

Technology Portfolio Planning by Weighted Graph Analysis of System Architectures

Peter Davison* and Bruce Cameron†

Massachusetts Institute of Technology, Cambridge, MA 02139

Edward F. Crawley‡

Skolkovo Institute of Science and Technology, Skolkovo 143025, Russia

Abstract

5 Many systems undergo significant architecture-level change throughout their lifecycles in order to adapt to new operating and funding contexts, to react to failed technology development, or to incorporate new technologies. In all cases early architecture selection and technology investment decisions will constrain the system to certain regions of the tradespace, which can limit the evolvability of the system and its robustness to exogenous changes. In this paper we present a method for charting development pathways within a tradespace of potential architectures, with a view to enabling robustness to technology portfolio realization and later architectural changes. The tradespace is first transformed into a weighted, directed graph of architecture nodes with connectivity determined by relationships between technology portfolios and functional architecture. The tradespace exploration problem is then restated as a shortest path problem through this graph. This method is applied to the tradespace of in-space transportation architectures for missions to Mars, finding that knowledge of pathways through the tradespace can identify negative coupling between functional architectures and particular technologies, as well as identify ways to prioritize future technology investments.

1 Introduction

20 Choosing a technology portfolio to invest in for a complex system places constraints on the architecture of the system. The architecture consists of a relatively small number of decisions that determine the envelope of functionality of the system, as well as the decomposition of the high level system elements [Crawley et al., 2004]. Technology investment decisions are typically made with an uncertain reference architecture in mind with the intent of demonstrating the technology or raising the technology readiness level before system development. However if either the anticipated technology portfolio or reference architecture changes once these decisions are made, the options available to the system designer may be severely restricted.

25 We assert that it would be desirable to make technology investment decisions that recognize the inherent coupling to architecture decisions. Where possible, the technologies chosen should enable architectural flexibility in the event that project requirements change, and the architecture chosen should be robust to technology demonstration failure, in that a new architecture should be available that uses many of the same systems minus the failed technology. Exogenous changes (by which we mean changes that occur outside of the technical system) that induce changes in the architecture such as technology failure, changes to the project budget, stakeholder demands for system extension, or the development of competing systems are common in space systems and other complex systems applications [Mehr and Tumer, 2006]. In such systems the cost of making large, unplanned architecture changes is often high, resulting in schedule and cost overruns, suboptimal designs, or project cancellation. This high impact highlights the need to make architecture and technology portfolio decisions with system flexibility and design robustness in mind.

*Research Assistant, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue Rm 33-409, Cambridge, MA 02139

†Lecturer, Engineering Systems Division, 77 Massachusetts Avenue 31-161B, Cambridge, MA 02139

‡President, Skolkovo Institute of Science and Technology, ul. Novaya d. 100, Skolkovo 143025 Russia

We showcase a method for analyzing the coupling between these two types of decisions to find development pathways of a system through a tradespace of possible options: a process we call tradespace exploration. In essence we identify neighboring architectures with similar technology portfolios to find the best way to move around the tradespace. The key to understanding these pathways through the tradespace is that they do not necessarily represent the set of sequential changes that a decision maker would actually make to transform the architecture, rather they represent the smallest incremental developments that *could* be made from one end of the path to the other. The value of these incremental paths in terms of assessing system development flexibility and robustness lays in their ability to decompose large system changes into intermediate steps.

A recent example of this class of problem can be found in the major restructuring of NASA’s human spaceflight program with the cancellation of Constellation and the introduction of the Space Launch System (SLS) and Multi Purpose Crew Vehicle (MPCV) [U.S. Senate, 2010; NASA, 2011]. Following a restructuring of strategic goals and with a Congressional mandate to “utilize existing contracts, investments...and capabilities from the Space Shuttle and Orion and Ares I projects” [U.S. Senate, 2010], NASA decision makers had to select a new system architecture that met stakeholder needs while making use of as much of the previous project architecture as possible. In addition, such an architecture was required to “incorporate capabilities for evolutionary growth” [U.S. Senate, 2010] to higher performance missions along an uncertain timeline. Choosing such an architecture and set of investments that both make use of existing assets and allow for flexible performance evolution is a challenging and highly uncertain problem.

Figure 1 shows a generic representation of this tradespace exploration problem faced by NASA and many other large organizations managing complex systems. A starting system architecture A and several candidate architectures are plotted along simple metrics. Connections between architectures represent the presence of a feasible development path between two architectures.

Consider one scenario in which a decision maker can only move across a single connection for each iteration of the system. Such a constraint might arise if the amount of time between product iterations or an annual development budget constraint restricts the magnitude of system development that can occur at each iteration. In such a scenario, if the decision maker wishes to optimize performance in the long run, a local optimization approach would yield the suboptimal path $A-B-D$. Only by understanding the global connectivity of the tradespace could the optimal path $A-C-F-G$ be found.

As a second scenario consider the case in which the decision maker is not constrained to move across only one edge per iteration, but can ‘jump’ to any architecture in the tradespace. If he or she again is concerned with performance only, then the architecture will be transformed directly from A to G . However we argue that knowledge of the intermediate architectures along the incremental path - C and F - still delivers significant value to the decision maker. For instance, if during the transformation of the system to G a budget reduction makes G infeasible, an awareness of other feasible architectures on the development path is crucial. In the same way if G requires a number of technology investments to be made, knowledge of the intermediate steps and the investments they require allows these technologies to be prioritized so that contingency architectures are available should a technology fail to mature.

Both of these conclusions in the context of Figure 1 are trivial, however when the number of architectures and connections increases, this analysis becomes too complex for a human to perform. The goal of this paper is to model how these connections between architectures are determined, how this process of tradespace exploration can be performed computationally, and how the decision maker might use this information under a number of different scenarios.

The core of our approach is to formulate the tradespace as a directed graph, and then to restate the tradespace exploration problem as a shortest path problem through the graph. By building a computational tool that can rigorously search a tradespace of several thousand architectures for patterns that a human cannot discern, we hope to provide a method for the decision maker to augment his or her detailed understanding of complex social and technical issues with a broad analysis of the high level design trades.

The specific application that has motivated this research is the exploration of possible architectures for the in-space transportation infrastructure for manned planetary exploration. Starting from the results of a tradespace enumeration tool [Rudat et al., 2012] that produces tens of thousands of feasible architectures, our goal is to understand which of those are the best candidate architectures for a set mission and how those architectures along with the technology portfolio should evolve as the context of the system changes.

