Exploring the F6 Fractionated Spacecraft Trade Space with GT-FAST

AE 8900 Special Problems Project
in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

Fall 2009

Space Systems Design Laboratory (SSDL)
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia

Author:
Jarret M. Lafleur

Advisor:
Dr. Joseph H. Saleh

December 11, 2009

Two conference papers on this topic are available through AIAA:

Lafleur, J.M. and Saleh, J.H., "GT-FAST: A Point Design Tool for Rapid Fractionated Spacecraft Sizing and Synthesis," AIAA 2009-6563.
Lafleur, J.M. and Saleh, J.H., "Exploring the F6 Fractionated Spacecraft Trade Space with GT-FAST," AIAA 2009-6802.

Exploring the F6 Fractionated Spacecraft Trade Space with GT-FAST

Jarret M. Lafleur
Georgia Institute of Technology, Atlanta, Georgia 30332

Released in July 2007, the Broad Agency Announcement for DARPA's System F6 outlined goals for flight demonstration of an architecture in which the functionality of a traditional monolithic satellite is fulfilled with a fractionated cluster of free-flying, wirelessly interconnected modules. Given the large number of possible architectural options, two challenges facing systems analysis of F6 are (1) the ability to enumerate the many potential candidate fractionated architectures and (2) the ability to analyze and quantify the cost and benefits of each architecture. This paper applies the recently developed Georgia Tech F6 Architecture Synthesis Tool (GT-FAST) to the exploration of the System F6 trade space. GT-FAST is described in detail, after which a combinatorial analysis of the architectural trade space is presented to provide a theoretical contribution applicable to future analyses clearly showing the explosion of the trade space as the number of fractionatable components increases. Several output metrics of interest are defined, and Pareto fronts are used to visualize the trade space. The first set of these Pareto fronts allows direct visualization of one output against another, and the second set presents cost plotted against a Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) score aggregating performance objectives. These techniques allow for the identification of a handful of Pareto-optimal designs from an original pool of over 3,000 potential designs. Conclusions are drawn on salient features of the resulting Pareto fronts, important competing objectives which have been captured, and the potential suitability of a particularly interesting design designated PF0248. A variety of potential avenues for future work are also identified.

Nomenclature

B_N	=	Bell number / size of SEA for N components
$C_{add/replace}$	=	average cost of adding or replacing component
$C_{i,existing}$	=	cost of adding component via an existing module
$C_{i,separate}$	=	cost of adding component via a dedicated module
D_N	=	size of SED for N components
f_{100}	=	smoothing function near 100 W power boundary
f_{500}	=	smoothing function near 500 W power boundary
F_N	=	size of Super-SEA for N components
m	=	number of modules considered
N	=	number of components considered
n	=	number of components in cluster
O_x	=	design objective number x
P	=	total power requirement
S	=	Stirling number of second kind
t	=	time on-orbit
V	=	average orbital velocity
β	=	average ballistic coefficient
ΔV	=	velocity change requirement
ρ	=	average atmospheric density

I. Introduction

IN July 2007, the U.S. Defense Advanced Research Projects Agency (DARPA) released a Broad Agency Announcement soliciting proposals for development of System F6 (Future Fast, Flexible, Fractionated, Free-Flying Spacecraft united by Information eXchange).[1] DARPA's goal for F6 is ultimately a flight demonstration of an architecture in which the functionality of a traditional "monolithic" satellite is fulfilled with a "fractionated" cluster of free-flying, wirelessly interconnected modules. The potential benefits of the F6 approach include enhanced responsiveness in delivering initial capabilities to commercial or government (especially defense) customers, greater flexibility in responding to mid-life changes in requirements, and superior robustness against internal failure and external attack (i.e., enhanced survivability).

Two systems analysis challenges that are especially critical for the flexible and architecturally complex F6 concept are (1) the ability to thoroughly and systematically generate candidate fractionated architectures and, more importantly, (2) the ability to assess and quantify the cost and benefits of each architecture, and in so doing to order-rank the different proposed architectures according to the right metrics. System attributes such as flexibility and survivability, which are essential for systems operating in distinctly uncertain and rapidly changing environments, are not properly captured and valued in the traditional cost- or performance-centric mindsets of system design and acquisition (e.g., achievement of a given level of performance for the least cost, the preferred policy of former Defense Secretary Robert McNamara[2,3]). As a result, a value-centric approach is required to properly assess and benchmark the benefits of fractionation compared with those of the traditional monolith spacecraft. Value-informed decisions regarding F6 architectures hinge upon analysis of uncertainties and value generation throughout the life of the system.

One element necessary in enabling such a probabilistic, value-centric analysis of F6 architectures is a systematic method for enumerating, sizing, and costing the many candidate architectures that are introduced by fractionating subsystems or resources. For example, in one previously published design for F6,[4] twelve instances of six distinct types of fractionatable components are distributed among seven free-flying modules. However, this distribution of components is just one of many possibilities. As this paper will show, if only six components exist in the system and each can be independently placed in any of up to six modules, 203 distinct cluster configurations exist. If the number of components increases to twelve (akin to the design in Ref. 4), the number of possible configurations explodes to over 4.2 million. Furthermore, these numbers do not include the multitude of launch manifesting options.[*] Clearly there is a need to be able to evaluate more than a handful of these alternative configurations in order to make an informed decision on the design of an F6 architecture. The Georgia Tech F6 Architecture Synthesis Tool (GT-FAST) is a point design computer tool designed to help solve this problem by allowing rapid, automated sizing and synthesis of candidate F6 architectures.

This paper is divided into two parts. In Part 1, the sizing procedures and assumptions of the GT-FAST point design tool are detailed, covering the manner in which a GT-FAST point design is specified, the current models for mass, power, and cost, and example outputs including a comparison of GT-FAST outputs against the Jason-2 and TIMED satellite mass, power, and cost budgets. An 8-component fractionated spacecraft example design is used throughout Part 1 to illustrate important concepts and capabilities. In Part 2, GT-FAST is used to conduct a trade study for a 6-component trade space. Covered here are the combinatorial definition and enumeration of the fractionated spacecraft trade space, the definition of several output metrics, and finally visualization and analysis of the 6-component trade space.

Part 1

The GT-FAST Point Design Tool

The primary function of GT-FAST is to convert a user-defined configuration of fractionated components (i.e., a specification of which fractionatable components are assigned to which modules) and launch manifest (i.e., which modules are carried on which launch vehicles) into a point design.[†] The information output by GT-FAST for each point design is a mass, power, and cost budget for the cluster and for each module in the cluster. Also integral to GT-FAST's sizing procedures are user inputs for continuous variables such as orbit altitude, inclination, module design lifetime, and assumptions such as engine specific impulse (I_{sp}), payload mass, and payload power. Because GT-FAST automatically (and relatively quickly) sizes an F6 design, the tool is well-suited for trade studies and has a built-in capability to run a series of input sets and track any number of user-defined output metrics.[‡]

[*] The nomenclature distinguishing components from modules, clusters, and designs is presented in Sec. II.

[†] As a rapid sizing and synthesis point design tool, GT-FAST is similar in concept to numerous others in academia and industry, such as FLOPS[5], ATLAS[6,7], PESST[8,9], EXAMINE[10], and ROSETTA models[11]. GT-FAST is unique in that it is specifically designed for fractionated satellite architectures.

[‡] These input sets are analogous to experiments that the designer might like to run to characterize his design space and determine an optimum design, if one such design exists. If all inputs into GT-FAST were continuous variables, this process would be well-suited to a classical design-of-experiments approach[12].

GT-FAST is currently implemented in Microsoft Excel with approximately 3,200 lines of supporting Visual Basic code. The selection of Excel/Visual Basic as a programming language is due largely to the ability of Excel to automatically iterate among circular references that may exist, a common occurrence in sizing programs. Additionally, this choice allows a great deal of portability in allowing the code to be distributed and used by a large number of engineers in various organizations, if necessary. Computing time depends on the complexity of the design in question and on processor speeds, but in the trade study covered by Part 2 of this paper, computational time was demonstrated at an average of about 20 seconds per point design.

II. Defining a Design in GT-FAST

The first step in any execution of GT-FAST in its point-design mode is the definition of the point design itself. This is accomplished through specification of both discrete and continuous inputs. Because of the size of the combinatoric design space, the discrete inputs have been the focus of GT-FAST F6 analyses and will be covered in the most detail in this paper.

A. Discrete (Fractionation Scheme) Inputs

The principal discrete inputs into GT-FAST deal with specification of which fractionatable components are present in which modules and which modules are carried on which launch vehicles. On this point, it is important to clarify issues of nomenclature. In this paper, the basic unit of fractionation is called a fractionatable component, or a component for short. Depending on the resolution one desires in examining fractionated designs, these components can be subsystems (as in Ref. 13) or resources/payloads (as in Ref. 4). As will shortly be described, the current version of GT-FAST uses the latter as definitions of components.

Next, a compilation of components (and any required essential support subsystems, such as structure, thermal, and others) into a single free-flying vehicle is called a module. A compilation of modules into an independent on-orbit F6 system is called a cluster or architecture. Finally, a cluster with the specification of their launch manifest (e.g., on what vehicle each module is launched, acknowledging that multiple modules may launch on the same launch vehicle) is called a design. This nomenclature is illustrated graphically in Fig. 1.

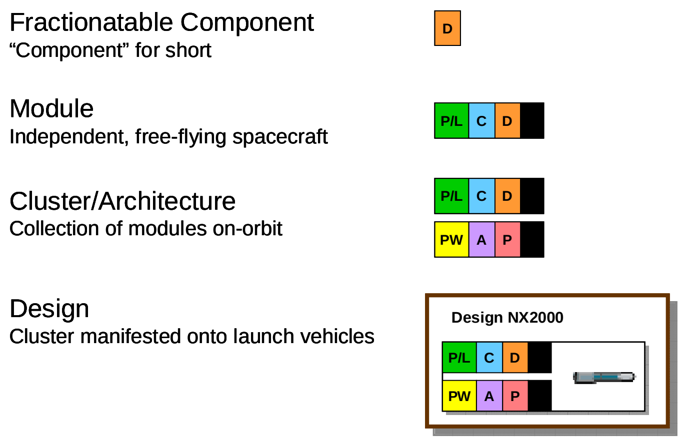

Figure 1. Nomenclature for F6 designs used in this paper.

1. Fractionatable Components currently modeled in GT-FAST

The current implementation of GT-FAST uses five different classes of fractionatable components, consistent with those of Ref. 4. An architecture can contain up to three payloads, up to two 24/7 communication units, up to two high-bandwidth downlinks, a solid-state recorder, and a mission data processor. Icons used in this paper to represent these nine individual fractionatable components are shown in Fig. 2. Payloads are specified by their mass, sunlight and eclipse power requirements, and pointing requirement. Unlike the Air Force Satellite Control Network (AFSCN) communications unit which every module is sized to include, a 24/7 communication unit provides near-continuous communications capability through a relay satellite such as one of the NASA Tracking and Data Relay Satellites (TDRSs). High-bandwidth downlink units allow for high-volume downlinks that could not otherwise be provided with AFSCN or 24/7 links. A solid state recorder allows high-volume data storage, and a mission data processor is a resource allowing for onboard high-speed computing.

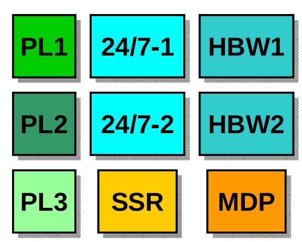

Figure 2. Icons for fractionated components currently implemented in GT-FAST.

2. Example Specification of Fractionation Scheme in GT-FAST

To illustrate the way in which an arbitrary architecture can be input into GT-FAST, in this part of the paper we use the example design shown in Fig. 3. In this design, there are four modules. The first holds Payload #1, the primary solid state recorder, and the primary mission data processor. The second module holds one of two high bandwidth downlink units within the architecture. The third module holds Payload #2 and the second high bandwidth downlink unit, and the fourth module holds Payload #3 and a 24/7 communication unit. Note that there is only one 24/7 communication unit within this architecture even though the current version of GT-FAST can support up to two 24/7 communication units (i.e., in general, the fact that a component is available does not mean that it must be used in a module or a cluster). The black block on each module signifies that all modules also include all essential support subsystems, such as structure, thermal, power, and others.

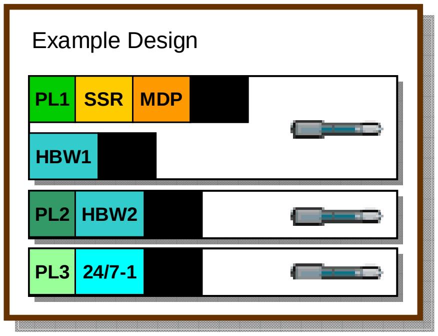

Figure 3. Architectural depiction of example design used in this paper.

Figure 3 also represents that Modules #1 and #2 are manifested to be flown on the same launch vehicle. Modules #3 and #4 each launch separately. Note that launch order is not represented (or needed) by GT-FAST; that is, the representation in Fig. 3 does not preclude Module #4 from launching first or second. Also, as will be discussed, the actual launch vehicle is selected by GT-FAST based on required launch mass, launch vehicle payload capabilities, and launch costs.

The example design shown in Fig. 3 is specified within GT-FAST through two matrices.[§] The first, shown in Fig. 4, maps the fractionatable components (columns) to the modules that carry them (rows). Thus, each row represents the configuration of a single module and is color coded to appear similar to the representation in Fig. 3. Each element of the matrix is allowed to take one of three character values: P, F, or N. The letter “P” indicates that the particular component exists in the design and is present on the corresponding module. The letter “F” indicates that the component exists in the design but is not present on the module. The letter “N” indicates that the component in question does not exist in the design. Thus, any column which is not filled entirely by the letter “N” is allowed to have only one “P” (and all other elements of the column must have the letter “F”).[**] Thus, the first row of the matrix in Fig. 4 shows that Module #1 carries Payload #1, the solid state recorder (SSR), and the mission data processor (MDP), just as indicated by Fig. 3. Note that the column for the second 24/7 communication unit is filled with the letter “N” since the second 24/7 communication unit does not exist in this example design.

The second matrix, shown in Fig. 5, maps the modules (rows) to the launch vehicles that carry them (columns). Thus, each column shows the modules that launch on a given launch vehicle. Each element of the matrix is allowed to take one of two character values: O or N. The letter “O” indicates that a particular module is carried onboard a particular launch vehicle. The letter “N” indicates that a particular module is not carried aboard a particular launch vehicle. Thus, the element in the first row and first column of the matrix in Fig. 5 is marked “O”, indicating that Module #1 is carried by Launch Vehicle #1. By necessity, all other elements in the first row are marked “N”, since Module #1 can only be launched on one vehicle.

[§] Although the matrices in the current implementation of GT-FAST are limited in dimension to 9 × 9, this can be easily modified for future implementations involving more fractionatable components.

[**] It is reasonable to ask why there is a need to distinguish between the “F” and “N” designations since this implementation of GT-FAST focuses on the distribution of payloads and resources (i.e., to size a module, all that is necessary to know is whether a particular component is onboard, regardless of whether it is present on another module. The distinction between “F” and “N” does, however, become useful if the components are subsystems, as in Ref. 13. If we take the case of a fractionated power subsystem through power beaming, for example, we see that an “F” indicates that power is produced in another module and beamed to the module in question, so this module must carry power receiving hardware. An “N”, however, would indicate that no power beaming occurs in the design at all, so the power subsystem could be sized in a more traditional manner. Thus, although the “F” vs. “N” distinction is unimportant in this implementation of GT-FAST using payloads/resources, the nomenclature is retained for future flexibility of the tool.

ARCHITECTURAL									
Number of Modules	4	Consistency Check							
	Payload 1	Payload 2	Payload 3	24/7 Comm 1	24/7 Comm 2	High BW 1	High BW 2	SSR	MDP
Module 1	P	F	F	F	N	F	F	P	P
Module 2	F	F	F	F	N	P	F	F	F
Module 3	F	P	F	F	N	F	P	F	F
Module 4	F	F	P	P	N	F	F	F	F
Module 5	F	F	F	F	N	F	F	F	F
Module 6	F	F	F	F	N	F	F	F	F
Module 7	F	F	F	F	N	F	F	F	F
Module 8	F	F	F	F	N	F	F	F	F
Module 9	F	F	F	F	N	F	F	F	F
	Create this Architecture!								

Figure 4. Input Matrix Mapping of Components to Modules for the Example Design.

LAUNCH									
Number of Launches	3	Launch Consistency Check							
	LV 1	LV 2	LV 3	LV 4	LV 5	LV 6	LV 7	LV 8	LV 9
Module 1	O	N	N	N	N	N	N	N	N
Module 2	O	N	N	N	N	N	N	N	N
Module 3	N	O	N	N	N	N	N	N	N
Module 4	N	N	O	N	N	N	N	N	N
Module 5	N	N	N	N	N	N	N	N	N
Module 6	N	N	N	N	N	N	N	N	N
Module 7	N	N	N	N	N	N	N	N	N
Module 8	N	N	N	N	N	N	N	N	N
Module 9	N	N	N	N	N	N	N	N	N
Tot. Launch Mass (kg)	340.61	148.06	134.49	0.00	0.00	0.00	0.00	0.00	0.00
Launch Cost ($FY08M)	22.00	22.00	22.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 5. Input Launch Manifest Matrix for the Example Design.

An additional note to make about Fig. 4 and Fig. 5 is that, prior to any execution of the GT-FAST sizing program, a series of consistency checks are performed on both of the two matrices to ensure that no implicit constraints are violated. For the matrix in Fig. 4, this involves verifying that the following conditions hold:

- The number of modules input above the matrix (four in Fig. 4) agrees with the number of non-blank rows (modules) within the matrix.
- Components are assigned to modules sequentially starting with Module #1 (i.e., if any rows are left blank, they are at the bottom of the matrix).
- If any column is not filled by N's, then there must be exactly one element in that column marked with the letter "P" (and all other elements in the column must contain the letter "F").
- An SSR and MDP must be present in any design to provide data storage and processing capability; thus, the last two columns in the Fig. 4 matrix cannot contain any N's.
- At least one high bandwidth downlink unit must be present in any design; thus, at least one of the two high bandwidth downlink columns in Fig. 4 must not contain N's.
- At least one 24/7 communication unit must be present in any design; thus, at least one of the two 24/7 communication columns in Fig. 4 must not contain N's.
- At least one payload must be present in any design; thus, at least one of the three payload columns in Fig. 4 must not contain N's.

