautics and Astronautics: Space 2009 Conference and Exposition* **AIAA 2009-6765** (2009).

[5]Eichenberg-Bicknell, E., Wisniewski, M. J., Choi, S. W. & Westley, D. M. Using a Value-centric Tool Framework to Optimize Lifecycle Cost, Value, and Risk of Spacecraft Architecture. *American Institute of Aeronautics and Astronautics: Space 2009 Conference and Exposition* **AIAA 2009-6765** (2009).

[6]McCormick, D., Barrett, B. & Clapp-Burnside, M. Analyzing Fractionated Spacecraft Architectures using RAFTIMATE – A Boeing Tool for Value-Centric Design. *American Institute of Aeronautics and Astronautics: Space 2009 Conference and Exposition* **AIAA 2009-6767** (2009).

[7]Defense Advanced Research Projects Agency: System F6 (2010). URL `http://www.darpa.mil/tto/programs/systemf6/`.

[8]Brown, O. C., Eremenko, P. & Collopy, P. D. Value-Centric Design Methodologies for Fractionated Spacecraft: Progress Summary from Phase 1 of the DARPA System F6 Program. *American Institute of Aeronautics and Astronautics: Space 2009 Conference and Exposition* **AIAA 2009-6540** (2009).

[9]Nag, S., Yue, H., O'Neill, M. G. & Grogan, P. System F6 Phase 1 Value-Centric Design Methodology Benchmarking and Best Practices: Technical Memorandum (2010). Submitted to Orbital Sciences Corporation.

[10]Willcox, K. & de Weck, O. Class Lecture. Multi-disciplinary Design Optimization. (2010). Massachusetts Institute of Technology, Cambridge, MA.