2 Literature Review

Technology Portfolio Selection

The dual problems of selecting a technology portfolio and selecting a system architecture have been studied extensively in the literature as two separate, decoupled issues. The standard approach for NASA and many other organizations responsible for the management of complex technology portfolios is to create broad technology roadmaps based on the input of subject matter experts and current organizational priorities; see for example [Steering Committee for NASA Technology Roadmaps and National Research Council, 2012; Office of the USAF Chief Scientist, 2010; Eder and Linde, 2011]. These broad technologies are then assigned to specific projects, with the goal of increasing the maturity of a particular technology either by performing basic research or implementing the technology in a functioning system. The problem of allocating resources to the technology development process has been shown to be highly uncertain since the cost of maturing a technology along with its final impact on system performance is difficult to predict [Hawes and Duffey, 2008; Szajnfarber et al., 2011]. Technology development for large government portfolios adds additional layers of complexity and uncertainty to the process [Szajnfarber et al., 2011].

Heidenberger and Stummer [1999] describe several methods to quantitatively evaluate the process of selecting and allocating resources to technology projects. They divide these approaches into six categories: benefit measurement techniques, mathematical programming, decision and game theory models, simulation models, heuristic methods, and cognitive emulation. A commonly used method is real options analysis, which has been used to evaluate investment decisions for NASA [Hawes and Duffey, 2008; Shishko et al., 2004; Saleh et al., 2002] as well as for information technology project investments [Benaroch and Kauffman, 1999] and automotive industry research and development [Avadikyan and Llerena, 2010] among many other applications. Other technology portfolio decision aids studied in the literature include multi-attribute utility theory [Mehrez, 1988], financial portfolio theory [Kumar et al., 2008], and knowledge-based expert systems [Liberatore and Stylianou, 1993].

Whereas many of these methods calculate technology impact using a baseline architecture or parametrically defined model of a project portfolio, Battat et al. [2013] take a different approach by considering the effect of a particular technology on a large number of architectures spanning the tradespace, a perspective that is useful if no baseline architecture exists or the set of projects to which the technology will be applied is highly uncertain.

System Architecture Models

Since the concept of an architecture is by nature an abstract one, there exists a number of general models that can be applied in a wide variety of domains. The design structure matrix (DSM) [Browning, 2001] is a flexible tool used to model the relationships between different system elements and system functions using a matrix representation. Object Process Methodology (OPM) [Dori, 2002] is a graphical modeling language that describes the relationships between system entities and functions, and if organized correctly can provide an elegant view of a complex system.

Other specialized architecture models have been developed to aid in the rapid enumeration of architectures across an entire tradespace. Architectures modeled as sets of decisions assert that each aspect of the architecture can be traced to a decision among multiple options, with each decision narrowing the design space until a complete architecture is reached [Covaliu and Oliver, 1995; Simmons, 2008]. A path through a network or graph has also been used to model an architecture. Example include the Object Process Network Tool [Koo et al., 2009] which focuses on a functional view of the architecture, and a rule-based graph representation which is developed around the transfer of system elements between steady states [Arney, 2012].

Simple mathematical structures can also be used to represent an architecture. A partition, which groups a finite set of entities into larger chunks, is used by Rudat et al. [2012] to define an architecture as the assignment of major system functions to elements of form. Architectures are enumerated by taking all possible function partitions, checking feasibility using logic constraints, and using a parametric solver to calculate output metrics

Perhaps the most widely used architecture representation is a design vector model, which relates input design parameters to output metrics such as performance through scientific or engineering models of con-

stituent systems. At one end of the spectrum are deep point designs such as that presented in NASA's Design Reference Architecture 5.0 Mars Exploration study [2009], which enumerates a single architecture to a high degree of detail using an expert-based system model. At the opposite end are design models that allow outputs to be automatically calculated from design parameters, which themselves can be highly complex as in much of the Multidisciplinary Design Optimization (MDO) literature [Martins and Lambe, 2012] or can consist of simpler high level relationships to more exhaustively enumerate diverse architectures [de Weck et al., 2004].

Architecture Tradespace Exploration

Once a tradespace has been defined, the next challenge often faced is how to find the optimal architecture or set of architectures, which is often termed tradespace exploration. One approach is to start from a baseline architecture and to make incremental changes to find the global optimum using an Algebra of Systems [Koo et al., 2009] or Ant Colony Optimization algorithm [Arney, 2012]. Similarly a Genetic Algorithm (GA), an optimization tool modeled after natural selection, can be used to search the tradespace [Brown and Thomas, 1998; Singh and Dagli, 2009]. Many other nonlinear programming algorithms are implemented in MDO methods to perform this searching function [Martins and Lambe, 2012].

In cases where all architectures in the tradespace are enumerated, multiple evaluation metrics can be calculated for each architecture, so rather than defining a single best architecture, the set of non-dominated architectures is found and analyzed further [Rudat et al., 2012; McManus and Warmkessel, 2004]. Ross [2005] presents a review of several tradespace exploration methods that expand this basic Pareto front analysis to take into account performance or cost uncertainty, changing system requirements, and emergent system properties such as flexibility and robustness. Similarly, Walton [2004] develops a framework to supplement the classic decision making metrics - cost and performance - with uncertainty to find groups of architectures that can be pursued concurrently in the concept exploration phase.

If an optimal architecture exists, exogenous changes may introduce a disruption. Maher and Poon study this phenomenon by allowing both the solution space and the problem statement to 'co-evolve' using a GA [Maher and Poon, 1996]. Haris and Dagli [2011] take a similar evolutionary approach, by explicitly changing system requirement once an optimal architecture has been found and using a GA to adapt the old architecture.

Several other studies analyze the problem of evolving or extending an architecture after deployment. De Weck, de Neufville, and Chaize [2004] study the staged deployment of a constellation of communications satellites in response to uncertain demand. A real options approach is used to select an initial architecture with modest capacity that can later be supplemented with additional assets to respond to increases in demand. A retrospective analysis of architecture development is performed by Nakamura and Basili [2005] to chart the evolution of software versions from initial deployment to its final release, with a focus on the magnitude of change made at each intermediate step rather than on changes in performance. Arney [2012] analyzes how systems within a particular architecture might be reused between missions to different destinations, and then uses this analysis to assess the viability of a 'flexible path' through the multiple destinations.

Silver and de Weck [2007] develop a quantitative tool to perform automatic exploration of development pathways through a design space, which they call Time-Expanded Decision Networks. Starting with a tradespace of architectures and a detailed cost model of system development, operation, and retirement, a dynamic network is formulated which encodes the possible sequences of active architectures over several time steps. The problem of finding the progression of architectures among the possible options for a given demand for is then restated as a shortest path problem through the network. The effect of altering the switching costs of moving between any two architectures is also examined as a way to introduce additional flexibility into the tradespace [Silver and de Weck, 2007].