For the launch manifest matrix in Fig. 5, the consistency check is somewhat simpler. This check involves verifying that the following conditions hold:

- The number of launches input above the matrix (three in Fig. 5) agrees with the number of non-blank rows (modules) within the matrix.
- Modules are assigned to launch vehicles sequentially starting with Module #1 (i.e., if any rows are left blank, they are at the bottom of the matrix).

- No modules may be assigned to multiple launch vehicles; thus, a maximum of one letter "O" may exist per row in Fig. 5.
- All existing modules defined in Fig. 4 are assigned to a launch vehicle in Fig. 5. If four modules are described in Fig. 4, then all four must be manifested in Fig. 5.

In concluding this discussion of the discrete fractionation scheme input into GT-FAST, it is important to note that the example used in Fig. 3 is just one of many possible fractionation schemes that an F6 design could take. As Part 2 of this paper will show, the combinatorics involved in placing components into modules and modules into launch vehicles results in the fact that the possible designs for this problem actually number in the millions. A clear advantage of a tool like GT-FAST is that, when automated, it can allow for a rapid sizing, synthesis, and trade-space evaluation even for large numbers of possible designs. Topics related to such a trade-space evaluation are covered in Part 2.

B. Continuous Inputs

In addition to the discrete inputs involving fractionation scheme, several inputs to GT-FAST are directly controllable from the main input sheet (additional continuous-variable parameters are documented as assumptions in Sec. III and can also be changed if necessary by modifying the models used). These inputs can be grouped into the three broad categories of orbit, payload, and margin.

In terms of orbit-related inputs, the GT-FAST user must specify the altitude and inclination of the desired orbit for the F6 cluster. The baseline implementation of GT-FAST assumes that the orbit is a circular low-Earth orbit (LEO), although the program has been demonstrated to be adaptable to non-LEO orbits. These altitude and inclination inputs allow GT-FAST to select launch vehicles and to budget propellant for orbit maintenance against atmospheric drag. If a higher-fidelity power subsystem model is used by GT-FAST in the future, this information can also be used to estimate the percentage of an orbit in eclipse (i.e., for battery charging and discharging). The example design used throughout Part 1 of this paper assumes a 370 km altitude and 28.5° inclination.

In addition to orbit altitude and inclination, the estimation of orbit maintenance ΔV requires inputs for mission duration and vehicle ballistic coefficient. As will be documented in Sec. III, the propellant estimation model for GT-FAST also includes attitude control propellant and residuals; any propellant that does not fit one of these three categories can be input by the user as a ΔV. Engine I_{sp} is required to convert all ΔV values to propellant masses. Currently all of these inputs are assumed to be the same for each module, although future versions of GT-FAST may allow for non-homogeneous mission durations, ballistic coefficients, orbital elements, etc. The example design used throughout Part 1 of this paper assumes a 2-year mission duration, 110 kg/m² ballistic coefficient (based on average values from Ref. 14), 300 s specific impulse (representative of a bipropellant hypergolic thruster), and no additional user-defined ΔV requirements.

Payload inputs include the mass, power, and pointing requirement for each of the up to three payloads allowed in the current GT-FAST implementation. Power inputs are divided into both sunlit and eclipse requirements, allowing a user to input a low or zero eclipse power requirement, for example, if a payload is a visual imager. Mass and power inputs directly feed into the mass and power budgets for the modules carrying the corresponding payloads. Pointing requirements (coupled with a fourth non-payload pointing requirement which could be used to account for communications antenna pointing, for example) affect attitude determination and control system (ADCS) cost estimates from the Small Satellite Cost Model 2007 (SSCM07).[15] It deserves note that the GT-FAST requirement of only four inputs per payload allows portability in that only minimal information need be passed between payload designers and GT-FAST users. In the example design for this part of the paper, Payload #1 is modeled after the NOAA-N Search and Rescue Repeater (SARR) instrument,[16] Payload #2 is modeled after the transponder payload of the Orbcomm LEO communications satellite,[14,17] and Payload #3 is modeled after the science sensor payloads of the recent Interstellar Boundary Explorer (IBEX) spacecraft.[16] Although this payload set is notional, it highlights the potential for F6 to accommodate a variety of diverse payloads within a single fractionated infrastructure.

Finally, the user may specify four independent margin percentages to be used for dry mass, propellant mass, power, and cost. These margins are added to each of the respective budgets for each module to account for possible growth during the development, production, and operations of the program. Special notes to make are that the cost margin is not applied to the launch vehicle, and the mass margin is not applied to the launch adapter mass. In the example design for this paper, 25% margin is used for dry mass, propellant mass, power, and cost.

Table 1. Assumed Payload Characteristics for Example Design.[14,16,17]

Payload No.	Payload Description	Flight Heritage	Mass (kg)	Sunlit & Eclipse Power Requirement (W)	Pointing Requirement (deg.)
1	Search & Rescue Repeater	NOAA-N	24.0	53	1.0
2	LEO Transponders	Orbcomm	8.4	10	5.0
3	Sensors and Electronics Unit	IBEX	26.0	16	0.5

III. Sizing and Costing Models

At the core of GT-FAST is a set of mass, power, and cost estimating relationships constructed primarily from Refs. 14 and 15 and complemented by estimates from one satellite manufacturer. In this section, we survey the sizing and costing models used by GT-FAST. First, we survey the power and mass models by subsystem. Second, we survey the cost models by line item, including a discussion of launch vehicle selection. Although this section describes the GT-FAST models as currently implemented, it should be kept in mind that these models are modular and can be (and have been) adapted if a user prefers to use a model better suited for a particular application.

A. Mass and Power Modeling

Individual modules are sized to be independent, free-flying spacecraft, allowing for the application of mass and power estimating relationships from sources such as Ref. 14. The mass and power models for the majority of subsystems (propulsion, attitude determination and control, thermal, power, and structures) aboard each module are no different from typical models for conceptual design which will be described next. Depending on the components present on a given module, the communications subsystem and command and data handling (C&DH) subsystem may use modified mass and power models, which will also be described next.

1. Models for Typical Subsystems

Since the only fractionatable components in this implementation of GT-FAST involve communications, data storage, and data processing, the subsystems of propulsion, attitude determination and control, thermal, power, and structures are sized as usual for a first-order conceptual design. In terms of mass, this means that historical percentages[14] are used which relate a subsystem mass to the total dry mass of the module. For example, using historical data for LightSats,[14] the structural mass of a module is expected to be 22.7% of the total dry mass. In the example design of Fig. 3, the resulting dry mass of Module #3 is 88.3 kg (before margin is applied); correspondingly, the structures subsystem mass estimate is 20.0 kg. This method of modeling based on historical percentages also applies to the communications and C&DH subsystems when no high bandwidth downlink or 24/7 communication units are included on a given module.

In terms of power, typical subsystems use a set of power estimation relationships from Ref. 14. These relationships are more complex than the mass percentages and use different models depending on the total power requirement of the module. If the total module power requirement is below 100 W, Ref. 14 recommends a particular fixed power level for each subsystem. If the total power is between 100 W and 500 W, a percentage of the total power is recommended, and if the total power is above 500 W, a different percentage is recommended.

To avoid convergence issues near the 100 W and 500 W boundaries and to provide continuity in the power estimate, a smoothing function is applied to the power model in the vicinity of the boundaries. The smoothing function f below is a third-order polynomial which describes the relative weighting between the two power estimates of the Ref. 14 model in the vicinity of a boundary. At the boundary itself (i.e., 100 W or 500 W), the two estimates are equally weighted and $f = 0.50$. At 20% above the boundary (i.e., 120 W or 600 W), $f = 1$, and at 20% below the boundary (i.e., 80 W or 400 W), $f = 0$. Thus, f describes the weighting on the power estimate above the boundary; as a result, the weighting on the power estimate below the boundary is $1-f$. The polynomials that describe f as a function of total power P are shown in Eq. (1) and plotted in Fig. 6. As an example, Fig. 7 shows the result of smoothing on one representative subsystem (ADCS) power requirement. Note the C^1 (and C^0) continuity[††] provided by the smoothing function as opposed to the original discontinuous model from Ref. 14.

Note that it is assumed that these power relationships apply both to sunlit and eclipse periods; thus if the power requirement of the payload for a given module is also constant between sunlight and eclipse, the sunlight and eclipse

[††] For a brief discussion of simple C^0 continuity and C^1 first-derivative continuity, the reader is referred to Ref. 18.

power requirements are identical. Additionally, it should be noted that this model has no coupling between power and mass estimates (although higher-fidelity, coupled models could easily be implemented in the future).

$$
\begin{aligned}
f_{100} &= -0.00003125P^3 + 0.009375P^2 - 0.90P + 28 \\
f_{500} &= -0.00000025P^3 + 0.000375P^2 - 0.18P + 28
\end{aligned}
\tag{1}
$$

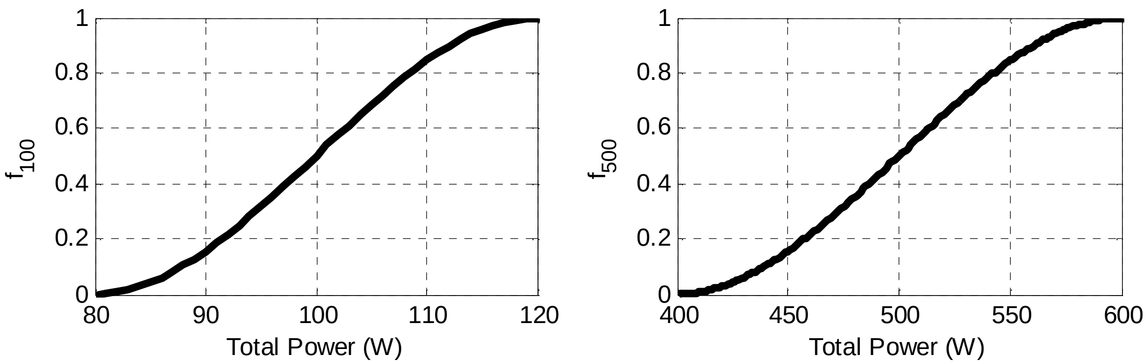

Figure 6. Smoothing functions for the 100 W boundary (left) and 500 W boundary (right).

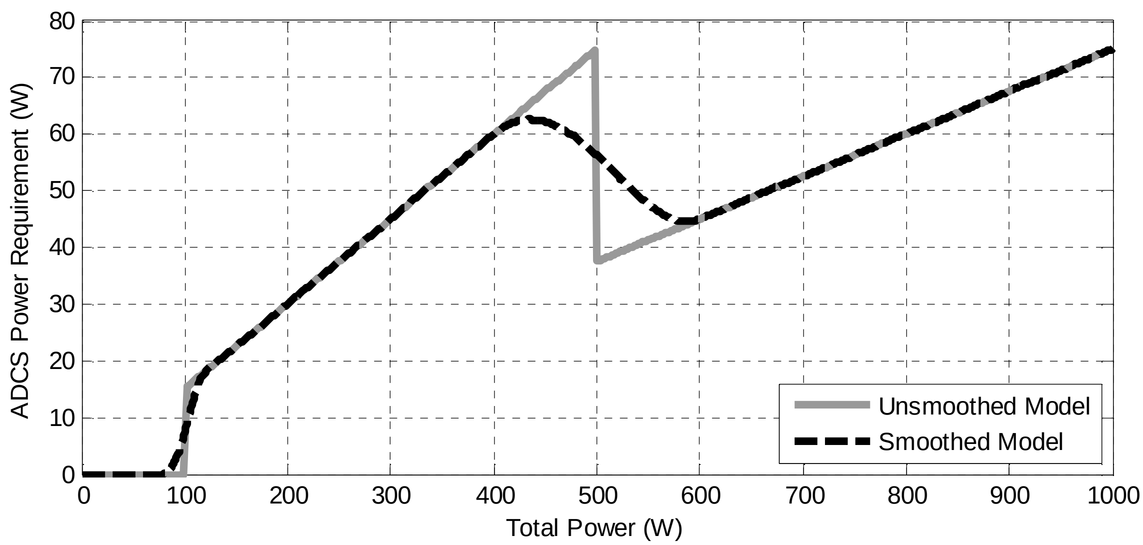

Figure 7. Example (ADCS) variation in power requirement with total spacecraft power.

2. Models for Fractionation-Affected Subsystems

In the current implementation of GT-FAST, the sizing of the communications and C&DH subsystems is directly affected by fractionation, and these cannot be properly sized based on historical data (since no such data exist for fractionated systems). Instead, these subsystems are sized using a set of rules which define the subsystem components that are present on a module given the fractionation scheme.

The components of the communications subsystem for any given module include the high bandwidth downlink and 24/7 communication units allocated to that module as well as an intra-cluster wireless unit and AFSCN link equipment. The intra-cluster wireless unit and AFSCN link equipment are included by default for all modules; the former allows for wireless communications between modules, and the latter allows low-bandwidth communication with an AFSCN-equivalent ground station network. The only exception to the inclusion of the intra-cluster wireless unit on all modules is that the unit is excluded in instances where only one spacecraft exists in the architecture (in which case there is presumably no need for wireless communications between modules). The mass of the communications subsystem is the sum of the masses of each component present. The power requirement of the communications subsystem is based on the assumption that the module always uses the intra-cluster wireless unit and only one external link at a time. Thus, if a module carries a 24/7 communication unit, high bandwidth downlink unit, and AFSCN link equipment, only the largest of these three power requirements is added to the power required by the intra-cluster wireless unit. No distinctions are made between sunlit and eclipse periods, so the power requirements during sunlight and eclipse are equal.

The command and data handling subsystem for any given module consists of the solid state recorders (SSRs) and mission data processors (MDPs) allocated to that module as well as a minimum C&DH unit providing basic processing and storage capabilities. The minimum C&DH unit, which has a mass of 5.5 kg and power requirement of 15.5 W based on Ref. 14, is present on all modules to provide basic functionality even if an SSR or MDP is not present on that module. The mass of the C&DH subsystem is the sum of the masses of each component present, and the power requirement is the sum of the power requirements for each component present. As with the communications subsystem, no distinctions are made between sunlit and eclipse periods.

3. Propellant Mass Estimation

Propellant mass for each module created in GT-FAST is budgeted among the five areas of orbit maintenance, additional maneuvers, attitude control, residuals, and margin. Orbit maintenance propellant is estimated through the expression for ΔV in Eq. (2), which uses a module's mission duration (t), altitude, and estimated ballistic coefficient (β). Altitude is used to estimate velocity (V) assuming a circular orbit, and altitude is also used to estimate atmospheric density (ρ) under the conservative assumption of a solar maximum period using data from Ref. 14. The ΔV for additional maneuvers not associated with any of the other four categories is left as a user input. This ΔV is converted to a propellant mass through the ideal rocket equation. In the example design shown throughout Part 1 of this paper, the ΔV for orbit maintenance is 110.0 m/s per year, totaling to 220.0 m/s overall. The ΔV associated with additional maneuvers is assumed to be zero for the example application.

The remaining three areas of propellant budget are estimated as percentages of propellant mass rather than ΔV values. Attitude control propellant is estimated at 6.5% of the total propellant budget, and residuals are estimated at 1.5% of the total budget according to Ref. 14. Propellant margin is user-defined, and in the example used in Part 1 of this paper, propellant margin is set at 25%. As a result of these three additional propellant requirements, the total propellant budget for each module in the example design is 322.6 m/s.

The propellant mass estimation model described here is applied to all modules in a GT-FAST architecture, although specialized assumptions (for example, different mission durations for different modules) can be applied in future implementations of the tool. In one example of a modified GT-FAST tool, a geosynchronous communications satellite was modeled; modifications to the propellant estimation models involved the addition of geosynchronous-orbit-specific stationkeeping requirements and disposal orbit requirements.

$$\Delta V = \frac{\rho V^2 t}{2\beta} \tag{2}$$

B. Cost Modeling

In terms of cost modeling, GT-FAST in its current implementation for Earth-orbiting F6 designs primarily draws upon the Small Satellite Cost Model 2007 (SSCM07)[15], although other models are used for estimates that SSCM07 does not provide (for example, software, ground segment development, and launch costs). One challenge in using traditional satellite cost models for fractionated architectures is that these models are regressions from previous programs and are inherently biased toward architectures consisting of a single spacecraft. As a result, the regressor variables in the traditional cost estimating relationships (CERs), which are often subsystem masses, refer to properties of a single monolithic spacecraft and not to a spacecraft cluster. Thus, some of these CERs are reasonably applied to properties of individual modules in the cluster, while others are more reasonably applied to properties of the cluster as a whole. For example, it would make little physical sense to apply the propulsion subsystem CER, which uses propulsion subsystem dry mass as the regressor, at the cluster level (which would imply that several small propulsion subsystems on independent, free-flying modules should have the same cost as one large subsystem on a monolithic spacecraft). On the other hand, the program management/systems engineering CER would be more appropriately applied to overall metrics of the entire cluster (i.e., applying this CER on a per-module basis would imply that program management on each vehicle is independent, which would likely be an overestimation). Thus, costs are divided into costs estimated at the module level and costs estimated at the cluster level.[‡‡] In the present implementation, all costs are reported in fiscal year 2008 dollars ($FY08).

1. Module-Level Cost Estimation

GT-FAST accounts for subsystem, payload, and assembly, test, and launch operations (ATLO) costs on a per-module basis. In terms of subsystem costs, SSCM07 is used almost exclusively. For module wet masses below 125 kg, the SSCM07 Micro Satellite CERs are used, and for all other wet masses the Small Satellite CERs are used. Typical inputs into an SSCM07 subsystem cost model include the dry mass of the subsystem and subsystem-specific parameters. For example, the propulsion subsystem cost model requires as inputs the propulsion subsystem dry mass as well as type of propellant (cold gas, hydrazine monopropellant, or hydrazine bipropellant). Within GT-FAST, propulsion subsystem dry mass is known from mass sizing described in Sec. III.A of this paper, and propellant type is automatically inferred from the continuous input of I_{sp} also described in Sec. II. The SSCM07 equation relating these inputs to an estimated cost (which can be broken into recurring and nonrecurring parts) is typically nonlinear. The only deviation from SSCM07 in terms of subsystem CERs is due to the fact that SSCM07 costs the C&DH and communications subsystems as a single unit based on the total mass of the two subsystems; for accounting purposes, this total cost is split such that 58% is assigned to the C&DH subsystem and 42% is assigned to the communications subsystem based on dry mass split data from Ref. 14.

Payload cost is estimated as 40% of the module bus cost based on a CER from Ref. 14. Assembly, test, and launch operations (ATLO) cost is estimated for each module based on SSCM07. By the SSCM07 definition, ATLO consists of the combination of the categories of integration, assembly, and test (IA&T) and launch and orbital operations support (LOOS). ATLO cost estimation inputs include module wet mass (calculated from the mass sizing of Sec. III.A) to determine whether to use the Micro Satellite or Small Satellite CERs, design lifetime (from user inputs), and module power (from the larger of either sunlit or eclipse operating power calculated as in Sec. III.A).

2. Cluster-Level Cost Estimation

GT-FAST accounts for program management/systems engineering (PMSE), flight software development, ground segment development, operations, and launch costs at the cluster level. PMSE cost is estimated using SSCM07 relationships. Inputs to the PMSE cost model are the total cost of all module buses (each calculated on a per-module basis as described in Sec. III.B.1), the total ATLO cost for all modules (also each calculated on a per-module basis), and the total wet and dry masses of all modules (calculated as described in Sec. III.A).

Flight software cost is estimated based on relationships available in Ref. 14 for cost per thousand lines of code. Nominally, it is estimated based on Ref. 14 that each module requires 26,000 lines of flight software code, and GT-FAST scales lines of code directly with the number of modules in the cluster. This is likely to produce a conservative estimate of software cost since, although each module would have unique components aboard, there may be cost savings due to elements of commonality.

Ground segment development cost is also estimated based on Ref. 14. The ground segment development cost includes ground station facilities, equipment, software, logistics, and system-level costs. The breakdown between each of these various components of the cost is given by a set of typical percentages from Ref. 14, and the absolute magnitude of the ground segment development cost is anchored upon a ground software cost estimate under the assumption of 100,000 lines of code from Ref. 14. GT-FAST can also allow the user to override the Ref. 14 ground segment development cost model with a custom estimate (for example, if new ground stations do not need to be developed).

Operations cost is estimated based on a publicly-available mission operations cost model from NASA.[19] Inputs into this first-order model include the investment cost of the system (total development and production cost, excluding launch costs) and duration of the mission (a user input described in Sec. II.B). Normally, use of the NASA model requires specification of mission type; in this implementation of GT-FAST, estimates are produced for both Earth observation and communication mission types and averaged since either (or both) of these missions may be executed by a fractionated design, depending on the payloads carried.

Launch cost estimation is accomplished through a three-step process for each launch in the prescribed manifest (e.g., see Fig. 5). First, the total mass capability required by the launch vehicle is calculated by adding the individual wet masses of each module aboard. In order to account for structural mating of each module to the launch vehicle, an adapter mass of 18.8 kg is added to each individual module mass based on example vehicles in

[‡‡] This modeling strategy can be rigorously refined when actual fractionated spacecraft are flown and cost data becomes available.

Ref. 14. Second, a database of launch vehicle capability relationships[§§] is used to compute the maximum payload deliverable to the user-specified orbit for each of 20 expendable launch vehicles[***]. Third, GT-FAST identifies the launch vehicles with deliverable payload capabilities greater than or equal to the mass to be launched and selects the lowest-cost option based on a launch vehicle cost database compiled from Refs. 21 and 22. This three-step procedure is repeated for each scheduled launch in the manifest. Note that this assumes launch vehicles are purchased independently for each of the launches at the prescribed price (no discounts are assumed, for example, if all launches use the same vehicle). Although launch vehicle reliability does not factor into the computer's automated selection of launch vehicles, GT-FAST does allow the user to exclude launch vehicles from consideration (for example, if reliability is too low to merit consideration).

C. Model Integration

As mentioned in Sec. II, one reason for selecting Microsoft Excel as a platform for GT-FAST was its automatic iteration capability. As a result, circular references among cells can be made and automatically evaluated without explicit programming of iteration procedures. This capability is utilized extensively by GT-FAST. Each module in a given architecture is represented by a power, mass, and cost breakdown in a dedicated worksheet within the tool (see Fig. 8), and formulae within the cells of each worksheet are allowed to reference other values within the worksheet or within other worksheets. Use of Visual Basic code within Microsoft Excel allows worksheets to be automatically created and configured according to the number of modules, components contained within modules, and other inputs specified by the user.

As a result, once a user inputs a fractionation scheme and series of continuous inputs as described in Sec. II, GT-FAST creates a sizing worksheet for each module and automatically iterates both within worksheets and among worksheets in order to determine the mass, power, and cost breakdown for each module and for the entire architecture. It should be noted that most sizing and costing relationships are based on parametric scaling relationships and not discrete unit masses, power requirements, or costs.

Figure 8. Worksheets from GT-FAST for the example design in Part 1 of this paper.

IV. Example Outputs

In this section, examples of GT-FAST outputs are provided. Shown first are the mass, power, and cost budgets for a fully-sized point design (the example design used throughout Part 1 of this paper). Second are results of a partial validation of GT-FAST against two monolithic satellites.

A. Example Point Design

Tables 2-5 show the mass, power, and module-level cost budgets for each of the four modules for the example design used throughout Part 1. Recall that the configuration of this design is shown by Fig. 3, the payloads it carries are defined in Table 1, and it is assumed to be in a 370 km circular orbit at 28.5° inclination for a two-year mission. Table 6 documents the estimated cost budget for the entire system, which includes costs estimated at the module level and cluster level. These mass, power, and cost budgets represent the typical core outputs of GT-FAST.

Figure 9 graphically shows the cost breakdown of Table 6. Note that the GT-FAST cost models here assume a ground segment development cost and margin, which together comprise 33% of this budget; these items are particularly easy to adjust if the user wishes to use custom estimates. As shown in Fig. 10, the launch vehicle selected for all three launches is the Pegasus XL at a cost of $22 million (FY08).[†††] The Pegasus XL's 450 kg payload capacity to the desired orbit was sufficient for all launches, and $22 million was the lowest launch cost in the database used for this launch vehicle selection (foreign and under-development vehicles were excluded).

[§§] For an example of such relationships using response surface equations (RSEs), the reader is referred to Ref. 20.

[***] Currently, GT-FAST's launch vehicle database is limited to American launch vehicles, but this database will be expanded in the future to include foreign options.

[†††] Note that GT-FAST does not require all launches to use the same launch vehicle; this coincidence is due to the particular payload requirements for this set of launches.

Table 2. Mass, Power, and Cost Budgets for Module #1

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	24.0	53.0	53.0	16.0
2.0	Bus Subsystems				
2.1.	Propulsion	4.4	0.1	0.1	6.4
2.2.	Attitude Control	18.4	61.1	61.1	2.9
2.3.	Communications	10.1	95.0	95.0	7.8
2.4.	Command & Data Handling	26.0	70.4	70.4	10.8
2.5.	Thermal	2.8	20.4	20.4	0.5
2.6.	Power	40.0	108.7	108.7	9.2
2.7.	Structures & Mechanisms	36.9	0.0	0.0	2.6
	Pre-Margin Subtotal	*162.4*	*408.7*	*408.7*	*56.2*
	Margin	40.6	102.2	102.2	[See Table 6]
	Post-Margin Subtotal	*203.0*	*510.9*	*510.9*	
3.0	Propellant	23.5			
	Loaded Mass	*226.5*			
4.0	Adapter	18.8			
	Boosted Mass	*245.3*			
	ATLO Cost				5.5

Table 3. Mass, Power, and Cost Budgets for Module #2

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	0.0	0.0	0.0	0.0
2.0	Bus Subsystems				
2.1.	Propulsion	1.8	0.0	0.0	5.4
2.2.	Attitude Control	7.4	58.2	58.2	1.0
2.3.	Communications	18.8	177.6	177.6	0.3
2.4.	Command & Data Handling	5.5	15.5	15.5	0.4
2.5.	Thermal	1.1	19.4	19.4	0.5
2.6.	Power	16.1	117.3	117.3	0.8
2.7.	Structures & Mechanisms	14.9	0.0	0.0	2.8
	Pre-Margin Subtotal	*65.6*	*388.0*	*388.0*	*11.1*
	Margin	16.4	97.0	97.0	[See Table 6]
	Post-Margin Subtotal	*82.1*	*484.9*	*484.9*	
3.0	Propellant	9.5			
	Loaded Mass	*91.6*			
4.0	Adapter	18.8			
	Boosted Mass	*110.4*			
	ATLO Cost				2.0

Table 4. Mass, Power, and Cost Budgets for Module #3

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	8.4	10.0	10.0	4.7
2.0	Bus Subsystems				
2.1.	Propulsion	2.4	0.0	0.0	5.7
2.2.	Attitude Control	10.0	59.5	59.5	1.0
2.3.	Communications	18.8	177.6	177.6	0.3
2.4.	Command & Data Handling	5.5	15.5	15.5	0.4
2.5.	Thermal	1.5	19.8	19.8	0.6
2.6.	Power	21.7	114.0	114.0	0.9
2.7.	Structures & Mechanisms	20.0	0.0	0.0	2.8
	Pre-Margin Subtotal	*88.3*	*396.4*	*396.4*	*16.4*
	Margin	22.1	99.1	99.1	[See Table 6]
	Post-Margin Subtotal	*110.3*	*495.5*	*495.5*	
3.0	Propellant	12.8			
	Loaded Mass	*123.1*			
4.0	Adapter	18.8			
	Boosted Mass	*141.9*			
	ATLO Cost				2.0

Table 5. Mass, Power, and Cost Budgets for Module #4

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	26.0	16.0	16.0	14.8
2.0	Bus Subsystems				
2.1.	Propulsion	3.4	0.0	0.0	6.1
2.2.	Attitude Control	14.3	44.6	44.6	3.1
2.3.	Communications	15.2	95.0	95.0	6.9
2.4.	Command & Data Handling	5.5	15.5	15.5	9.6
2.5.	Thermal	2.1	14.9	14.9	0.4
2.6.	Power	31.0	111.6	111.6	8.6
2.7.	Structures & Mechanisms	28.6	0.0	0.0	2.3
	Pre-Margin Subtotal	*126.2*	*297.6*	*297.6*	*51.8*
	Margin	31.5	74.4	74.4	[See Table 6]
	Post-Margin Subtotal	*157.7*	*372.1*	*372.1*	
3.0	Propellant	18.3			
	Loaded Mass	*176.0*			
4.0	Adapter	18.8			
	Boosted Mass	*194.8*			
	ATLO Cost				4.5

Table 6. Overall Example Design Cost Budget

Cost Element	Cost (FY08$M)
Module-Level Costs	
Module #1	61.6
Module #2	13.1
Module #3	18.4
Module #4	56.3
Program Management	23.4
Software	51.7
Ground Segment Development	75.6
Operations	33.1
Pre-Margin Subtotal	*333.2*
Margin (25%)	83.2
Post-Margin Subtotal	*416.4*
Launch	66.0
Total	**482.4**

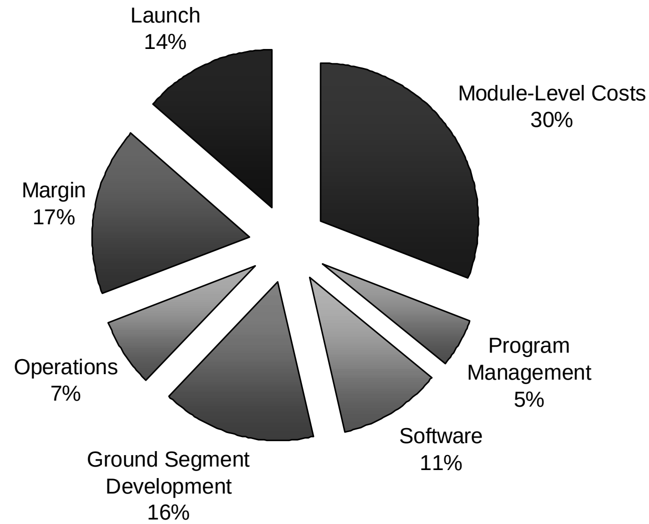

Figure 9. Breakdown of Costs from Table 6.

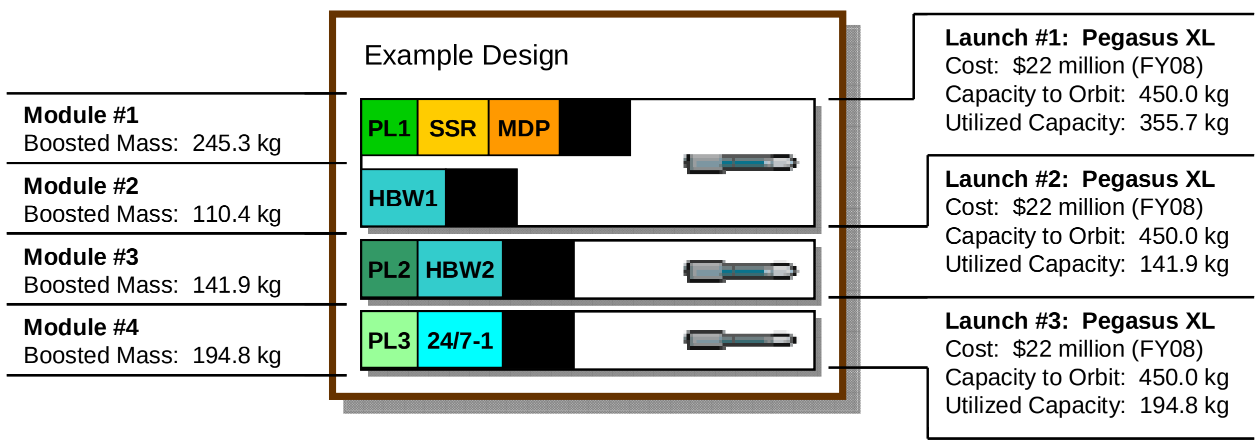

Figure 10. Launch Summary for Example Design.

B. Comparison against Operational Monolithic Spacecraft

While a fractionated spacecraft has yet to launch, it is possible to partially demonstrate the accuracy of GT-FAST in comparison with existing monolithic spacecraft. Used in this comparison are the Jason-2 and TIMED spacecraft, both of which are approximately of the small-satellite class and are currently operational in orbit. Neither spacecraft was used in the generation of the models in Ref. 14 that GT-FAST draws upon for several mass and power estimates. Jason-2 (see Fig. 11) is a follow-on mission to Jason-1, aiming to continue the data record of Jason-1 and measure sea surface levels to a 2.5 cm accuracy[16]. Jason-2 is a cooperative undertaking between NASA, NOAA, CNES, and EUMETSAT, and it has orbited at a circular orbit altitude of 1336 km and inclination of 66° since its launch in June 2008.[16,24] TIMED (see Fig. 12), launched on the same Delta II rocket as Jason-1 in December 2001, operates in a circular 625 km altitude, 74.1° inclination orbit.[16,26] TIMED is sponsored by NASA and was designed, built, and is operated by the Johns Hopkins University

Figure 11. Artist's Concept of Jason-2 Satellite.[23]

Applied Physics Laboratory. Its mission is the global study of the physical and chemical processes acting within Earth's upper atmosphere.[16]

Figure 12. Artist's Concept of TIMED Satellite.[25]

In order to complete this comparison, GT-FAST is used to size a single-module cluster (i.e., a traditional monolithic spacecraft). As a result, the intra-cluster wireless unit (which has no use in a monolithic spacecraft) is automatically excluded. Additionally, the 24/7 communication units and high-bandwidth downlink units are excluded since these do not represent the actual components flown on TIMED or Jason-2.[‡‡‡] Thus, the remaining communication subsystem in GT-FAST consists of an AFSCN-equivalent link. In terms of command and data handling, the SSR and MDP are both included in the GT-FAST model. Propellant estimates assume an orbit maintenance ΔV as given by Eq. (2), although in the cases of TIMED and Jason-2 this number is very small due to the high altitudes of the orbits. For Jason-2, an additional 120 m/s of ΔV is included as indicated by Ref. 24. For the purposes of this comparison, no design margin is included in any budget (i.e., mass, power, propellant, or cost). The remaining inputs into GT-FAST are summarized in Table 7.

Table 8 summarizes the comparison between several actual metrics from the Jason-2 mission and their calculated counterparts in the GT-FAST model. Note that wet and dry masses agree very well (within 2.2% and 3.2%, respectively), and average power also agrees quite well for this first-order model (within 14.9%). A significant discrepancy exists in terms of cost, but this may be partially explained by substantial cost overruns and schedule slippage encountered in the Jason-2 project (to the extent that a major new instrument, the Wide Swath Ocean Altimeter, was entirely descoped in 2005).[27] An earlier 2005 Jason-2 cost estimate of $250 to $300 million[27] is much closer to GT-FAST's estimate of $250 million. Finally, although the actual Jason-2 spacecraft launched on a Delta II 7320-10,[23] GT-FAST selects the smaller and less costly Taurus 2210 launch vehicle. It deserves note that a modified Peacekeeper missile (with a smaller payload capacity than the Delta II) was considered for Jason-2 after an offer from the Department of Defense Space Test Program but was not selected because of certification and risk concerns.[27]

Table 9 is identical in format to Table 8 and summarizes the comparison between actual metrics from the TIMED mission and their calculated counterparts in the GT-FAST model. Again, wet and dry masses agree quite well, and average power is acceptable given this first-order model. In this case, cost is also very accurate (within 8.4%). In this case again, GT-FAST selects the smaller Taurus 2210 instead of the Delta II 7920-10. However, it should be noted that the Taurus 2210 could not have launched both TIMED and Jason-1 (as was done in reality); if 500 kg is manually added to the required launch capacity in GT-FAST, the model correctly predicts that a Delta II is required.