Architecture Distance

In many of these methods examining the development or evolution of an architecture, some concept of distance between architectures is required. Software architectures can be represented as graphs of system elements with edges connecting interfacing elements, with a distance metric defined that measures the number of similar elements and connections over all the different substructures [Nakamura and Basili, 2005]. The

number of common systems in a space exploration context also can be used as a measure of distance [Arney, 2012].

The *delta* DSM is a matrix representation of the components and relationships that change between two architectures encoded as a DSM [Smaling and de Weck, 2007]. The switching cost defined by de Weck and Silver [2007] is a much more labor intensive definition of architecture distance, since the cost of decommissioning retiring systems and developing new systems must be estimated from the bottom up for each architecture pair.

For cases where the architecture is modeled as an abstract entity such as a partition (which will be used to model architecture in this paper), it is more difficult to parametrically estimate the cost of moving from one architecture to another. However, the literature regarding metrics on partition spaces is quite mature. A set of partition metrics is presented by Simovici and Jaroszewicz [2002] using the concept of generalized entropy between all pairwise combinations of partition blocks. Using a much simpler approach, Day [1981] characterizes six fundamental transformations for partitions on discrete sets - removal, augmentation, mutation, division, merge, and transfer - that are used to define several simple metrics on a partition space.

3 Problem Framing

Problem Vocabulary

Before introducing the tradespace exploration problem in more detail, we first define a number of terms that will be used throughout this paper to clarify our approach.

An *architecture* is a description of a specific system solution that solves the problem posed by the customer. For complex systems, the architecture describes the high-level mapping of function to form and the relationship between major system elements. Any two architectures are separated by a *distance*, which measures the number of changes that must be made to transform one architecture into the other. The distance serves as a proxy for the cost of switching between two architectures. The *tradespace* of architectures is the collection of all the architectures under evaluation that could meet the requirements of the customer.

An *evolution* of architectures is a sequence of architectures that characterizes the transformation of one architecture into another. An evolution represents an incremental development path from one point in the tradespace to another. Crucially, an evolution is not necessarily the sequence of architecture changes that should be made in actual system development, since often it is more efficient to have a few major system iterations than many small ones. The evolution simply decomposes these major iterations down into their atomic steps to maximize the amount of information available to the decision maker. Finding the optimal evolution of architectures in different contexts is a major goal of this work.

Each element in the evolution is the *active architecture* for that step, while other architectures in the tradespace are *candidate architectures*. The *initial* and *final* architectures characterize the active architectures at the beginning and end of the evolution respectively.

A *technology* is a broadly applicable capability such as (in the context of space systems) nuclear thermal propulsion or high-efficiency solar cells that must be developed through investments in research. Some technologies are required by certain architectures in order for the architecture to function, while other technologies change the performance of an architecture but are not strictly necessary. The *technology portfolio* consists of the technologies that will be available to be deployed in the architecture. The *progression* of the portfolio is a sequence of the technology portfolios analogous to the evolution of architectures.

Problem Scope and Assumptions

We choose to frame the tradespace exploration problem as one of finding the optimal evolution of a system's architecture and progression of its technology portfolio through an unorganized tradespace. The purpose of finding these sequences is to provide the decision maker with the best options for: i) maintaining flexibility in the architecture in the concept selection phase of the project; ii) mitigating the cost of changing the architecture when one or more exogenous factors makes the current architecture infeasible or sub-optimal; or iii) developing new, planned iterations of the system over time to improve system performance or reduce recurring cost.

It is important to note that several key assumptions have been made in order to reduce the complexity of the tradespace exploration problem. First, although the development of a system from an initial state to a goal state would presumably take place on a temporal scale, the concept of time and along with it the time value of assets has been deliberately left out of our analysis since in many cases the amount of time between events is highly uncertain. Additionally, by isolating steps through the tradespace from instants in time, we can analyze all of the potential architecture and technology portfolio increments along a path rather than just those that would represent the physical manifestation of the system over time.

Second, we make the assumption that the development of technologies is deterministic. In other words, we assume that when a project invests in a technology it becomes available to be incorporated into the architecture. Viewing technologies as ‘switches’ rather than probabilistic events allows us to more closely examine the coupling between architecture and technology at the expense of abstracting the risk inherent in technology development.

A final assumption is that technology development and architecture development can be decomposed into two distinct processes. This decomposition allows us to isolate architectural change from technological change and to analyze the effects of each type of change separately.

Notation

Let the indexed variable a_i represent a distinct architecture defined during the enumeration of the tradespace. Note that a_i can be encoded in any way we choose (e.g. function to form mapping, form DSM, decision sequence, etc.). The whole tradespace is the set A of N distinct architectures

$$A = \{a_1, a_2, \dots, a_N\}.$$

We denote the distance between two architectures as

$$d_{i,j} = d(a_i, a_j).$$

The evolution of architectures is the ordered sequence:

$$E = \{e_1, e_2, \dots, e_D\} \quad e_i \in A,$$

where D is the number of architectures in the evolution, and is not known *a priori*. Each element of E is an architecture within the tradespace, with the ordering corresponding to the sequence of the evolution such that e_1 is the initial architecture and e_D is the final architecture. Note once again that the elements of E do not necessarily all represent recommended physical iterations of the system architecture, rather they describe the *incremental* pathway through the tradespace.

For M possible technologies, which are defined as part of the tradespace, let the variable t_i , $i = 1, \dots, M$ represent the availability of the i^{th} technology. For our purposes t_i can either take on the value 1 or 0.

It is important to note that the availability of a technology may change from step to step as we move through the tradespace, so we will use $t_i^{(k)}$ to represent the availability of technology i at progression step k . The technology portfolio, $T^{(k)}$, is the vector of all technology availabilities at step k :

$$T^{(k)} = [t_1^{(k)}, t_2^{(k)}, \dots, t_M^{(k)}].$$

The technology progression \mathbf{T} is the $D \times M$ matrix

$$\mathbf{T} = \begin{bmatrix} T^{(1)} \\ T^{(2)} \\ \dots \\ T^{(D)} \end{bmatrix}.$$

Finally, let us define a performance metric for an architecture that we can use to objectively compare architectures. In general, performance will also depend on the available technologies, so let the scalar $P(a_i, T)$ be the performance of architecture a_i paired with technology portfolio T .

Problem Statement

Our precise statement of the tradespace exploration problem is motivated by Henderson and Clark’s [1990] framework for defining innovation. Henderson and Clark characterize a system along two dimensions: core concepts, which we call technologies; and linkages between concepts and components, which we call the system architecture. If neither of these system dimensions change, only *incremental innovation* can be realized. If the core concepts change while the linkages remain constant (technological change without architecture change) they term this change *modular innovation*. If the opposite is true, technology is constant across a changing architecture, they use the term *architectural innovation*. Finally if both core concepts and linkages are in flux there is *radical innovation*.

Although this framework is highly abstract, it motivates us to study the tradespace exploration problem as three separate problems according to what type of innovation - modular, architectural, or radical - is being analyzed. The only change we make to this framework is that in the modular innovation case, we allow the architecture to vary within its immediate family, which is determined by the architecture distance metric. The three tradespace exploration problems are as follows:

1. For a fixed family of architectures such that $d(e_1, e_k) = 0 \quad \forall k = 2, \dots, D$, find the optimal evolution of architectures E and progression of technologies \mathbf{T} .
2. For a fixed technology portfolio $T^{(k)} = T^{(1)} \quad \forall k = 2, \dots, D$, find the optimal evolution of architectures E .
3. For a variable technology portfolio and architecture, find the optimal progression of technologies \mathbf{T} and evolution of architectures E .

The statements above only broadly categorize the type of tradespace exploration problem being analyzed; much more information is needed to fully define the problem for a specific application, as we will demonstrate for Problems 1 and 2 in Section 5. It is also important to note that we leave the exact definition of optimality open to interpretation, although we do present our own definition for the application studied in the following sections.

Each of these problems corresponds to different contexts in which a decision maker would wish to discover possible pathways through the tradespace. For Problem 1, the system might have a mature, stable architecture when demands for improved performance or planned system evolution allows additional technology investments to be made and incorporated, without major changes made to the architecture. One example from the space systems domain is the modernization of the Global Positioning System over the last two decades, with improved communications and processing technologies incorporated in all three segments without changing the underlying architecture [National Research Council, 1995].

A sample context for Problem 2 is a system with an active architecture in development optimized for a particular technology portfolio. If this technology portfolio is externally disrupted or the architecture otherwise becomes infeasible, understanding how to modify the architecture without being able to add new technologies would be valuable to decision makers. The transition from Constellation to Orion could be thought of in this context, as decision makers had to create a new architecture using the same set of broad technologies, with tight constraints on heritage asset reuse [NASA, 2011].

Problem 3 is a context applicable to decision makers looking towards long term planning and system evolution through multiple technology cycles. Here a decision maker would want to know what architecture changes need to be made at different stages to enable prospective technologies to be incorporated to maximum effect, and what initial architecture and technology decisions should be made to provide a number of flexible responses to uncertain future developments.

4 General Solution Approach

With the problem stated in detail and the notation explained we now move to the framework we have developed to solve the tradespace exploration problems posed. Our approach is to consider the tradespace of architectures as a weighted, directed graph, and then to restate the tradespace exploration problems above as well-understood problems in graph theory.

Starting with a fully enumerated and evaluated tradespace of point architectures, we transform this tradespace into a directed graph. An initial node is selected to represent the starting architecture and technology portfolio, followed by the selection of the final node that represents the desired system state. Finally the optimal shortest path between these nodes is determined.

5 Graph Generation

Like any weighted graph, our tradespace graph consists of nodes connected by edges with a defined weight. For our tradespace graph a node v is defined as an architecture paired with a technology portfolio.

$$v_i = (a_m, T_n)$$

where i , m and n are arbitrary indices within the set of possible nodes, architectures, and technology portfolios respectively. It is important to remember that our definition of a technology portfolio is the set of technologies *available* to the project, not just the technologies required by the architecture. Therefore there are many nodes that may share the same architecture, but whose technology portfolios will differ because they contain surplus technologies in addition to the architecture's required technologies.

An edge in the graph represents a feasible incremental development to transform one architecture into another. The generation of edges and the assignment of their weight varies for each of the three problems listed above.

For Problem 1 (fixed architecture family), the presence of an edge is determined by the relationship between technology portfolios. If the portfolios differ by a single technology, there is a directed edge from the node without the technology to the node with the technology. If the portfolios are identical there is an edge in both directions. If an edge exists its weight is equal to the cost of developing the technology that is added along the edge.

For Problem 2 (fixed technology portfolio), an edge exists between two architectures if the distance d between them is less than some cutoff value δ , with the weight equal to the value of d between them. Note that this edge can be traversed in either direction.

The approach for Problem 3 is a combination of the other two. The edge existence criterion is the same as that for Problem 1. Edge weight is equal to a weighted sum of the distance d between the two architectures and the cost of developing the technology added along the edge.

25 Final Node Selection

The selection of the initial node will be discussed further in Section 5 since it is sufficiently complex to warrant an application-specific explanation. Since the choice of final architecture and technology portfolio will be determined by a number of factors beyond the scope of this model, our strategy is to select a small number of final nodes that are objectively optimal within the modeling framework we have defined, and to allow the decision maker to trade among this small subset. These nodes lay along a Pareto front defined by the performance metric P and a cost metric that varies by application.

Shortest Path Algorithm

After the tradespace graph has been generated, the core of the approach is to implement a shortest path algorithm on the graph to find the optimal sequence of nodes connecting the initial node to the final node. The shortest path problem is a rigorously studied problem in graph theory for which a number of optimal and heuristic solutions exist [Cherkassky et al., 1996]. For the application presented in this paper, the size of the tradespace is small relative to the size of many graphs studied in graph theory and edge weights are non-negative so we implement an optimal algorithm known as Dijkstra's algorithm [Dijkstra, 1959] to solve the shortest path problem.

In most applications of Dijkstra's algorithm, it suffices to only know a single shortest path through the network. For the case of tradespace exploration however, two paths which have the same path length (i.e. the total edge weight traversed from initial to final) may not be the same in terms of performance of intermediate architectures. In other words, one pathway may stray from the Pareto front more than the other. For this reason, we must find all of the shortest paths through the network, if there exists more than one, and define a metric that takes into account performance along the path.

To find all shortest paths we make use of a variation of Yen’s algorithm [Yen, 1971] which successively deletes different edges along the initial shortest path to find other paths of equivalent length. While Yen’s algorithm finds the K shortest paths through the network, we make a slight modification so that it finds all shortest paths. As an evaluation criteria among these shortest paths we use the sum of the performance metric at each intermediate step, and select the path that optimizes this performance sum. In other words, when determining the optimal path between two nodes we find all paths of minimum length according to the sum of edge weight, and then select the path from this set with the highest performing intermediate architectures.