Table 7. Inputs into GT-FAST for Jason-2 and TIMED spacecraft models.[24, 26]

Spacecraft	Payload		Orbit		Pointing Requirement (deg.)	Mission Duration (years)
	Mass (kg)	Power (W)	Altitude (km)	Inclination (deg.)		
Jason-2	111.0	145.0	1336.0	66.0	0.1	5.0
TIMED	162.0	174.0	625.0	74.1	0.5	2.0

Table 8. Comparison between Actual and GT-FAST Predictions of Key Metrics for Jason-2.[23]

Spacecraft	Dry Mass (kg)	Wet Mass (kg)	Average Power (W)	Program Cost ($FY08M)	Launch Vehicle
Actual Jason-2	462.0	490.3	468.9	424.4	Delta II 7320-10
Predicted Jason-2	447.3	479.6	538.7	249.5	Taurus 2210
Prediction Error	- 3.2%	- 2.2%	+ 14.9%	- 41.2%	

[‡‡‡] The exclusion of the 24/7 communications unit and high-bandwidth downlink unit required slight changes to GT-FAST's internal logic since, as noted in Sec. II.A.2, these components are normally required within a cluster to pass internal consistency checks.

Table 9. Comparison between Actual and GT-FAST Predictions of Key Metrics for TIMED.[25,26,28]

Spacecraft	Dry Mass (kg)	Wet Mass (kg)	Average Power (W)	Program Cost ($FY08M)	Launch Vehicle
Actual TIMED	592.0	592.0	401.7	239.5	Delta II 7920-10 (with Jason-1)
Predicted TIMED	521.6	523.5	498.7	219.4	Taurus 2210
Prediction Error	- 11.9%	- 11.6%	+ 24.2%	- 8.4%	

V. Summary and Conclusion of Part 1

In summary, Part 1 has presented the internal mechanics and application of the Georgia Tech F6 Architecture Synthesis Tool (GT-FAST), a point design tool for rapid sizing and synthesis of fractionated satellite architectures. The manner in which fractionated designs are specified, including both discrete and continuous-variable inputs, was discussed, including the matrix representations of the launch manifest and placement of fractionated components in Figs. 4 and 5. Next described were the methods, models, and assumptions used in estimating elements of mass, power, and cost. The final section included sample outputs from GT-FAST for a notional fractionated architecture and presented a partial validation of the GT-FAST outputs against the currently-operational Jason-2 and TIMED satellites.

One important note to make is that the implementation of GT-FAST shown throughout Part 1 has been directed toward analysis of a DARPA F6 demonstrator intended for a circular low or medium Earth orbit. However, there is little that precludes GT-FAST from being modified for other fractionated spacecraft applications. In fact, it has already been adapted in one instance for analysis of a geosynchronous communications satellite. Existing subsystem mass, power, and cost models are interchangeable with other application-specific models a user may prefer, and launch vehicle capacity and cost models can also be easily updated. Additionally, the framework provided by the matrices in Figs. 4 and 5 makes the use of other fractionatable components (i.e., other than SSRs, MDPs, high-bandwidth downlinks, etc.) relatively simple to implement with minimal changes to internal logic. For example, earlier implementations of GT-FAST included power subsystem fractionation options through microwave power beaming hardware.

A wealth of possibilities exists for future expansion of GT-FAST. Currently, GT-FAST can size architectures consisting of up to nine fractionatable components, and future analyses may require the consideration of more components. This poses no problem to the current architecture of the GT-FAST point design tool, although it does present challenges in evaluating the resultant very large fractionated architecture trade space, as addressed in Part 2.

Additional future work on GT-FAST includes updates to the default launch vehicle database to include the most recent available launch vehicle performance and cost data for foreign, developmental, and proven domestic launch options (for more details, see Ref. 20). A more complete consideration of launch vehicle reliability may also be also worth consideration in future implementations of GT-FAST, and parametric cost models for fractionated spacecraft (as opposed to traditional monolithic spacecraft) would also be useful in future implementations. Also, future versions of GT-FAST might include options to size spacecraft based on discrete "parts kits" rather than based on "rubberized" parametric scaling relationships used in the present implementation. Consideration may also be given to a faster-running version of GT-FAST in MATLAB rather than the current (but more flexible) Excel platform.

Finally, a useful route for future work is the development of a comprehensive approach to defining and standardizing performance metrics for fractionated architectures. Currently, the primary metrics output by GT-FAST are mass, power, and cost. Other metrics specifying vehicle or payload performance characteristics are allowed to be user-specified and user-programmed, but it would be desirable for a standard set of such metrics to be hard-coded into GT-FAST and available to every user. A discussion of the selection of some of these metrics (and how they are traded against each other) is provided in Part 2.

The author believes that GT-FAST holds significant potential for future analyses of fractionated spacecraft and represents a critical piece of any framework aimed at permitting value-informed decisions for such architectures. The rapid analysis enabled by this tool becomes particularly useful when coupled with trade-space exploration strategies such as in Part 2, and the expandability and adaptability of the tool permit its use for potentially a wide variety of fractionated spacecraft applications. It is hoped that GT-FAST and the ideas it represents find broad use with engineers and decision-makers considering fractionated systems in the future.

Part 2

Trade-Space Analysis

With GT-FAST having been described in Part 1, the discussion of trade-space exploration for fractionated spacecraft can begin. In Part 2, the fractionated spacecraft trade-space (specifically aimed at the trade-space of the DARPA F6 demonstrator system) is explored for a clusters consisting of up to 6 fractionatable components, which is somewhat less than the maximum 9-component modeling capability of GT-FAST. The implementation of GT-FAST demonstrated in Part 2 uses five different classes of fractionatable components, consistent with those of Ref. 4. An architecture contains one 24/7 communication unit, one high-bandwidth downlink, a solid-state recorder, a mission data processor, and up to two payloads. Icons used in this part of the paper to represent these six individual fractionatable components are shown in Fig. 13. Payloads are specified by their mass, sunlight and eclipse power requirements, and pointing requirement. Unlike the Air Force Satellite Control Network (AFSCN) communications unit which every module is sized to include, a 24/7 communication unit provides near-continuous communications capability through a relay satellite such as the Tracking and Data Relay Satellites (TDRSs). High-bandwidth downlink units allow for high-volume downlinks that could not otherwise be provided with AFSCN or 24/7 links. A solid state recorder allows high-volume data storage, and a mission data processor is a resource allowing for onboard high-speed computing.

Figure 13. Icons for fractionated components in the Part 2 study.

Figure 14 is a depiction of an example six-component design named PF0248 that has been modeled using GT-FAST.§§§ PF0248 includes three modules, the first of which holds both payloads. The second module holds the 24/7 communication unit, high bandwidth downlink unit, and mission data processor. The third module holds the solid state recorder. The black block on each module signifies that all modules also include all essential support subsystems, such as structure, thermal, power, and others. Figure 14 also represents that Modules #1 and #2 are manifested to be flown on the same launch vehicle. Module #3 launches separately. Note that launch order is not represented by GT-FAST; that is, the representation in Fig. 14 does not preclude Module #3 from launching first or second. This three-module cluster is assumed to operate in a 370 km, 28.5° inclination circular orbit and is designed for a two-year mission with two communications payloads.

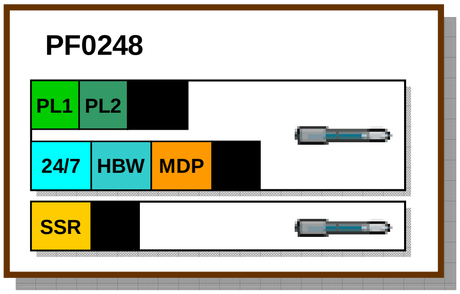

Figure 14. Architectural depiction of example design PF0248.

Outputs for this design from GT-FAST include three tables with mass, power, and module-level cost budgets broken down by subsystem for each of the three modules in the design. For brevity, these tables are not shown here, but examples can be found in Part 1. For reference, the boosted masses of Modules 1, 2, and 3 are 201.9 kg, 138.6 kg, and 138.9 kg, respectively, and the total power requirements are 522.9 W, 503.5 W, and 425.5 W, respectively. Table 10 shows the estimated cost budget for the entire system, which includes costs estimated at the module level and cluster level. Figure 15 graphically shows the cost breakdown of Table 10. As shown in Fig. 16, the launch vehicle selected for all three launches is the Pegasus XL at a cost of $22 million (FY08).**** The Pegasus XL's 450 kg payload capacity to the desired orbit was sufficient for all launches, and $22 million was the lowest launch cost in the database used for this launch vehicle selection (foreign and under-development vehicles were excluded).

§§§ Design PF0248 is actually a Pareto-optimal F6 design that will appear again later in this analysis.

**** Note that GT-FAST does not require all launches to use the same launch vehicle; this coincidence is due to the particular payload requirements for this set of launches.

Table 10. Overall PF0248 Cost Budget

Cost Element	Cost (FY08$M)
Module-Level Costs	
Module #1	55.9
Module #2	13.8
Module #3	13.4
Program Management	13.6
Software	38.7
Ground Segment Development	75.6
Operations	24.6
Pre-Margin Subtotal	*235.5*
Margin (25%)	58.9
Post-Margin Subtotal	*294.4*
Launch	44.0
Total	**338.4**

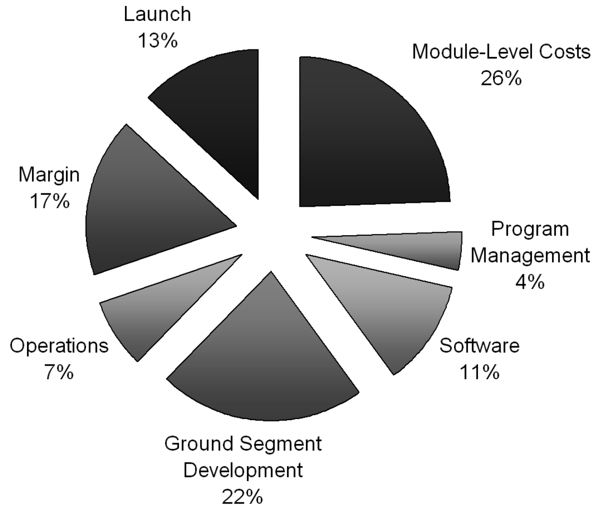

Figure 15. Breakdown of Costs from Table 10.

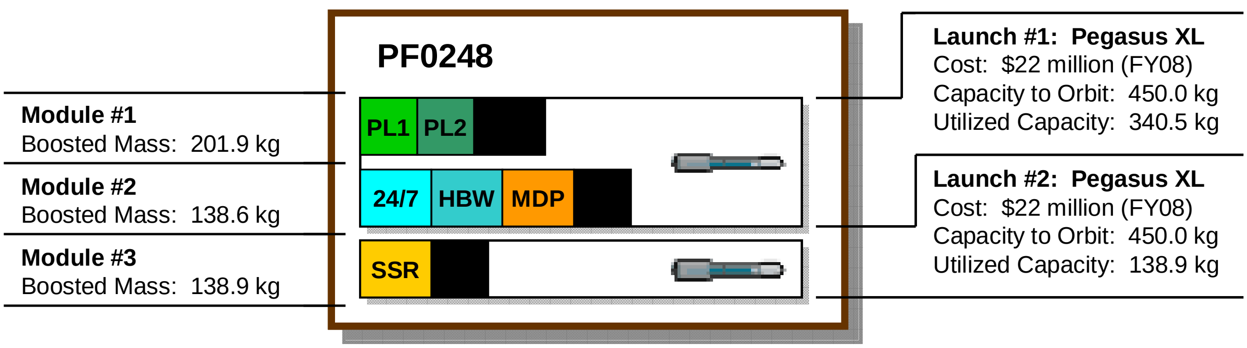

Figure 16. Launch Summary for Design PF0248.

VI. The Combinatorial Trade Space for Fractionated Spacecraft Designs

The focus of Part 2 of this paper is the application of GT-FAST, which has been described extensively in Part 1, to the exploration of the F6 fractionated spacecraft trade space. As such, it is necessary to define this trade space in terms of the variables or system characteristics that can be controlled by the designer. For the purposes of this study, the trade space consists of the discrete combinatorial options for the configuration of an F6 design. Continuous variables are set at their default values (i.e., all designs are set to a 370 km altitude, 28.5° inclination orbit, have a two-year design life, use a given set of payload mass and power assumptions, etc.). In this context, the PF0248 design described in earlier is just one of 3,190 designs considered in this trade study; the "PF" indicates that the cluster is partially fractionated (i.e., at least one module contains more than one fractionatable component) and the "0248" designation indicates that this design is the 248th design (out of 3,190) in the chosen enumeration scheme. The enumeration of these combinatorial options is covered in this section.

A. Theoretical Development: Size of Trade Space with no Constraints

As mentioned in Part 1 of this paper, the principal discrete inputs into GT-FAST – and the combinatorial configuration options in this trade study – deal with specification of which fractionatable components are present in which modules and which modules are carried on which launch vehicles. First we illustrate this problem for a simple 3-component architecture, and then we generalize this using the combinatorial definitions of Stirling and Bell numbers.

1. Example 3-Component Design Trade-Space

For this example problem, assume that the fractionatable components include exactly three distinct payloads (PL1, PL2, and PL3, using similar icons to Fig. 13).[††††] In the case of a monolithic architecture, all of these payloads would be housed in the same module (i.e., a one-module architecture). In the case of a fully fractionated architecture, each payload would be housed on its own module. In the fully fractionated case, this equates to three modules (one for each fractionatable component). However, solutions also exist for two-module architectures. For example, PL1 and PL2 could be housed in the same module while PL3 could have its own dedicated module. There are three such two-module architectures. All five architectures are shown pictorially in Fig. 17, and this collection of all possible architectures will be referred to as the Suite of Enumerated Architectures (SEA) for the 3-component trade-space.

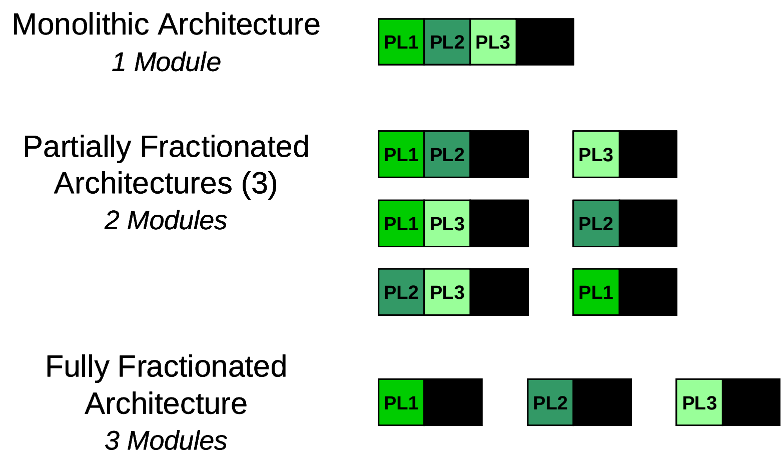

Figure 17. SEA for the 3-component case.

The SEA just defined accounts for all possible ways of placing components into modules. Another important consideration is how to place modules onto launch vehicles. Currently, GT-FAST considers launch vehicles independent of launch order; that is, from the perspective of manifesting, Launch Vehicle #1 (LV1) is indistinguishable from Launch Vehicle #2 (LV2). Thus, for each architecture in Fig. 17, several designs exist in terms of which modules are launched together or separately. For the one-module architecture (monolith), only one design exists since this one module must be launched on one launch vehicle. For each two-module architecture, there are two possible designs since the two modules can be launched together on a single vehicle or separately on two vehicles. For the three-module architecture, modules can be launched all together, all separately, or two can be launched together and one separately. For the three-component case considered in this example, this results in a total of 12 designs. All 12 designs are shown pictorially in Fig. 18, and this collection of all possible designs will be referred to as a Suite of Enumerated Designs (SED).

Finally, it must be recognized that the option to fractionate does not necessarily impose the decision to fractionate; the option can be exercised or not. That is, if one *can* fractionate a component, it does not necessarily mean that one *must* fractionate that component. Thus, if a 3-component case is considered, so must a 2-component case and 1-component case. This consideration introduces 12 additional designs, shown in Fig. 19. For example, a 2-component case can be defined for the sub-cases where the vehicle carries only PL1 and PL2, PL2 and PL3, or PL3 and PL1. For each of these 2-component sub-cases, a SEA and a SED can be defined using the logic from earlier; for a 2-component case, this simplifies into three designs, the first of which is a monolith, the second of which is two modules launched separately, and the third of which is two modules launched together on the same launch vehicle. The 1-component case consists of the simple scenarios where a payload is carried aboard a monolithic spacecraft. Thus, the 12 designs in Fig. 18 are added to the 12 designs in Fig. 19 to complete the collection of all designs that should be considered when deciding upon the configuration of a fractionated architecture which can accommodate up to three fractionatable components. This collection of designs is referred to in this paper as a Super-SEA.

[††††] In DARPA terminology, this is an example of a distributed-payload monolith since no subsystems are fractionated.[29]

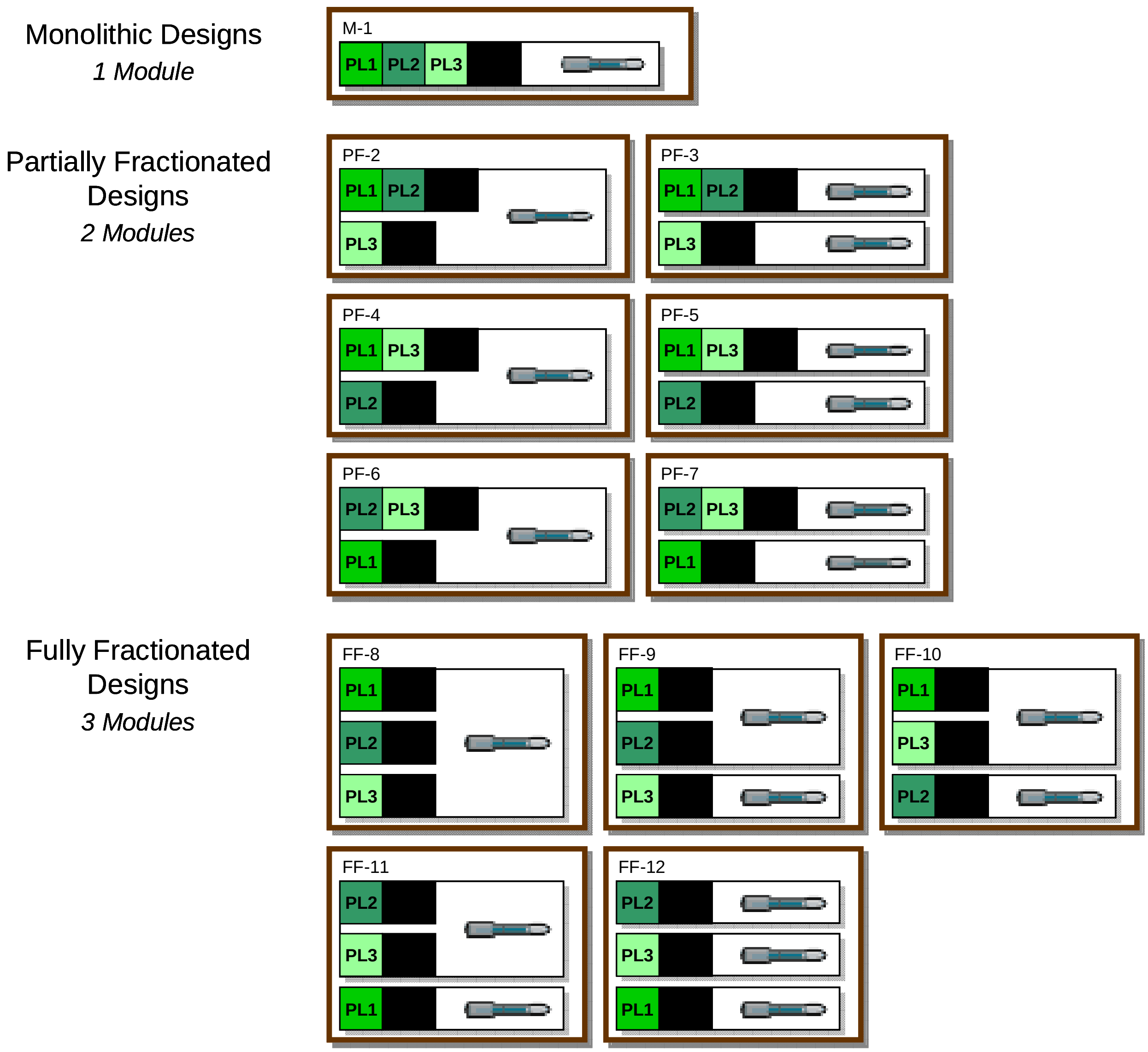

Figure 18. **SED for the 3-component case.**

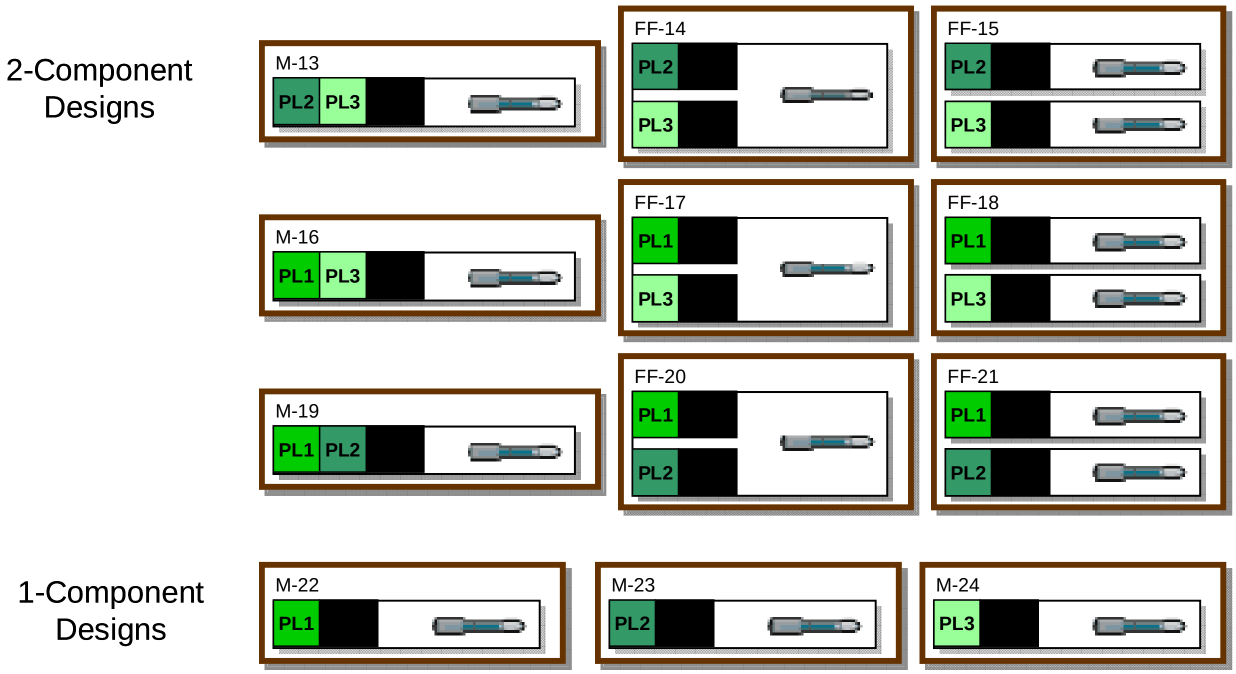

Figure 19. Designs that must be added to the 3-component SED to create a 3-component Super-SEA.

2. Generalized N-Component Design Trade Space

Thus far we have shown that, even in the relatively simple case where only 3 fractionatable components are considered, 24 designs exist which should be considered by the system designer or decision-maker. Now we generalize this combinatorial problem to one consisting of *N* fractionatable components.

As in the example problem, first we define the size of a SEA for an *N*-component case. Recalling that the size of the SEA is defined by the number of ways that exist to place *N* distinguishable fractionatable components into any of one to *N* modules, it can be seen that this is actually the sum of Stirling numbers of the second kind.[30] From the study of combinatorics, a Stirling number of the second kind, denoted as $S(n,m)$, physically describes the number of ways that exist of placing *n* distinct objects into *m* numbered but otherwise identical containers with no container left empty. Mathematically, $S(n,m)$ is defined by Eq. (3) below:

$$S(n,m) = \frac{1}{m!} \cdot \sum_{k=0}^{m} (-1)^k \cdot {}_m C_{m-k} \cdot (m-k)^n \tag{3}$$

Thus, in the 3-component example from earlier, we have 3 distinct objects (components) distributed into one, two, and three containers (modules), and the size of the SEA is $S(3,1) + S(3,2) + S(3,3) = 1 + 3 + 1 = 5$. A summation of this type has been defined in mathematics as a Bell number. Formally, a Bell number, and consequently the size of a SEA, is defined by Eq. (4). A table of these values is given by the second column of Table 11.

$$\text{Size of SEA} = B_N = \sum_{k=1}^{N} S(N,k) \tag{4}$$

Next, we concern ourselves with defining a SED, recalling that a SED is the number of ways that exist to place *N* distinguishable fractionatable components into any of one to *N* modules, and those modules into launch vehicles. The first step in this development is to recognize that the number of ways to distribute *K* distinguishable modules into any of one to *K* launch vehicles (considered indistinguishable, since launch order is not considered) is actually a

Bell number itself (by the same logic from above that distributing distinguishable components among indistinguishable modules is described by a Bell number). Thus, for each architectural possibility in a SEA (there were 5 in the demonstration case), there are B_K ways to distribute that architecture onto launch vehicles, where K is the number of modules in the architecture. Mathematically, we define the size of a SED consisting of N fractionatable components by the symbol D_N in Eq. (5). A table of values for D_N is given by the third column of Table 11. For the demonstration case, $D_N = 12$.

$$\text{Size of SED} = D_N = \sum_{k=1}^{N} S(N,k) \cdot B_k \tag{5}$$

Finally, what remains is to define the size of a Super-SEA, or the total number of designs a decision-maker should consider in his trade-space given that not all components that *can* be fractionated *must* be fractionated. As described earlier, assuming that all components have the option of being non-fractionated[‡‡‡‡], the Super-SEA considers the possibilities of including N components, N-1 components, N-2 components, etc., until the case where only one component is included. Initially, one might consider this to be simply the sum of D_N from N=1 to N. However, this does not account for the fact that, for example, there are multiple ways of choosing which components are included in the (N-1)-component SED (i.e., when defining the (N-1)-component SED, which component should be left out?). The number of ways of choosing X components for each new number of components is described mathematically by ${}_NC_X$. In the example case, there were ${}_3C_2 = 3$ ways of creating a 2-module SED. Thus, the total number of designs that a decision-maker should consider (the Super-SEA), denoted by F_N, is defined by Eq. (6). A table of values for F_N is given by the fourth column of Table 11. For the demonstration case, $F_N = 24$.

$$\text{Size of Super-SEA} = F_N = \sum_{j=1}^{N} \left(D_j\right) \cdot \left({}_N C_{N-j}\right) \tag{6}$$

Table 11. Sizes of SEA, SED, and Super-SEA as a function of number of fractionatable components (*N*).

N	Size of SEA (B_N)	Size of SED (D_N)	Size of Super-SEA (F_N)
1	1	1	1
2	2	3	5
3	5	12	24
4	15	60	130
5	52	358	813
6	203	2,471	5,810
7	877	19,302	46,707
8	4,140	167,894	416,510
9	21,147	1,606,137	4,073,412
10	115,975	16,733,779	43,289,930
11	678,570	188,378,402	496,188,630
12	4,213,597	2,276,423,485	6,095,737,867

‡‡‡‡ For payloads, this implies that some payloads can be left out, if trade studies warrant it. For subsystem components (e.g., communication equipment, data storage and processing equipment), non-fractionation implies that these components can either be omitted or automatically included within the generic mass and power budgets of the sizing and synthesis tool; in the latter case, clearly the location of the component (i.e., on which spacecraft it is housed) is no longer in the trade space. In practice, the designer may not wish to allow some components to be non-fractionated, in which case, the Super-SEA must be modified (and the trade-space will become smaller).

Table 11 summarizes the sizes of the SEA, SED, and Super-SEA for values of N up to 12. Recall that the SEA deals only with the number of possible clusters/architectures, the SED includes consideration of all launch options, and the Super-SEA addresses the fact that not all components that *can* be fractionated *must* be fractionated. The Super-SEA is of most relevance to the study in this paper, and the major observation one can take away from Table 11 is that the size of the Super-SEA increases dramatically with N. If N is doubled from 6 (as in this study) to 12, the number of designs that must be considered increases from 5,810 to 6,095,737,867 – a factor of over one million! The sheer size of the trade space for practical values of N – as well as the rapidity with which it expands as N increases – illustrates the need for a systematic method for enumerating and evaluating F6 designs.

B. Accounting for Constraints

According to Table 11, the Super-SEA for a 6-component design is F_6 = 5,810. However, earlier it was mentioned that the PF0248 example design is one of 3,190 designs considered in this trade study. This apparent discrepancy (between 5,810 and 3,190 possible designs) is the result of the exclusion of cases in the trade space due to practical application-specific constraints.

In the case of the components considered in the present analysis, it was assumed that every cluster must include a 24/7 communication unit, high bandwidth downlink unit, solid state recorder, mission data processor, and at least one payload. As a result of this constraint, the smallest architecture allowed to be considered is a 5-component design in which one of the payloads is omitted. Which payload to omit is of course an option, and so the number of cases considered is almost perfectly described by $D_6 + D_5 + D_5$ = 3,187 (see Table 11). The final three cases examined in the set of 3,190 designs are monolithic (single-module) spacecraft that contain intra-cluster wireless units and are thus F6-enabled[§§§§]; one of these cases contains PL1, another contains PL2, and the third contains both PL1 and PL2.

C. Enumerating Designs

While the discussion thus far has been concerned with counting the number of possible designs, the task still exists to enumerate, or list, each design so that it can be input and analyzed using GT-FAST. Covered in this section is an overview of how a SEA is enumerated, followed by a discussion of how this is extended to a SED and translated into inputs for GT-FAST.

1. Enumerating a SEA

To illustrate this process, Fig. 20 graphically shows how a 3-component SEA is generated; the same logic applies to the larger-dimensional 6-component designs considered for the trade studies in this paper. This logic is based on the idea of dividing strings of component orderings using indistinguishable partitions.

As illustrated in Fig. 20, the enumeration process starts with the definition of both component permutations and partition schemes. The component permutations simply show all possible ways of ordering the N components (where each component is given a number from 1 to N). For an N-component design, there will be $N!$ such orderings. The partition schemes are more involved and are generated from a full factorial design with N-1 factors (since at maximum there can be N-1 partitions in an N-component string), each of which has N levels. The resulting list is filtered such that all remaining schemes are sequential; for example, since each number in the partition string represents the location of an indistinguishable partition, there is no difference between the schemes [1 2] and [2 1], both of which indicate that partitions are to be placed after the first and second components in the component permutation string.

Next, each partition scheme is applied to each component permutation; i.e., partitions as defined by each partition scheme are placed within each of the component permutations. In the case of the 3-component design, this results in a list of 36 clusters. However, many of these clusters are duplicates; for example, the 3--21 cluster is the same as the 3--12 cluster (where the dashes indicate partitions). Once duplicates are eliminated, the enumeration of a SEA has been completed.

[§§§§] As described in Part 1, if only a single module exists in a cluster, GT-FAST excludes the intra-cluster wireless unit from mass, power, and cost budget estimation.

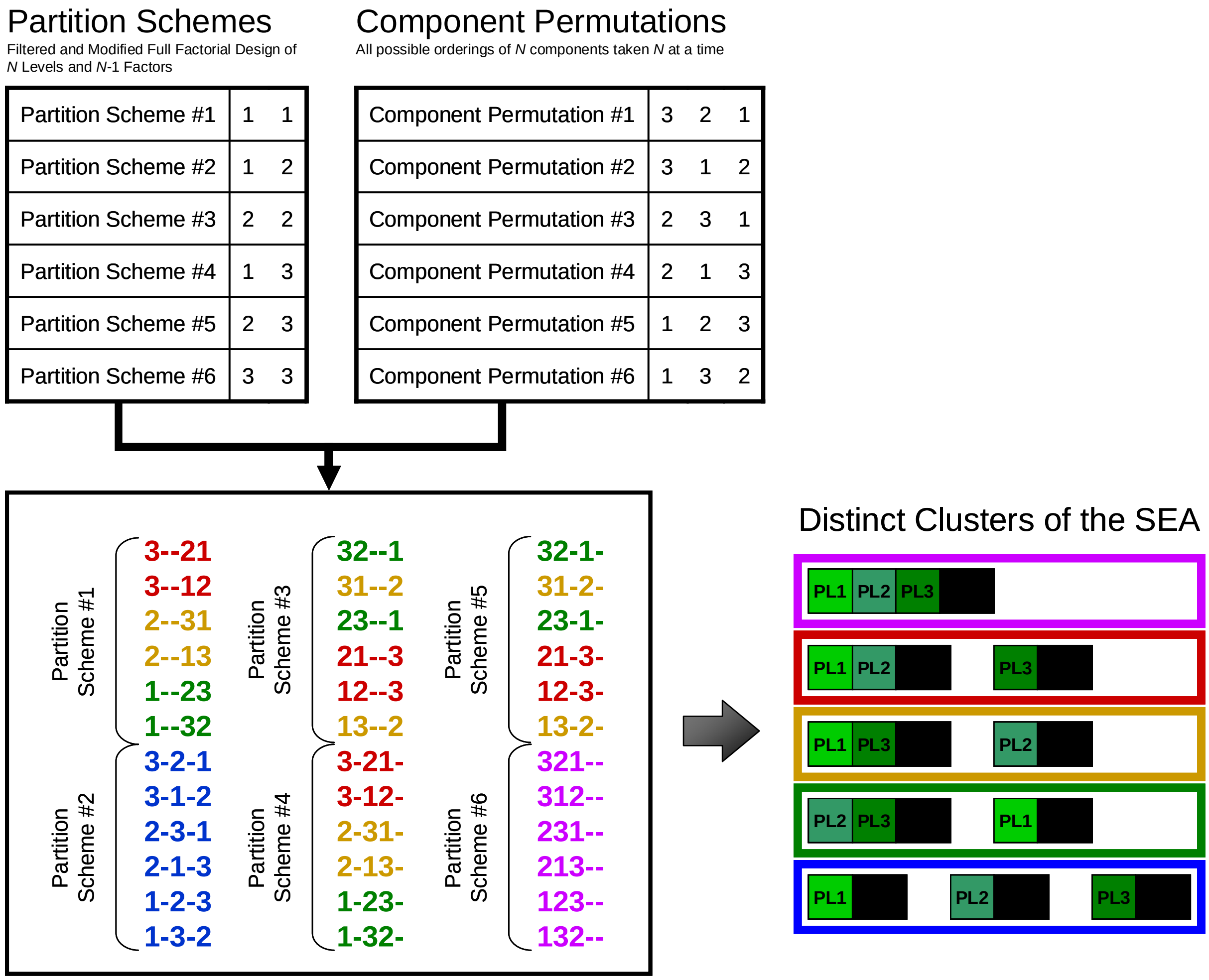

Figure 20. Example Enumeration Process for a 3-Component SEA.