5 In-Space Transportation Infrastructure Application

In this section we describe the application of the general framework for tradespace exploration outlined in Section 4 to the tradespace of potential transportation infrastructures for manned space exploration. Results will be presented in Section 6. In this paper we will only analyze the tradespace in the context of Problems 1 and 2 (fixed architecture family and fixed technology portfolio respectively). Problem 3 has been left to future work as it requires the additional development of the relative weighting between architecture distance cost and technology development cost, which is beyond the scope of this initial application to the HEXANE tradespace.

Tradespace Description

The starting point for the application of this framework is a fully enumerated tradespace of distinct architectures. We make use of HEXANE [Rudat, 2013], a tool developed by Rudat et al., to populate this tradespace and evaluate architectures along simple metrics. HEXANE allows the user to specify an exploration destination (e.g. Mars surface, near earth object), mission duration and number of crew along with many other mission parameters and constraints, and outputs all of the architectures that could in theory realize this mission.

The core of the tool is the partition of system functions into elements of form. Rudat et al. [2012] define 17 high-level (seven habitation, ten propulsion) functions that must be executed by appropriate elements of form. For example any mission to a planet or body must conduct an Earth departure burn as well as a destination arrival burn and have a habitat available for crew during the deep space transit. The assignment of these functions to form encodes the *functional architecture*, a component of the total architecture. Other components of the architecture include the propellant used for each maneuver, the deployment timeline of different elements and the required technologies, though we use only the functional architecture when measuring the distance between two architectures.

After all feasible alternatives are enumerated, each architecture is evaluated to size each habitat and propulsion element using a parametric model and iterative solver. This evaluation gives an estimate for the initial mass to low earth orbit (IMLEO) necessary to close the design. Since mass to orbit is a simple, objective metric along which missions can be compared [Rudat et al., 2012], IMLEO is used as the performance metric, P .

Using this output of HEXANE, Battat et al. [2013] define 13 major technological capabilities, shown in Table 1, that are utilized within the tradespace. Each technology is scored to give it a relative Technology Development Cost (TDC), with the TDC of a portfolio equal to the sum of the scores of the technologies present. Each architecture is analyzed to determine which of these technologies must be available for the architecture to operate, which determines the architecture’s required technologies. Figure 2 shows a decomposition of the different components of the tradespace node that will be used throughout the following analysis.

For a more detailed description of the tradespace enumeration methodology and model assumptions we refer the reader to [Rudat et al., 2012; Rudat, 2013; Battat et al., 2013] and the references therein.

Distance Metric

Also unique to this application is the calculation of the distance metric used to specify the edge weight between neighboring nodes. Though there are a number of options available, the distance metric used for

this application is similar to the partition metric Day [1981] terms $B(Q, P)$.

The metric that Day defines is the minimum number of *divisions*, *mergences* and *transfers* needed to transform one partition into another. Although these three operations have rigorous definitions, their intuitive meaning is simple. A division is the creation of a new partition block (element of form) by splitting out one set object (function) into its own block. A mergence is the deletion of a partition block by merging a block that contains one object with another block. Finally a transfer is the removal of an object from one partition block combined with its addition to a different block.

In terms of our present application this metric counts the number of functions that must be moved between elements of form to make two functional architectures identical. We refer the reader to Figure 3 for a graphical representation of the metric.

This particular metric was chosen as a simple proxy for the cost of changing the functional architecture due to its simplicity and first order approximation of growth in technical complexity of the system. The other characteristics of the architecture were left out of the distance metric as their cost is already captured in the TDC metric (required technologies and propellant), or changing the characteristic affects the operation of the system much more than the development (deployment timeline).

The action of creating additional elements of form or adding functions to existing elements can be viewed as growth in both formal complexity as defined by Boothroyd and Dewhurst [1987] and structural complexity as defined by Sinha and de Weck [2013]. Alternatively this growth in complexity can be viewed as large, tightly coupled and disruptive engineering changes [Eckert et al., 2004]. When a function such as deep space habitation is transferred to a different element both the old and new element must undergo a redesign of interfaces and the addition or deletion of existing components and modules, driving up the technical complexity of the project. It has been shown that technical complexity is a major source of both cost and schedule overruns for space systems [Bearden, 2003; Committee on Assessment of Impediments to Interagency Cooperation on Space and Earth Science Missions, 2011], which justifies our use of the present metric as a proxy for architecture switching cost.

Detailed Approach - Problem 1

Here we discuss the specifics of applying the general framework presented in Section 4 to the in-space transportation infrastructure tradespace. The definition of a node for this application along with a further decomposition of the architecture is shown for reference in Figure 2.

In Problem 1 the architecture family is fixed along the development pathway while the technology portfolio changes from step to step. Now that we have an understanding of the definition of distance, it is important to clarify that an architecture family is defined by a fixed functional architecture, while the other properties in Figure 2 are free to vary among family members.

The statement of Problem 1 allows for the selection of the initial architecture e_1 , which in reality is a highly complex problem in itself. We will leave the optimization of this selection to future work, and here will use a simple approach. We plot each architecture in the tradespace according to IMLEO and the TDC of its required technologies, and then select one architecture from the Pareto front of this plot as e_1 .

With the initial architecture specified the next step is to reduce the set of architectures in the tradespace to only those which are in the same family as e_1 by finding all nodes with the same functional architecture. In practice once e_1 has been selected there is some investment made in its set of required technologies, so it is necessary to reflect this fact in the rest of the tradespace. We define the technology portfolio of a node as the set of required technologies for the node's architecture plus any additional technologies required for the initial architecture. The TDC of the node is then calculated according to this (possibly) expanded set. Intuitively this modification moves nodes with surplus technologies farther along the x-axis, making them less desirable.

The connectivity of the tradespace graph and the edge weights can now be defined, with the weight of an edge equal to the difference in TDC between the two nodes. The final node is selected by the user from the set of Pareto optimal nodes along the metrics of IMLEO and portfolio TDC, at which point the optimal architecture evolution and technology progression are calculated.

Detailed Approach - Problem 2

In Problem 2 the technology portfolio is fixed along the development pathway while the architecture is allowed to vary. In this problem we seek to model the exogenous addition or deletion of a technology to or from the portfolio and then find the possible ways to change the architecture to account for this technology disruption. The process of selecting e_1 here is significantly more complex than in Problem 1 since the technology addition or deletion happens before the generation of the graph. We first assume that some pre-disruption active architecture and technology portfolio exist and are known.