Note that the red, yellow, green, blue, and magenta colors in the 36 partition/permutation combinations indicate to which of the final six clusters in the SEA each combination corresponds.

2. Enumerating a SED and Defining the GT-FAST Input

As defined in Sec. VI.A.2, a SEA is the set of architectures available when placing N distinguishable components in up to N indistinguishable modules. In comparison, a SED accounts for the placement of K distinguishable modules of a particular cluster into up to K indistinguishable launch vehicles. Thus, the process for enumerating the SEA can be reapplied for enumeration of the SED. Once SEDs are defined, they can be appended to each other to define all cases to examine; for example, recall that the 6-component design evaluated here consists essentially of one D_6 and two D_5 SEDs ($D_6 + D_5 + D_5 = 3{,}187$). Thus, the full evaluation here involves evaluation of each SED.

The input into GT-FAST for SED evaluations is thus a list of strings describing each design to be evaluated. This format involves a string of numbers and commas. The first nine numbers are associated with the available fractionatable components and vary from 1 to 9 for each of the nine components available for modeling in GT-FAST. The first nine commas indicate divisions between modules; two sequential commas with no numbers between them indicate an empty (nonexistent) module. Similarly, the second nine numbers in the string represent modules as they are to be placed in launch vehicle, and the second nine commas indicate divisions between launch vehicles. GT-FAST then translates this string into the input matrices described in Part 1 of this paper; the resulting matrices for PF0248 are shown in Figs. 21 and 22. Once the string has been read into GT-FAST, the sizing routines execute as described in Part 1. For the trade study discussed next, this process is repeated in an automated fashion for each string in the list of SEDs to be evaluated.

ARCHITECTURAL									
Number of Modules	3	Consistency Check							
	Payload 1	Payload 2	Payload 3	24/7 Comm 1	24/7 Comm 2	High BW 1	High BW 2	SSR	MDP
Module 1	P	P	N	F	N	F	N	F	F
Module 2	F	F	N	P	N	P	N	F	P
Module 3	F	F	N	F	N	F	N	P	F
Module 4	F	F	N	F	N	F	N	F	F
Module 5	F	F	N	F	N	F	N	F	F
Module 6	F	F	N	F	N	F	N	F	F
Module 7	F	F	N	F	N	F	N	F	F
Module 8	F	F	N	F	N	F	N	F	F
Module 9	F	F	N	F	N	F	N	F	F
	Create this Architecture!								

Figure 21. Input Matrix Mapping of Components to Modules for PF0248.

LAUNCH									
Number of Launches	2	Launch Consistency Check							
	LV 1	LV 2	LV 3	LV 4	LV 5	LV 6	LV 7	LV 8	LV 9
Module 1	0	N	N	N	N	N	N	N	N
Module 2	0	N	N	N	N	N	N	N	N
Module 3	N	0	N	N	N	N	N	N	N
Module 4	N	N	N	N	N	N	N	N	N
Module 5	N	N	N	N	N	N	N	N	N
Module 6	N	N	N	N	N	N	N	N	N
Module 7	N	N	N	N	N	N	N	N	N
Module 8	N	N	N	N	N	N	N	N	N
Module 9	N	N	N	N	N	N	N	N	N
Tot. Launch Mass (kg)	340.49	138.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Launch Cost ($FY08M)	22.00	22.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 22. Input Launch Manifest Matrix for PF0248.

VII. Defining Output Metrics

As mentioned earlier, key information output by GT-FAST for each point design is a mass, power, and cost budget for the cluster and for each module in the cluster. In addition, crucial to the evaluation and selection of potential F6 designs is the output of user-defined metrics that characterize performance attributes. In the present trade study, sixteen objectives are used in the assessment of potential designs, only one of which is a standard GT-FAST cost, mass, or power output. Five of these objectives are described in depth here.

A. Ability to Achieve Incremental, Independent-Order Launches

One objective of the F6 fractionated spacecraft system considered here is demonstration of the ability of the design to accommodate incremental buildup in capability and independence of launch order.[1] This objective, named O_6, is directed toward allowing System F6 to demonstrate attributes of flexibility and responsiveness that may be of interest to future customers of fractionation.

To capture the performance of a particular design with respect to this objective, quantified is the number of unique orders in which a given design can be launched with the restriction that the launch order must not launch a payload before an SSR, MDP, and high-bandwidth downlink unit are already on-orbit (or contained in the same launch as the payload). This restriction is here termed the "functional payload rule" and results in a launch order not being counted unless payloads can be operational once they reach orbit. The more usable launch orders that exist for a design, the greater its score is according to this objective.

To illustrate more clearly how this objective is computed, Fig. 23 shows the two possible launch orders for the PF0248 example design. In this case, the launch order on the left is not counted toward the O_6 score since both payloads are launched before an SSR is available on-orbit. However, in the launch order on the right, the payloads can begin operations soon after launch since the SSR has been pre-launched. Thus, the score for PF0248 in this category is $O_6 = 1$. This value is actually the median score for all 3,190 designs in the trade study; the mean is $O_6 = 1.54$, the minimum is $O_6 = 0$, and the maximum is $O_6 = 72$ (for the fully fractionated design).

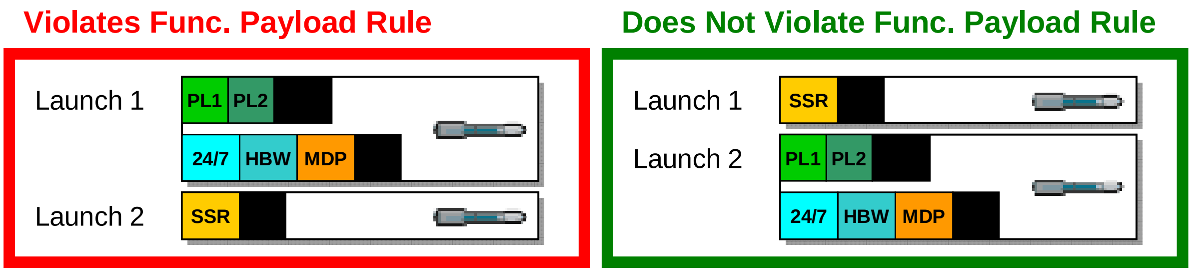

Figure 23. Application of the O_6 objective on PF0248.

B. Short Time to Operational Capability

Another objective of interest, named O_7, is minimization of the time required to reach operational capability for the F6 system, constrained by the requirement that the first launch be within four years of program start[1]. This objective is quantified by counting the minimum number of launches required for a given design to reach an initial operational capability with one or more payloads without violating the functional payload rule mentioned in Sec. VII.A. A low score for O_7 is desirable, and this imposes an inherent penalty on highly fractionated designs. For example, in a fully fractionated design where each module is launched separately, the minimum number of launches is four. The score for PF0248 is $O_7 = 2$ (both launches must take place for operational capability to be reached). The median score for all 3,190 designs is $O_7 = 2$, the mean is $O_7 = 1.84$, the minimum is $O_7 = 1$ (for example, a monolith), and the maximum is $O_7 = 4$ (e.g., for fully fractionated designs).

C. Relevance to Prospective Fractionation Customer

A third objective of the F6 demonstration program is to demonstrate the relevance of the fractionated spacecraft approach for future users. Since these future users will essentially be providing payloads, it is reasonable to believe that the most relevant design to them would be one with a dedicated payload module (in other words, a design in which the payload is alone in its own module, with the other supporting components in one or more separate modules). This notion gives rise to a third objective, quantified through objective O_{13}, which is the minimum number of non-payload components accompanying a payload for a given design. This effectively captures the degree of payload isolation, and O_{13} varies from $O_{13} = 0$ (for designs incorporating a dedicated payload module, such as PF0248) to $O_{13} = 4$ (for example, for monoliths), where low values are preferable. The median value of O_{13} for all 3,190 designs is $O_{13} = 0$, and the mean is $O_{13} = 0.36$.[*****]

D. Ease of Accepting New Components

A key flexibility-related metric is the ease with which new components can be added to the cluster. The desire to launch a new component may stem, for example, from increases in market and capability demand or availability of technology upgrades and enhanced capabilities; a desirable characteristic is for the cost of adding these components to be low.

The metric chosen here to represent the ease with which a design accepts new components is the average cost of adding or replacing a component of the cluster. This metric, named O_{10} or $C_{add/replace}$ as defined in Eq. (7), considers the fact that a given single component i can be added to the cluster in one of two practical ways. First, the user could choose to launch the needed component as part of a module that is a duplicate of one that is already on-orbit. This strategy takes advantage of the fact that no research, development, test, and evaluation (RDT&E) costs are incurred since the module has already been manufactured before. The cost to implement this option is reflected as $C_{i,existing}$ in Eq. (7). The second option for the user is to simply launch a module with the single component i that is needed (for an example of a single-component module, see Module #3 in Fig. 16). This strategy takes advantage of the low cost associated with a small, single-component module but has the disadvantage that, unless this module had been developed for the original cluster, RDT&E costs are incurred. The cost of this option is $C_{i,separate}$ in Eq. (7).

[*****] This is an interesting result, since it highlights the fact that over 50% (actually, 71%) of the 3,190 possible designs have the characteristic of payload isolation (i.e., a dedicated payload module).

The $C_{add/replace}$ metric is based on the idea that a user would prefer the lowest-cost option when it comes to adding or replacing a single component. However, since it is not obvious which components will require addition or replacement in the future, the average is taken over all the *n* possible components of the lowest-cost addition/replacement options. This is reflected in Eq. (7), and this metric is evaluated in GT-FAST for each of the 3,190 designs considered. These can be formed into a histogram, shown in Fig. 24. In this particular problem, the minimum $C_{add/replace}$ is $42.5 million and the maximum is $83.5 million, with a median of $52.2 million. If only this objective were considered, a fully-fractionated design (consisting only of single-component modules) would be optimal since each single-component module is pre-developed.

$$\overline{C}_{add\,/\,replace} = \frac{1}{n}\sum_{i=1}^{n}\min\left(C_{i,separate}, C_{i,existing}\right) \qquad (7)$$

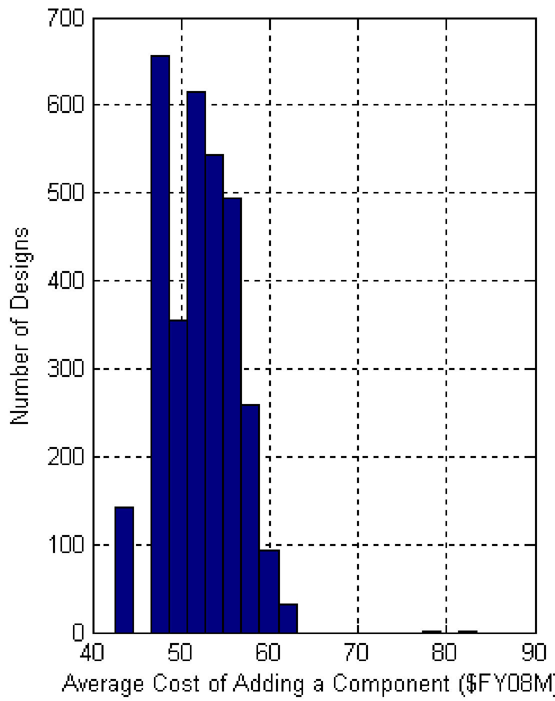

Figure 24. Histogram of $C_{add/replace}$ metric.

E. Robustness to Threats

An important advantage to a fractionated spacecraft is its inherent robustness to external threats. To generate a measure of this objective, named O_{15}, a score is formulated which reflects the expected degree of functionality for the cluster after it is subject to failure of an entire module (which could be caused, for example, by orbital debris strikes or anti-satellite missile attacks). For example, if both payloads are lost when a module fails, then it is assumed that functionality is effectively zero. If only subsystem components are lost (for example, the solid state recorder, the high-bandwidth downlink unit, etc.), then a lesser degradation is imposed. When this metric is computed, the loss of each module is considered to be equally probable and is equally weighted in determining the expected functionality score after a module failure; the value of the O_{15} functionality score is always between zero and unity.

The O_{15} = 0 value occurs for monolithic spacecraft where the loss of a module is also the loss of the cluster. In this study, a design with O_{15} = 1 is of course never found; the maximum performance value for this objective is O_{15} = 0.54 for fully fractionated designs with two payloads. This result itself is quite interesting in that it implies that a fully fractionated spacecraft on average might be expected to retain over half its functionality if a module is lost at random, which is quite a benefit over a monolithic spacecraft. For this metric, the median and mean were quite close at 0.42 and 0.41, respectively. For reference, the PF0248 design scores rather low in this category, with O_{15} = 0.23; this is largely due to the placement of both payloads in the same module, since the cluster will have virtually no functionality if one of the three modules is lost.

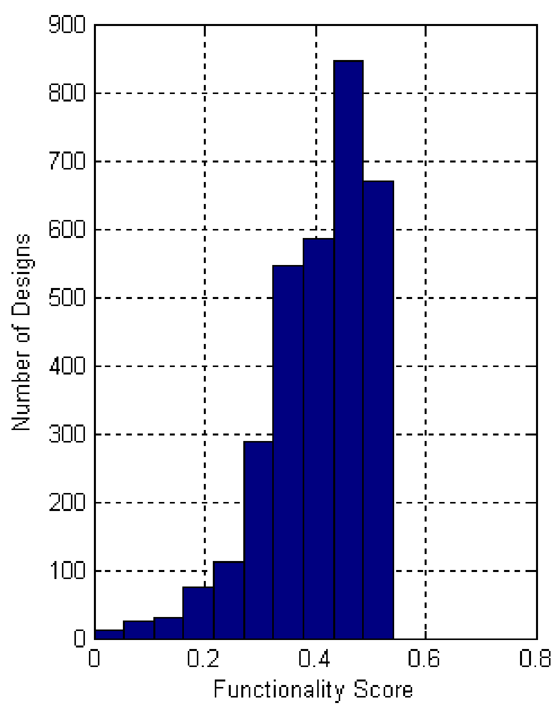

Figure 25. Histogram of O_{15} robustness to threats functionality score metric.

F. Other Objectives

The five quantified objectives above are only examples of the total 16 objectives used in the following trade space exploration. One of the most obvious objectives not discussed was total program cost, a histogram of which is shown in Fig. 26. Total program cost is a core output of GT-FAST and did not need to be programmed as a user-specific output. Details on the cost estimation assumptions and procedures can be found in Part 1 of this paper.

Table 12 summarizes the 16 objectives considered. These objectives were the result of an extensive brainstorming session, and although each is defined well in a conceptual and qualitative sense, not all can be resolved quantitatively to a level of fine detail with the sizing information available from GT-FAST. In the case of eight objectives, fine resolution is available similar to the five described earlier. In the case of four objectives, coarse resolution is given based on qualitative considerations (for example, programmatic risk is likely correlated with complexity in terms of the average size of modules and the number of modules that must be developed, but the exact correlation is unclear at this stage and is divided only into categories of low, medium, and high risk). Insufficient information existed for the evaluation of the final four objectives, and these were not analyzed.

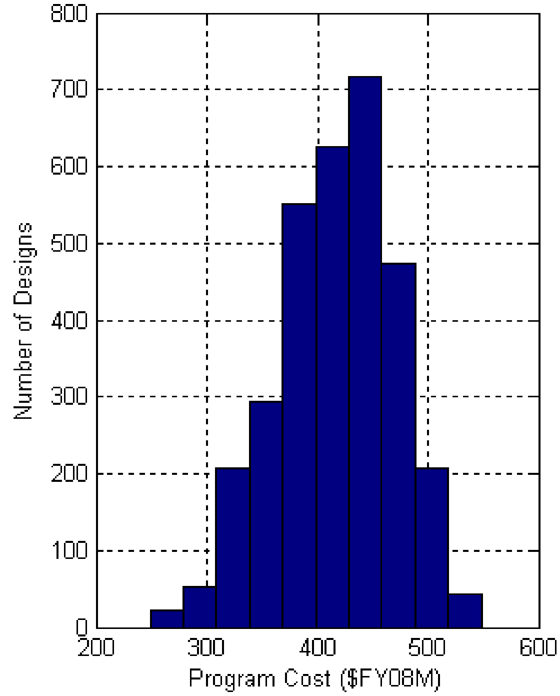

Figure 26. Histogram of F6 Program Cost.

The rightmost column of Table 12 provides the relative weighting assigned to each objective based on an interview session using an analytic hierarchy process (AHP) prioritization matrix. For example, this column shows that relevance to potential fractionation customers is the highest-priority objective. Note that a weighting is not given to total program cost because this objective will be displayed separately on the second axis of a Pareto front plot in the trade space exploration; the weightings shown here will only be used to aggregate all non-cost objectives into a single "performance" or "effectiveness" score. Also note that the fact that four objectives are unresolvable at this stage is accounted for by giving identical scores to every design in those categories (i.e., not by assigning a weight of zero).

Table 12. Summary of the 16 objectives considered for this trade space exploration.

Objective		Resolution	Weighting
No.	Name		(× 100)
1	Availability	Not Available	1.4
2	Ground Signature Minimization	Coarse	3.5
3	Payload/Mission Performance	Coarse	4.3
4	Low Total Program Cost	Fine	N/A
5	Low/Diversified Programmatic Risk	Coarse	5.1
6	Ability to Achieve Incremental, Independent-Order Launches	Fine	1.9
7	Short Time to Operational Capability	Fine	1.5
8	System Longevity	Not Available	4.5
9	Manufacturability & Testability	Fine	4.7
10	Ease of Accepting New Components	Fine	7.5
11	Ease of Changing Cluster Configuration	Not Available	8.4
12	Reprogrammability & Functional Reconfiguration	Not Available	11.1
13	Relevance to Potential Fractionation Customer	Fine	13.2
14	Robustness to Failure	Coarse	12.7
15	Robustness to Threats	Fine	6.0
16	Extensive Technology Demonstration	Fine	10.3

VIII. Visualizing the Trade Space

Having computed the sixteen metrics shown in Table 12 for each of the 3,190 designs in the defined trade space, the problem exists of how to filter and plot this data in a way conducive to selecting desirable designs and observing trends. A multitude of techniques exist, and by no means are the methods presented here the only ones possible. However, we have found these to be intuitive and helpful to the exploration of the F6 trade space.

The approach of this analysis makes extensive use of Pareto frontiers (or fronts), which allow for identification of non-dominated solutions in an objective space. In the representations that will be shown, each design will be represented by a point whose coordinates are the values of two objectives associated with the design. The Pareto front is the set of points which are non-dominated in the objective space (i.e., at a non-dominated point, it is impossible to find another design that improves all objectives simultaneously). This approach has the advantage that it helps narrow the trade space significantly and avoids the naming of a single optimum solution, which by definition does not exist for a multiobjective problem. It also provides helpful visualizations which allow identification of the "knees" of Pareto fronts, if they exist.

One disadvantage of Pareto fronts is that they quickly become unwieldy and difficult to visualize as the number of objectives being tracked increases past two. To overcome this limitation, the second part of this analysis uses the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to aggregate all non-cost objectives into a single "performance" or "effectiveness" metric (analogous to the effectiveness parameter in the technology frontier approach of Ref. 31). Although there are always limitations whenever multiple objectives are combined into a single metric, we find this useful in identifying several highly desirable designs. It is worth noting that nowhere in this analysis do we identify a definitive optimum design due to the multiobjective nature of the problem; we limit our discussion to noting several promising designs and, importantly, the relevant characteristics common to them.

A. Basic Two-Objective Pareto Fronts

1. Minimum Launches to Operational Capability vs. Feasible Launch Combinations

Figure 27 shows an interesting Pareto frontier in the O_7 vs. O_6 objective space. Recall that O_6 represents the number of feasible launch combinations (i.e., those that do not violate the functional payload rule) and O_7 represents the minimum number of launches required for an initial operational capability. The user would prefer to maximize O_6 and minimize O_7, all other things being equal, so the ideal solution would be in the bottom right corner of Fig. 27 and the Pareto front (the red line) has a positive slope. In part, what this Pareto front indicates is the inherent design trade between maximizing the number of possible launch orders and minimizing launches to initial operational capability. If the designer wishes to have operational capability after one launch, it is impossible to achieve more than three feasible launch orders. If the designer wishes to be able to choose from 72 possible launch orders, then the minimum launches to initial capability cannot be less than four.

Recall also that each blue "x" in Fig. 27 represents a design.[†††††] Thus, this allows for the identification of designs on the Pareto front (also called Pareto-optimal solutions). At the location marked as A in Fig. 27, 30 designs exist which have three feasible launch combinations and a single launch for initial operational capability. One of these designs is PF0031, shown in Fig. 27. This design groups all essential components into a single module launched on a dedicated launch vehicle (allowing for the single-launch initial capability) and the remaining two components on their own dedicated modules, each of which having their own dedicated launch vehicle (allowing for multiple feasible launch combinations). At the other extreme, at the location marked as D in Fig. 27 sits the fully-fractionated design with six dedicated launches (the maximum possible), allowing for the maximum possible feasible launch orders but simultaneously requiring at least four launches for an initial operational capability. The progression from A to D can be seen in example designs at locations B and C as shown in Fig. 27. Here, the usefulness of the Pareto front approach should be clear in that it identifies the set of best designs a designer could choose; if a designer were only interested in the O_6 and O_7 objectives, he should choose one of the designs (as indicated by Fig. 27) at the A, B, C, or D locations.

2. Ease of Accepting New Components vs. Total Program Cost

Figure 28 shows a particularly interesting Pareto frontier in the O_{10} vs. O_4 objective space. Recall that O_4 is the total program cost and O_{10} is the average cost of adding or replacing a component of the cluster. The user would prefer to minimize both O_4 and O_{10}, all other things being equal, so the ideal solution would be in the bottom left

[†††††] For the discrete outputs of Fig. 27, many different designs might have the same combination of O_6 and O_7, so the "x" marks overlap and 3,190 distinct marks are not visible.

corner of Fig. 28 and the Pareto front (the red line) has a negative slope. In part, what this Pareto front indicates is the inherent design trade between minimizing the total program cost and minimizing the average cost of replacement; that is, an additional investment must be made up-front in the form of total program cost if the user wishes to reduce the cost of replacement.

Recall also that each blue "x" in Fig. 28 represents a design, again allowing for the identification of Pareto-optimal designs. At the location marked as A lies a monolithic spacecraft design that excludes PL1 (the more massive and more costly payload).[‡‡‡‡‡] Design A has a very low total program cost but also has the highest average cost of adding a component since the addition of a component requires the launch of a new monolithic spacecraft. For the small added cost of an intra-cluster wireless unit (i.e., the creation of a "fractionatable" monolith), Design B has a $20 million reduction in the average cost of adding a component since the option exists to send small single-component modules instead of a new monolithic spacecraft. Design C is interesting because it lies at a very distinct "knee" on the Pareto front and has both a low program cost and low average component replacement cost. This design fractionates the payload and solid state recorder each into single-component modules but permits the 24/7 communication unit, high bandwidth downlink unit, and mission data processor to remain in the same module; this particular compromise between the economies of scale of the traditional monolith and flexibility of the fully fractionated spacecraft presents an appealing design from the perspective of objectives O_4 and O_{10}. Designs D and E, each of which is fractionated among more modules than Design C, have slightly lower costs of adding components but are significantly more expensive to develop and field.

B. Pareto Fronts involving TOPSIS Scores

In contrast to Figs. 27 and 28, which considered only two objectives at a time, Figs. 29 and 30 aggregate all non-cost objectives into a single score and plot this score against total program cost for each of the 3,190 designs. To create this score, the TOPSIS multi-attribute decision-making technique is used, and objective weightings are taken from Table 12. As a result, the designs that will be identified on the Pareto fronts in Figs. 29 and 30 are predominantly "compromise" solutions that perform well in many categories but perhaps are not the best in any single category.

The Pareto front in Fig. 29 exhibits several interesting characteristics. The lowest-cost design, the monolithic spacecraft carrying only PL2 as a payload, anchors the Pareto front in the bottom left. The design with the highest TOPSIS score, the fully fractionated design with dedicated launches and both payloads, anchors the Pareto front in the top right. Interestingly, the TOPSIS score increases dramatically at a program cost about $10 million above the anchoring monolithic spacecraft; the design at the knee of this segment of the Pareto front is PL2-PF2874, a three-module design in which all three modules launch on the same vehicle. To the right of PL2-PF2874 are six designs that offer incremental improvements in TOPSIS score for significant increases in program cost. These designs can be grouped into two families of designs, as outlined by red and blue boxes in Fig. 29. Each family has a common module configuration and has variations only in the number of launch vehicles. The final two Pareto-optimal designs are fully fractionated designs which offer significant TOPSIS score increases (and value to the customer) but at significant costs. One remark to make at this point is that Fig. 29 has effectively narrowed the trade space of 3,190 designs to just eleven designs (a reduction by a factor of 290).

One limitation of the Pareto-optimal designs identified in Fig. 29 is that many of them (particularly the lower-cost options) are single-launch solutions, which probably do not meet the expectations of DARPA for the F6 program. Another limitation is that most of these Pareto-optimal designs include only a single payload, another attribute that probably does not meet DARPA's expectations. Figure 30 shows the results of filtering out these undesirable cases, i.e., cases with only one launch or only one payload. The resulting Pareto front is shifted somewhat to the lower right compared to the first front, and a major difference is that the lowest-cost option is now above $300 million, whereas it was about $250 million in Fig. 29. This lowest-cost option is PF0730, a two-module, two-launch design, and the highest-TOPSIS-score design remains as the fully-fractionated design with dedicated launches. An interesting knee in this curve occurs at a cost of $340 million: The PF0248 design, the example design carried through examples earlier in this paper, is a two-launch solution with the significant advantage of a module dedicated only to carrying payloads, meaning PF0248 performs well in the heavily-weighted category of relevance to potential fractionation customers. PF0248 also has average performance in several other categories and, along with a low cost, makes an appealing option. Significantly, PF0248 is identical to the PL2-PF2874 design at the knee of the curve in Fig. 29 except that (1) PF0248 includes both PL1 and PL2 and (2) PF0248

[‡‡‡‡‡] In fact, the three designs that can be seen with a replacement cost above $75 million are the three monoliths (one with PL1 only, another with PL2 only, and the third with both PL1 and PL2).

launches on two launch vehicles instead of one. These differences are identical to the constraints imposed on Fig. 29 to result in Fig. 30, and so it is interesting that the design at the knee of the Fig. 29 Pareto front is nearly the same as the design at the knee of the Fig. 30 Pareto front. Again, however, it is not claimed that PF0248 is the "best" design since Fig. 30 shows that higher TOPSIS scores are possible with the expenditure of additional funds on the total program cost; the designs that achieve these higher scores gradually consist of more modules and more launches, eventually leading to the highest-scoring fully fractionated design mentioned earlier.

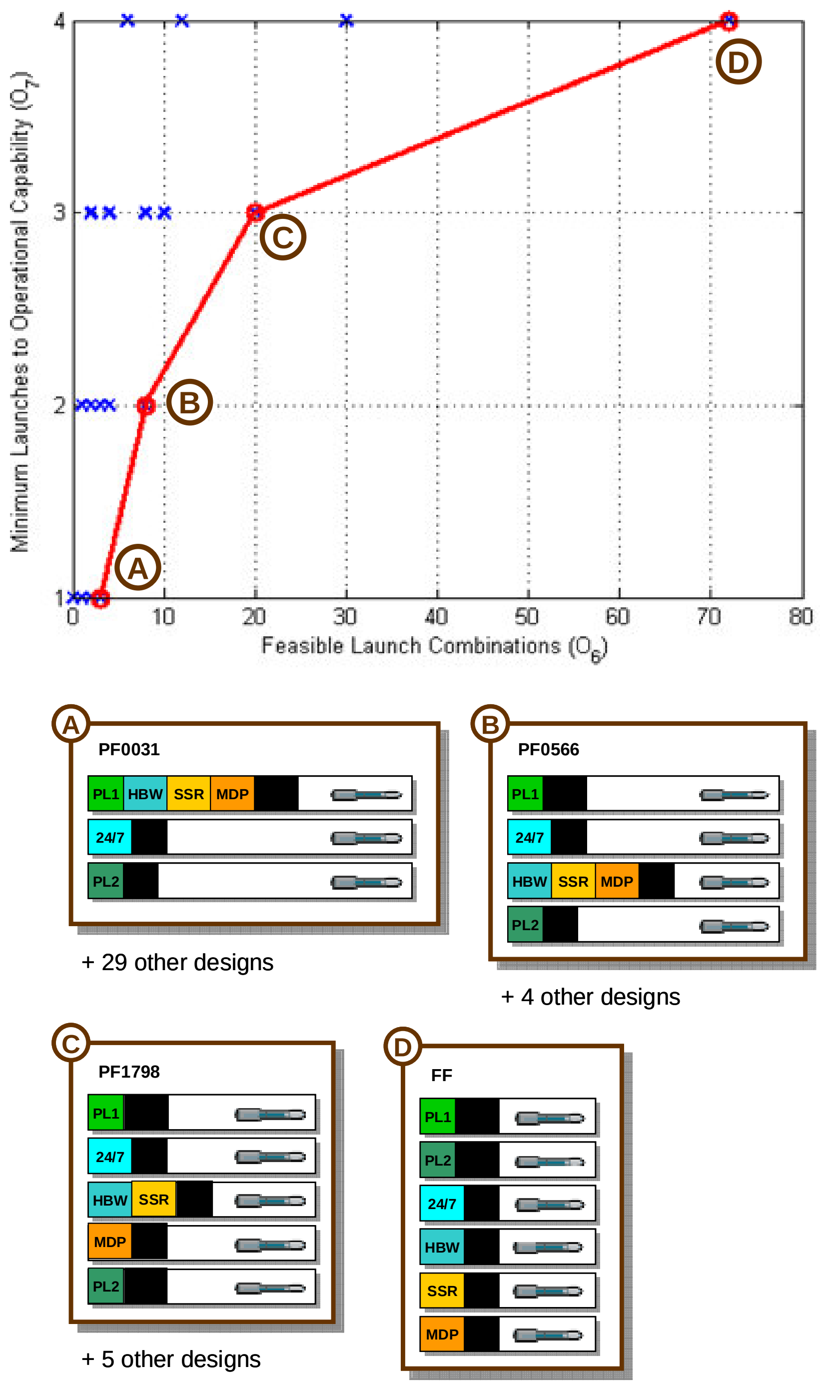

Figure 27. Pareto front between objectives O_6 and O_7.

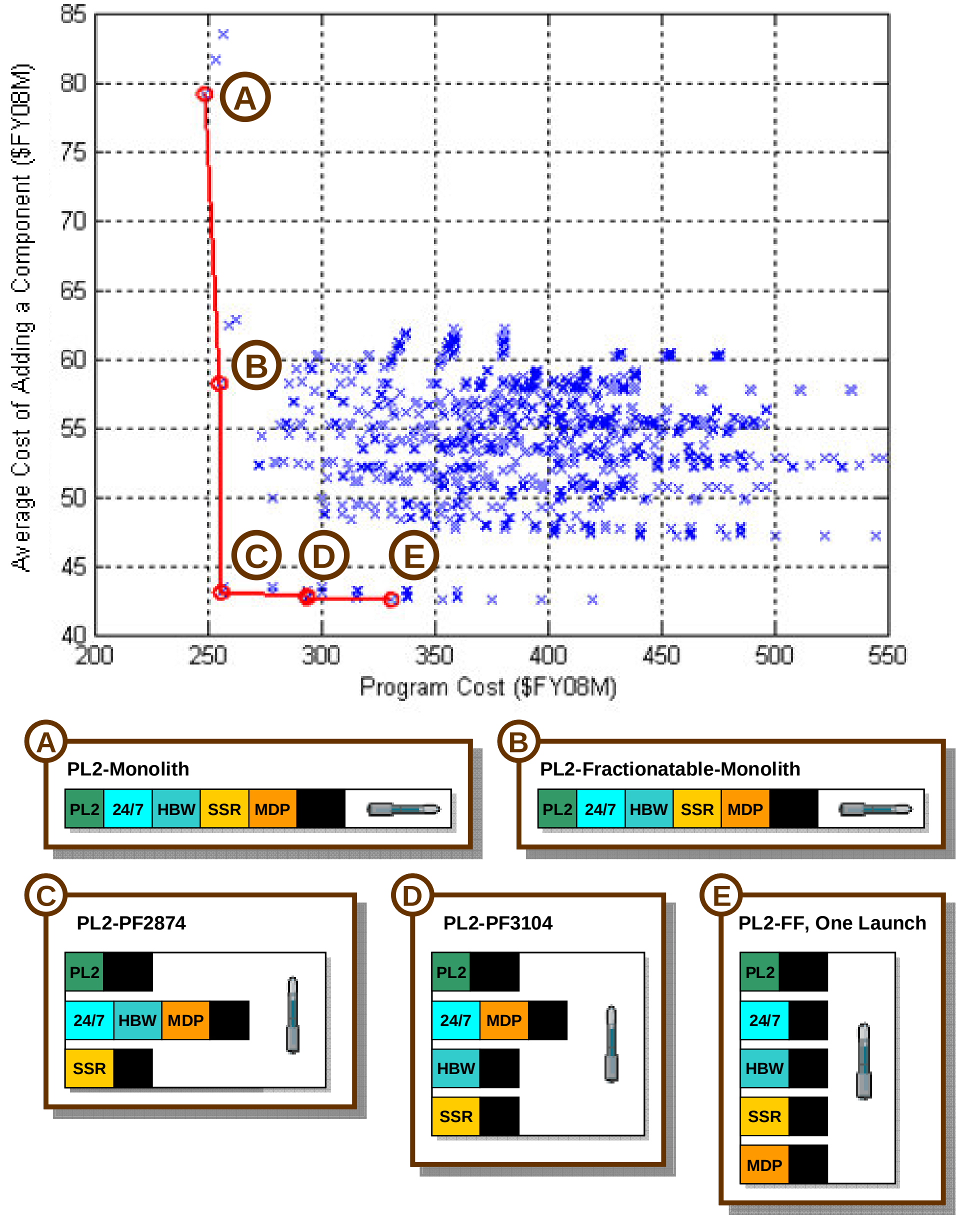

Figure 28. Pareto front between objectives O_4 and O_{10}.