The user then specifies the technologies to add or delete from the portfolio, which establishes T^k for all k . The initial architecture, e_1 , is determined by finding the set of all architectures with the same functional architecture as the pre-disruption architecture that are also feasible given the disrupted technology portfolio. If this set contains multiple architectures, the minimum IMLEO one is e_1 , and if it contains no feasible architectures we specify that e_1 is infeasible.

The tradespace graph is generated using only those architectures which are feasible given the post-disruption portfolio. Edge weights are determined as described above, with $\delta = 1$ so that only architectures with distance one from each other are connected. The set of possible final architectures is defined in a manner similar to that used in Problem 1, except instead of using the IMLEO and TDC Pareto front we find the Pareto front for IMLEO and distance from e_1 .

6 Results

In this section we present the results of the application of our tradespace exploration framework to the in-space transportation infrastructure for human exploration. We will analyze the tradespace of a 500-day exploration mission to the surface of Mars for Problems 1 and 2 identified in Section 3. This tradespace is well suited to the application of our framework since architectures in it require substantial technology investment, architecture and technology decisions are tightly coupled, and the tradespace spans a wide variety of concepts.

Problem 1 - Fixed Architecture Family

The starting point for this problem is a representation of the entire tradespace, shown in Figure 4. Here the grey markers represent all 75,351 feasible architectures for the 500-day Mars surface mission, plotted along the metrics of IMLEO (in metric tons) and Technology Development Cost. The Pareto optimal architectures of this tradespace are plotted as circles. For Problem 1 the minimum TDC architecture from among this set has been specified as e_1 .

This architecture, which is similar to many architectures found in the low-TDC region of the tradespace distributes both habitation and propulsion functions among many separate elements. The technologies required for this architecture, which define the initial technology portfolio, $T^{(1)}$ are: in-space LOX-hydrogen, Mars descent hypergol, Mars ascent hypergol, solar electric propulsion for cargo predeployment, and aerocapture.

The black markers in Figure 4 represent the 288 architectures which have the same functional architecture as e_1 . This family of architectures spans the range of TDC values, but moves farther from the Pareto front at higher values, indicating that this functional architecture is not well suited to advanced technology portfolios.

This same family of architectures is shown in Figure 5, with the necessary adjustment made to TDC as described in Section 5, which stretches the tradespace along the x-axis and alters the Pareto front. Let us assume for this example that a decision maker wishes to find the optimal prioritization of technology investments to minimize IMLEO for the selected architecture. In this case the the minimum IMLEO node in the e_1 family becomes the final node.

The optimal pathway from initial node to final node is shown in Figure 5 as the series of circle markers connected by a solid line. The technologies added at each step on the pathway are shown in Table 2. To find this path we use Yen's algorithm to find all of the paths from initial to final node that minimize the total technology development cost, and from this set select the one which minimizes the sum of IMLEO of the intermediate architectures. The fact that the pathway remains along the Pareto front of the tradespace is an ideal result since it means that all intermediate architectures maximize returns on technology investment.

However the optimal path will not coincide with the Pareto frontier in all cases, rather it will be the minimum cost path nearest to the Pareto frontier.

We can contrast this optimal path with a different path shown as the dashed line in Figure 5, which has the same total TDC as the optimal path with higher IMLEO intermediate architectures. This second path enforces the constraint that Nuclear Thermal Propulsion is the final technology added to the portfolio. This restriction moves the path far away from the Pareto front even though the starting and ending nodes are the same. This comparison suggests that the value of this approach as applied to Problem 1 is to inform the decision maker as to the best way to prioritize investments in technology in order to ensure that even if low priority technologies cannot be developed in time, the resulting architecture-technology combination will retain the largest possible performance improvement.

Regarding the optimal path, it is clear from both Figure 5 and Table 2 that the majority of the improvements in IMLEO occur with the introduction of both boil-off control and nuclear thermal propulsion during the first two steps of the pathway, while the latter three technologies provide much smaller improvements.

Among these three technologies, the one with the largest impact on IMLEO across the tradespace of all architectures is ISRU, however the functional architecture fixed by e_1 is not able to realize ISRU's full potential. Recognizing this negative coupling between architecture and technology is another valuable insight that this analysis provides.

Problem 2 - Fixed Technology Portfolio

As an example of Problem 2, we will use the framework presented in the previous sections to analyze the case in which the failure to develop a planned technology disrupts the tradespace. We use the same starting tradespace as that studied in Problem 1, only in this case we start with an architecture on the minimum IMLEO end of the Pareto front as shown in Figure 4.

The functional architecture of this selection from a habitation perspective is 'monolithic', that is to say all functions except Earth re-entry are assigned to a single habitat. From a propulsion perspective, Mars descent and ascent burns share a single stage, with the same true for Mars arrival and departure maneuvers. This architecture requires the full set of non-propellant technologies, with the descent and ascent propellant technologies both methane-based.

We consider the case that at some time after initial architecture and technology selection, some exogenous factor causes the technology portfolio to change, with methane being replaced by hydrogen for the descent and hypergol for the ascent. This post-disruption set becomes the fixed technology portfolio.

This disruption to the technology portfolio will force the architecture to change, as the required technologies for the active architecture are no longer available. Ideally the functional architecture would remain constant. In fact technology and functional architecture are so tightly coupled in this region of the tradespace that the shift in technology makes the current functional architecture infeasible.

Figure 6 shows this technology disruption process as well as the pathway that is found to change the functional architecture in order to reduce IMLEO. The tradespace shown is plotted with respect to IMLEO and distance from the functional architecture of e_1 . The grey markers represent all architectures in the tradespace, while the black markers are the ones that are feasible with the post-disruption technologies, which comprise the 255 nodes in the tradespace graph. The final node is selected to be the minimum IMLEO architecture within this new feasible set, and is labeled e_4 in Figure 6. We constrain each step along the development pathway to occur over only a single unit of architecture distance ($\delta = 1$).

The changes to the functional architecture made at each step in the evolution are shown in Table 3. We see that at each step functions are broken off from monolithic elements and assigned to specialized elements. This disaggregation of the architecture occurs because the coupling between propellant and ISRU is no longer strong enough to efficiently support monolithic elements after the technology disruption.

This type of information delivers value to a decision maker who is analyzing the robustness of his or her architecture to disruptions during the system development stage. The analysis above indicates that in this case technology and architecture are very tightly coupled and the system is not robust to technology disruption. However, should a disruption occur, the indicated path tells the decision maker what his or her contingency options could be and what the relationship is between spending additional resources to change the architecture and improving performance.

7 Conclusions and Future Work

This paper presents a graph-based framework for determining developmental pathways through large tradespaces defined by both system architecture and technology portfolio decisions. The basis of this framework is a division of the tradespace exploration problem into three subproblems based on the different situations in which a decision maker might wish to explore the relationships between architecture, technology, performance, and cost. By transforming the tradespace into a graph and restating the tradespace exploration problem as a shortest path problem, we make explicit the assumptions and decision criteria that determines the architecture and technology decisions to be made in order to travel from one region of the tradespace to another.

The application of this framework to a tradespace of tens of thousands of architectures to generate graphs of a few hundred nodes for the in-space transportation infrastructure example demonstrates the utility of applying this computational tool to a complex and disorganized tradespace. For the case in which the architecture family is fixed, we see that the proposed framework provides a means for decision makers to understand how the sequencing of technologies affects intermediate performance, while the fixed technology case allows the decision maker to determine how robust an architecture is to negative exogenous impacts. In both cases the determination of the shortest cost incremental path through the tradespace captures the coupling between technology and architecture that is not apparent from looking at the initial and final architectures individually.

A number of directions for future research to extend and improve this framework have been identified. The first of these is incorporating Problem 3 (varying architecture and technology portfolio) into the analysis of the in-space transportation infrastructure. Developing a more sophisticated method to select the initial architecture, e_1 , is another major goal for future work. Making this initial selection based on the quality or variety of available development paths has the potential to greatly improve the utility of this method. Finally, modifying the framework to allow for functional requirements to change along the evolution (e.g. changing destinations and missions for the in-space transportation example) would allow this method to be applied to a much wider and richer selection of tradespace exploration problems.

Acknowledgments

This research has been supported by the Skolkovo Institute of Science and Technology.

References

- D. Arney. *Rule-Based Graph Theory to Enable Exploration of the Space System Architecture Design Space*. PhD thesis, Georgia Institute of Technology, 2012.
- A. Avadikyan and P. Llerena. A Real Options Reasoning Approach to Hybrid Vehicle Investments. *Technological Forecasting and Social Change*, 77(4):649–661, 2010.
- J. A. Battat, B. Cameron, A. Rudat, and E. F. Crawley. Technology Decisions Under Architectural Uncertainty. *Journal of Spacecraft and Rockets*, 5(5), 2013.
- D. A. Bearden. A Complexity-Based Risk Assessment of Low-Cost Planetary Missions: When is a mission too fast and too cheap? *Acta Astronautica*, 52(2):371–379, 2003.
- M. Benaroch and R. J. Kauffman. A Case for Using Real Options Pricing Analysis to Evaluate Information Technology Project Investments. *Information Systems Research*, 10(1):70–86, 1999.
- G. Boothroyd and P. Dewhurst. *Product Design for Assembly*. Boothroyd Dewhurst Incorporated, 1987.
- A. Brown and M. Thomas. Reengineering the Naval Ship Concept Design Process. In *Ship Systems Engineering Symposium, ASNE*, 1998.
- T. R. Browning. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. *IEEE Transactions on Engineering Management*, 48(3), 2001.
- B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest Path Algorithms: Theory and Experimental Evaluation. *Mathematical Programming*, 73(2):129–174, 1996.
- Committee on Assessment of Impediments to Interagency Cooperation on Space and Earth Science Missions. Assessment of Impediments to Interagency Collaboration on Space and Earth Science Missions. Technical report, National Research Council, Washington D.C., 2011.
- Z. Covaliu and R. M. Oliver. Representation and Solution of Decision Problems Using Sequential Decision Diagrams. *Management Science*, 41(12):1860–1881, 1995.
- E. F. Crawley, O. L. de Weck, S. Eppinger, C. Magee, J. Moses, W. Seering, J. Schindall, D. Wallace, and D. Whitney. Engineering Systems Monograph: The Influence of Architecture in Engineering Systems. In *Engineering Systems Symposium*, Cambridge, MA, 2004.
- W. H. Day. The Complexity of Computing Metric Distances Between Partitions. *Mathematical Social Sciences*, 1(3):269–287, 1981.
- O. de Weck, R. de Neufville, and M. Chaize. Staged Deployment of Communications Satellite Constellations in Low Earth Orbit. *Journal of Aerospace Computing, Information and Communication*, 1(3):119–136, 2004.
- E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- D. Dori. *Object Process Methodology: A Holistic Systems Paradigm*. Springer, 2002.
- C. Eckert, P. J. Clarkson, and W. Zanker. Change and Customisation in Complex Engineering Domains. *Research in Engineering Design*, 15(1):1–21, 2004.
- A. Eder and M. Linde. Efficient and Dynamic - The BMW Group Roadmap for the Applications of Thermoelectric Generators, 2011.
- K. Haris and C. H. Dagli. Adaptive Reconfiguration of Complex System Architecture. *Procedia Computer Science*, 6:147–152, 2011.

- W. M. Hawes and M. R. Duffey. Formulation of Financial Valuation Methodologies for NASA's Human Spaceflight Projects. *Project Management Journal*, 39(1):85–94, 2008.
- K. Heidenberger and C. Stummer. Research and Development Project Selection and Resource Allocation: A Review of Quantitative Modelling Approaches. *International Journal of Management Reviews*, 1(2):197–224, 1999.
- R. M. Henderson and K. B. Clark. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, 35(1):9–30, 1990.
- B. H. Y. Koo, W. L. Simmons, and E. F. Crawley. Algebra of Systems: A Metalanguage for Model Synthesis and Evaluation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 39(3):501–513, May 2009. ISSN 1083-4427.
- R. Kumar, H. Ajjan, and Y. Niu. Information Technology Portfolio Management: Literature Review, Framework, and Research Issues. *Information Resources Management Journal*, 21(3):64–87, 2008.
- M. J. Liberatore and A. C. Stylianou. The Development Manager's Advisory System: A KnowledgeBased DSS Tool for Project Assessment. *Decision Sciences*, 24(5):953–976, 1993.
- M. L. Maher and J. Poon. Modeling Design Exploration as Co-Evolution. *Computer-Aided Civil and Infrastructure Engineering*, 11(3):195–209, 1996.
- J. R. R. A. Martins and A. B. Lambe. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal (Submitted for Publication)*, 2012.
- H. L. McManus and J. M. Warmkessel. Creating Advanced Architectures for Space Systems: Emergent Lessons from New Processes. *Journal of Spacecraft and Rockets*, 41(1):69–74, 2004.
- A. F. Mehr and I. Y. Tumer. Risk-Based Decision-Making for Managing Resources During the Design of Complex Aerospace Systems. *Journal of Mechanical Design*, 128(4):1014–1022, 2006.
- A. Mehrez. Selecting R&D Projects: A Case Study of the Expected Utility Approach. *Technovation*, 8(4):299–311, 1988.
- T. Nakamura and V. R. Basili. Metrics of Software Architecture Changes Based on Structural Distance. In *Proc. of the 11th IEEE International Software Metrics Symposium*, page 8, 2005.
- NASA. Human Exploration of Mars Design Reference Architecture 5.0. Technical report, NASA, 2009.
- NASA. Preliminary Report Regarding NASAs Space Launch System and Multi-Purpose Crew Vehicle. Technical report, 2011.
- National Research Council. The Global Positioning System: A Shared National Asset. Technical report, The National Academies Press, Washington D.C., 1995.
- Office of the USAF Chief Scientist. United States Air Force Technology Horizons 2010-2030. Technical report, 2010.
- A. M. Ross and D. E. Hastings. The Tradespace Exploration Paradigm. In *INCOSE International Symposium 2005*, pages 13–25, 2005.
- A. Rudat. *HEXANE: Architecting Manned Space Exploration Missions Beyond Low Earth Orbit*. Master's thesis, Massachusetts Institute of Technology, 2013.
- A. Rudat, J. A. Battat, B. Cameron, M. Dwyer, and E. F. Crawley. Tradespace Exploration Approach For Architectural Definition of In-Space Transportation Infrastructure Systems For Future Human Space Exploration. In *International Astronautical Conference*, 2012.
- J. H. Saleh, E. Lamassoure, and D. E. Hastings. Space Systems Flexibility Provided by On-Orbit Servicing: Part 1. *Journal of Spacecraft and Rockets*, 39(4):551–560, 2002.