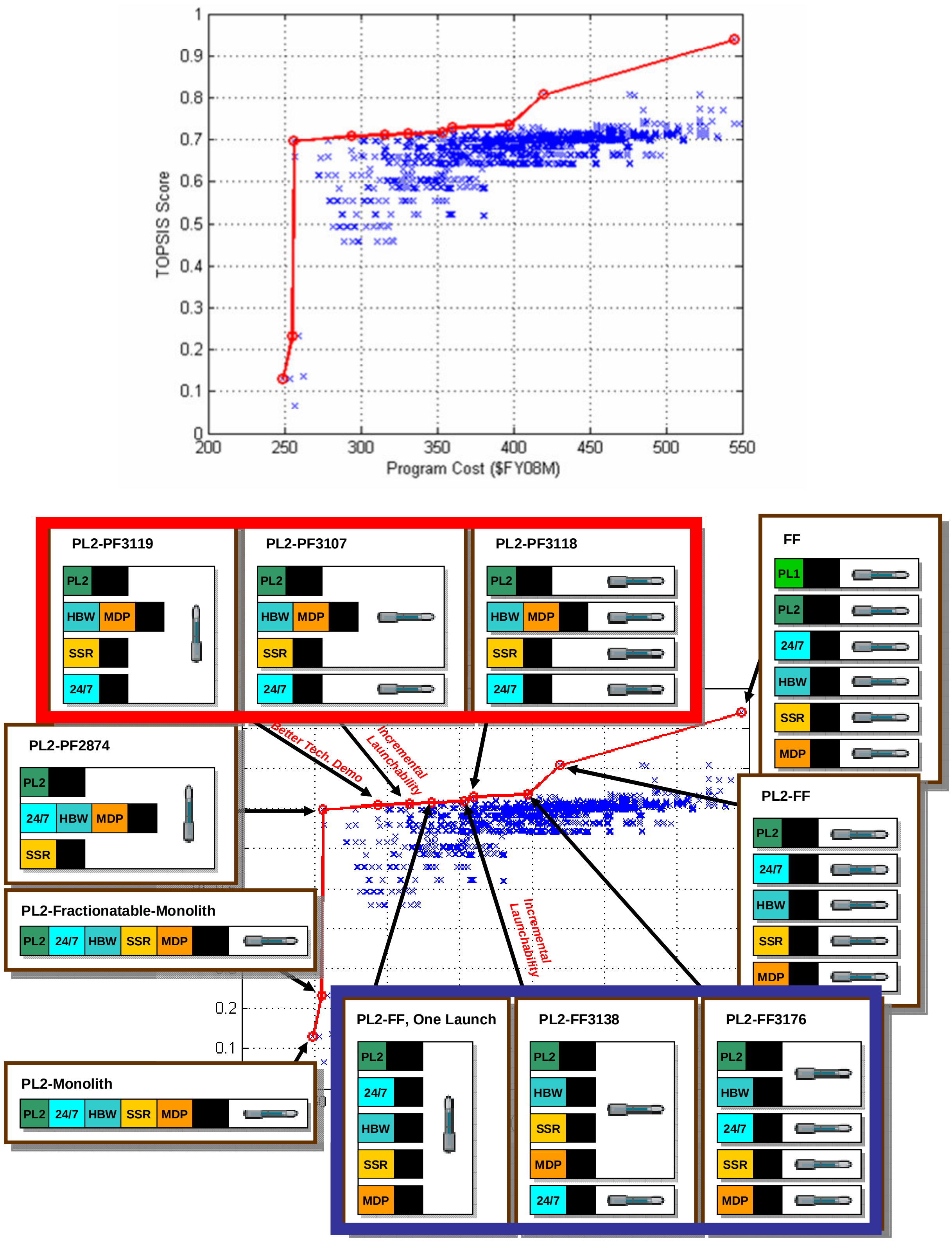

Figure 29. Pareto front between TOPSIS Score and Program Cost (O_4). *Blue marks denote individual designs, the red line indicates the Pareto front, and red text indicates reasons for increases in the TOPSIS score for particular designs as one moves to higher program costs.*

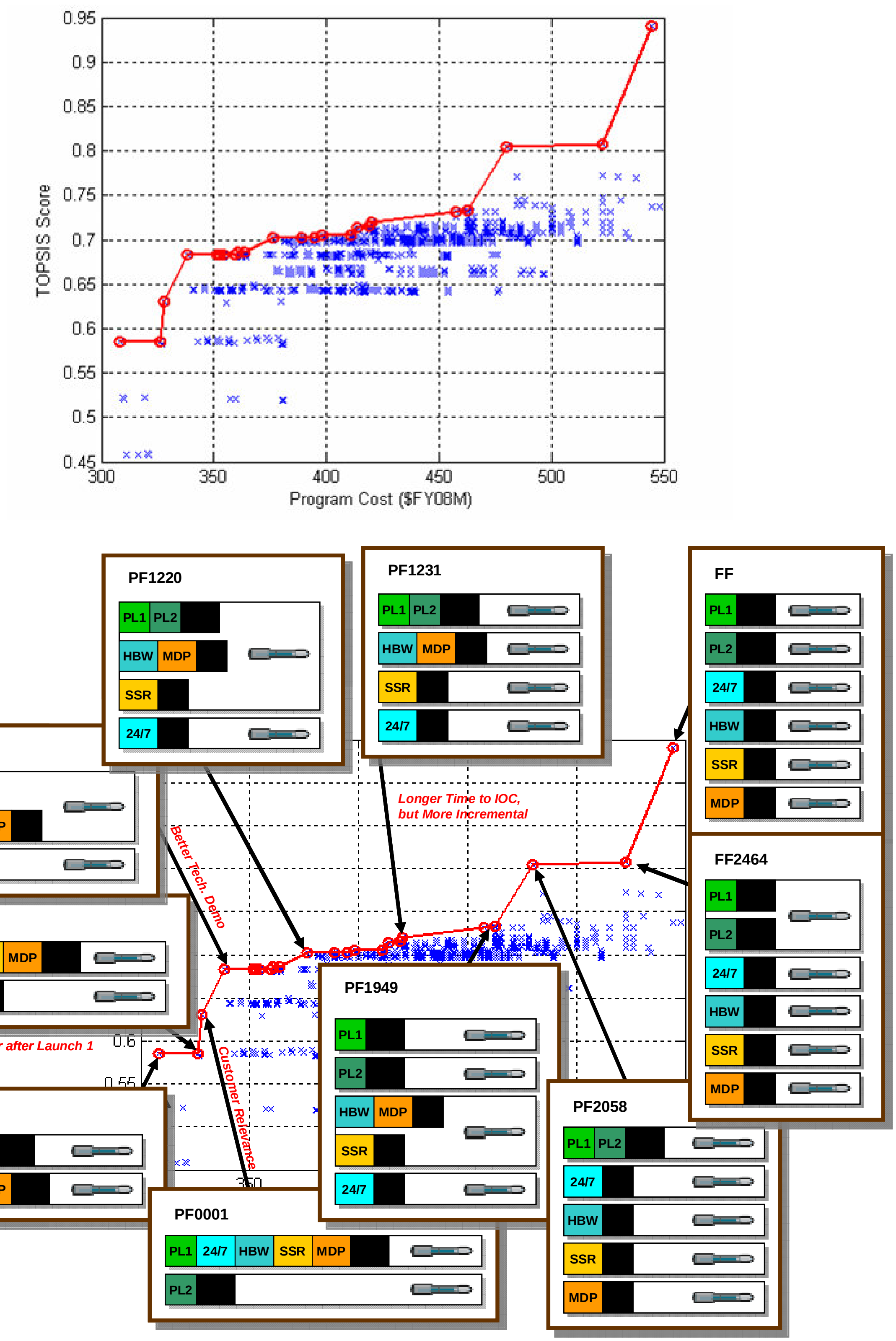

Figure 30. Pareto front between TOPSIS Score and Program Cost (O_4), excluding single-payload and single-launch cases. *Blue marks denote individual designs, the red line indicates the Pareto front, and red text indicates reasons for increases in the TOPSIS score for particular designs as one moves to higher program costs.*

IX. Summary and Conclusion of Part 2

Part 2 of this paper has presented the application of the Georgia Tech F6 Architecture Synthesis Tool (GT-FAST) to the exploration of the System F6 trade space. Combinatorial analysis of the architectural trade space itself was presented, providing a theoretical contribution applicable to future analyses and clearly showing the explosion of the size of the trade space as the number of fractionatable components increases. Several output metrics of interest to this study were defined, and next Pareto fronts were used to visualize the trade space. The first set of these Pareto fronts allowed direct visualization of one output against another, and the second set presented cost on the horizontal axis and on the vertical axis a TOPSIS score aggregating the performance characteristics of 15 of the 16 objectives. These techniques allowed for the identification of a handful of Pareto-optimal designs from an original pool of 3,190 designs.

A. Selecting a Design

Some of the most interesting and practical conclusions are derived from the Pareto fronts in Figs. 29 and 30. In both of these figures, there are large gains in TOPSIS score (which can be considered a performance or effectiveness metric) associated with small increase in program cost above the minimum-cost design. This is especially evident in Fig. 29. Following this initially steep rise is a shallow-slope region extending for $100-150 million before the Pareto front slope becomes steeper again as the complex and costly fully fractionated designs are approached. In this shallow-slope region, complex interactions exist among the various low-priority objectives to make some designs marginally more preferable than others; in this region, many designs have similar overall performance characteristics and designs are thus more difficult to distinguish on this basis, particularly considering that the objective weightings (see Table 12) are likely to be at least moderately dependent on the customer.

Perhaps one of the more appealing designs is PF0248 at a knee of the Pareto front in Fig. 30 (a design which is also related to the PL2-PF2874 design at the knee in Fig. 29). PF0248 is a two-launch design with the significant advantage of a module dedicated only to carrying payloads, meaning it performs well in the heavily-weighted category of relevance to potential fractionation customers. PF0248 also has average performance in several other categories and, importantly, has one of the lowest program costs.

Additionally, regardless of decision-maker preferences for any particular design, this analysis has captured the competing effects associated with increasing the number of launches for a particular architecture. While the additional launches are desirable in order to demonstrate the ability to field the architecture incrementally and on launches that are order-independent, they also increase the time it takes to finish fielding the system as well as increase cost. This analysis has also captured the potential advantage to dedicating one or more launches to modules containing components nonessential to initial operational capability, such as a second payload or 24/7 communications unit. A strategy such as this allows initial operational capability to be reached in a single launch (for example, if all essential components are on the first launch) while also providing other launches as opportunities to demonstrate the ability to add infrastructure components or payloads. However, this analysis has also revealed that decisions such as this must be traded against potential cost and flexibility implications; in the case of PF0248, had the 24/7 communication unit been placed in a dedicated module instead of the SSR, the total program cost would have been $50 million higher and the average cost of adding a component would have been $7 million higher.

B. Potential Future Work

A number of avenues for future follow-on work exist. First, the combinatorial analysis presented here is not easily applicable to scenarios with complex constraints (the $D_6 + D_5 + D_5$ scenario here was relatively simple), and it also overcounts the number of possible designs if two components are not distinct (e.g., if there were to be two indistinguishable SSRs in a cluster). Second, regardless of whether a combinatorial theory is available to count the number of possible designs, the task still exists of how to efficiently enumerate them; the algorithm illustrated by Fig. 20 works quickly for relatively small values of N but quickly becomes time-consuming as N increases.

In terms of the trade space exploration process, a helpful next step would be the analysis of a constrained Super-SEA with $N > 6$ (perhaps at about $N = 9$) to allow consideration of redundant components. Along with this should come refinement of the objective weightings and an assessment of the sensitivity of the TOPSIS-based Pareto fronts to these weightings (the pure two-objective Pareto fronts in Figs. 27 and 28 are of course unaffected). Another area of future work is the refinement of the objective metrics themselves and increased resolution on the objectives listed with “Coarse” or “Not Available” resolution in Table 12.

Overall, this paper has contributed theoretical and practical processes and results toward the systematization of future decision processes for fractionated architectures. All of the processes and theoretical developments here are applicable to future fractionated architecture analyses and decision processes, and the data generated here presents some relevant and interesting conclusions for the particular case of the System F6 demonstrator spacecraft. It is hoped that the ideas presented here find use with engineers and decision-makers considering fractionated systems in the future.

Acknowledgments

For their help in locating information on the Jason-2 and TIMED spacecraft, the author would like to thank Philippe Landiech at CNES, David Kusnierkiewicz at APL, Alan Buis and Stephen Cole at NASA, and John Christian at the University of Texas at Austin. Thanks are also due to James Wertz from Microcosm Space Mission Engineering and Kathy Tomajko from Georgia Tech. The author would also like to acknowledge the support of the Department of Defense through the National Defense Science and Engineering Graduate (NDSEG) Fellowship. Finally, and most importantly, the author thanks Joseph Saleh, without whom this contribution to the F6 program would never have happened.

References

[1]"Broad Agency Announcement: System F6," DARPA Tactical Technology Office, BAA07-31, 16 July 2007, URL: http://www.darpa.mil/TTO/solicit/BAA07-31/F6_BAA_Final_07-16-07.doc [cited 21 Nov. 2008].

[2]Brown, O. and Eremenko, P., "Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft." AIAA 2008-7869. AIAA Space 2008 Conference and Exposition, San Diego, 9-11 Sept. 2008.

[3]Hitch, C.J., *Decision-Making for Defense,* University of California Press, Berkeley, 1965, Chap. 3.

[4]LoBosco, D.M., Cameron, G.E., Golding, R.A., and Wong, T.M., "The Pleiades Fractionated Space System Architecture and the Future of National Security Space." AIAA 2008-7687. AIAA Space 2008 Conference and Exposition, San Diego, 9-11 Sept. 2008.

[5]McCullers, L.A., "Aircraft Configuration Optimization Including Optimized Flight Profiles, Multidisciplinary Analysis and Optimization, Part I." NASA CP-2327, 1984.

[6]O'Neil, D.A. and Mankins, J.C., "The Advanced Technology Lifecycle Analysis System (ATLAS)." IAC-04-IAA.3.6.3.01. 55th International Astronautical Congress, Vancouver, 4-8 Oct. 2004.

[7]Thompson, R.W., "System Architecture Modeling for Technology Portfolio Management using ATLAS." IEEE Aerospace Conference, Big Sky, 4-11 March 2006.

[8]Kipp, D.M., Dec, J.A., Wells, G.W., and Braun, R.D., "Development of a Planetary Entry Systems Synthesis Tool and Conceptual Design and Analysis." 3rd International Planetary Probe Workshop, Athens, 27 June – 1 July 2005.

[9]Alemany, K., Wells, G., Theisinger, J., Clark, I., and Braun, R., "Mars Entry, Descent, and Landing Parametric Sizing and Design Space Visualization Trades." AIAA 2006-6022. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, 21-24 Aug. 2006.

[10]Komar, D.R., Hoffman, J., Olds, A., and Seal, M., "Framework for the Parametric System Modeling of Space Exploration Architectures." AIAA 2008-7845. AIAA Space 2008 Conference and Exposition, San Diego, 9-11 Sept. 2008.

[11]Crocker, A.M., Charania, A.C., and Olds, J.R., "An Introduction to the ROSETTA Modeling Process for Advanced Space Transportation Technology Investment." AIAA 2001-4625. AIAA Space 2001 Conference and Exposition, Albuquerque, 28-30 Aug. 2001.

[12]Myers, R.H. and Montgomery, D.C., *Response Surface Methodology*, Wiley, New York, 2002.

[13]Mathieu, C. and Weigel, A.L., "Assessing the Fractionated Spacecraft Concept." AIAA 2006-7212. Space 2006, 19-21 Sept. 2006.

[14]Larson, W.J., and Wertz, J.R. (ed.), *Space Mission Analysis and Design*, 3rd ed., Microcosm Press and Kluwer Academic Publishers, El Segundo, 1999.

[15]Mahr, E.M., "Small Satellite Cost Model 2007 (SSCM07) User's Manual." ATR-2007(8617)-5. The Aerospace Corporation, 30 Sept. 2007.

[16]NASA Goddard Space Flight Center, *National Space Science Data Center Spacecraft Query* [online database]. URL: http://nssdc.gsfc.nasa.gov/nmc/SpacecraftQuery.jsp [13 Feb. 2009].

[17]Martin, D.H., Anderson, P.R., and Bartamian, L., *Communication Satellites*, 5th ed., The Aerospace Press, El Segundo, 2007.

[18]Mortenson, M.E., *Geometric Modeling*, John Wiley & Sons, New York, 1985.

[19]NASA Johnson Space Center, Mission Operations Cost Model, Online Software, 25 May 2007, URL: http://cost.jsc.nasa.gov/MOCM.html [8 May 2009].

[20]Fleming, E.S., Lafleur, J.M., and Saleh, J.H., "Response Surface Equations for Expendable Launch Vehicle Payload Capability." AIAA 2009-6656. AIAA 2009 Conference and Exposition, Pasadena, 14-17 Sept. 2009.

[21]Isakowitz, S.J., Hopkins, J.P., and Hopkins, J.B., *International Reference Guide to Space Launch Systems*, 3rd ed., AIAA, Reston, 1999.

[22]Brown, O., "Common Launch Vehicle Data for Econometric Analysis." DARPA/TTO Memorandum, 10 April 2008.

[23]Cole, S., Buis, A., et. al., "Ocean Surface Topography Mission / Jason 2 Launch." Press Kit, NASA, June 2008. URL: http://www.nasa.gov/pdf/228260main_jason-2.pdf [10 May 2009].

[24]Terrenoire, P., Douillet, F., Carucci, A., et. al., "Jason-2: Consolidating and Enlarging the Jason-1 Performance in Ocean Altimetry." IAC-03-B.2.07. 54th International Astronautical Congress, Bremen, 29 Sept. – 3 Oct. 2003.

[25]Brown, D., Neal, N., and Marren, K., "TIMED: Thermosphere, Ionosphere, Mesosphere, Energetics and Dynamics." Mission Guide, NASA and Johns Hopkins University Applied Physics Laboratory, December 2006. URL: http://www.timed.jhuapl.edu/WWW/common/content/pdfs/missionGuide.pdf [10 May 2009].

[26]Kusnierkiewicz, D.Y., "An Overview of the TIMED Spacecraft." *Johns Hopkins APL Technical Digest*, Vol. 24, No. 2, 2003, pp. 150-155.

[27]de Selding, P.B., "Data Continuity May Be Affected by Delays in Jason-2 Development." *Space News*, 31 May 2005. URL: http://www.space.com/spacenews/archive05/Jason_053005.html [10 May 2009].

[28]Yee, J., Cameron, G.E., and Kusnierkiewicz, D.Y., "An Overview of TIMED." SPIE Conference on Optical Spectroscopic Techniques and Instrumentation for Atmospheric and Space Research III, Denver, July 1999, SPIE Vol. 3756, pp. 244-254.

[29]Brown, O., "Program of Record Analysis." DARPA/TTO Memorandum, 6 May 2008.

[30]Grimaldi, R.P., *Discrete and Combinatorial Mathematics: An Applied Introduction*, 5th ed., Pearson, Boston, 2004.

[31]Kirby, M.R. and Mavris, D.N., "A Method for Technology Selection Based on Benefit, Available Schedule and Budget Resources." AIAA 2000-5563. 2000 World Aviation Conference, San Diego, 10-12 Oct. 2000.