- R. Shishko, D. H. Ebbeler, and G. Fox. NASA Technology Assessment Using Real Options Valuation. *Systems Engineering*, 7(1):1–13, 2004.
- M. R. Silver and O. L. de Weck. Time-Expanded Decision Networks: A Framework for Designing Evolvable Complex Systems. *Systems Engineering*, 10(2):167–186, 2007.
- W. L. Simmons. *A Framework for Decision Support in Systems Architecting*. PhD thesis, Massachusetts Institute of Technology, 2008.
- D. A. Simovici and S. Jaroszewicz. An Axiomatization of Partition Entropy. *Theory, IEEE Transactions on Information*, 48(7):2138–2142, 2002.
- A. Singh and C. H. Dagli. Multi-objective Stochastic Heuristic Methodology for Tradespace Exploration of a Network Centric System of Systems. In *IEEE Systems Conference*, pages 218–223, 2009.
- K. Sinha and O. de Weck. Structural Complexity Quantification for Engineered Complex Systems. In *Proceedings of the International Design Engineering Conference*, Portland, OR, 2013.
- R. Smaling and O. de Weck. Assessing Risks and Opportunities of Technology Infusion in System Design. *Systems Engineering*, 10(1):1–25, 2007.
- Steering Committee for NASA Technology Roadmaps and National Research Council. NASA Space Technology Roadmaps and Priorities: Restoring NASA’s Technological Edge and Paving the Way for a New Era in Space. Technical report, 2012.
- Z. Szajnfarber, M. G. Richards, and A. Weigel. Challenges to Innovation in the Government Space Sector. *Defense Acquisition Review Journal*, 59:257–276, 2011.
- U.S. Senate. National Aeronautics and Space Administration Authorization Act of 2010, 2010.
- M. A. Walton and D. E. Hastings. Applications of Uncertainty Analysis to Architecture Selection of Satellite Systems. *Journal of Spacecraft and Rockets*, 41(1):75–84, 2004.
- J. Y. Yen. Finding the k Shortest Loopless Paths in a Network. *Management Science*, 17(11):712–716, 1971.

Table 1: List of technologies and relative development costs developed in Battat et al [2013].

Technology Description	Score
In-Space Nuclear Thermal Rocket (NTR)	1
In-Space LOX-Hydrogen Propulsion	$\frac{1}{6}$
In-Space LOX-Methane Propulsion	$\frac{1}{2}$
Descent LOX-Hydrogen Propulsion	1
Descent LOX-Methane Propulsion	1
Descent Hypergol Propulsion	1
Ascent LOX-Hydrogen Propulsion	1
Ascent LOX-Methane Propulsion	1
Ascent Hypergol Propulsion	1
Solar Electric Propulsion	$\frac{1}{3}$
Propellant Boil-Off Control	$\frac{1}{2}$
Aerocapture	$\frac{1}{3}$
Surface In-Situ Resource Utilization (ISRU)	1

Table 2: Technology progression and IMLEO savings for fixed architecture family problem.

	Technology Added	Δ IMLEO [mt]
$e_1 \rightarrow e_2$	Boil-Off Control	19
$e_2 \rightarrow e_3$	Nuclear Thermal Propulsion	143
$e_3 \rightarrow e_4$	In-Situ Resource Utilization	45
$e_4 \rightarrow e_5$	Methane Ascent Engine	24
$e_5 \rightarrow e_6$	Hydrogen Descent Engine	10

Table 3: Changes to the functional architecture at each step in the evolution.

Step	Change to Functional Architecture
$e_1 \rightarrow e_2$	Divide ascent/descent propulsion into separate stages.
$e_2 \rightarrow e_3$	Divide deep-space return habitation into separate element.
$e_3 \rightarrow e_4$	Divide ascent habitation into separate element.

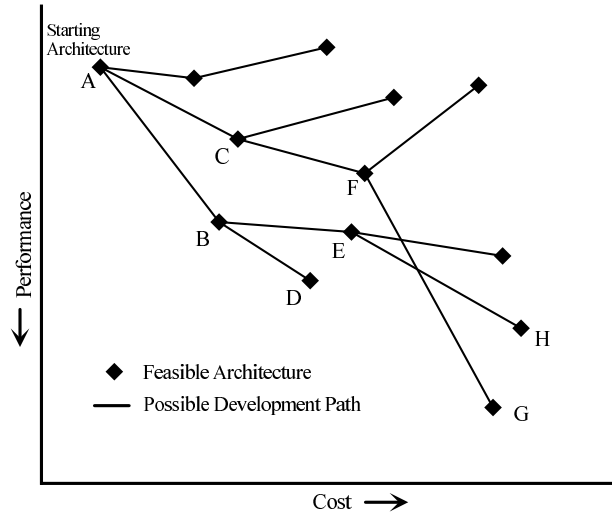


Figure 1: An example of the tradespace exploration problem. Points represent architectures in the tradespace, plotted according to performance (improving performance in negative direction) and cost. Lines represent development paths between architectures.

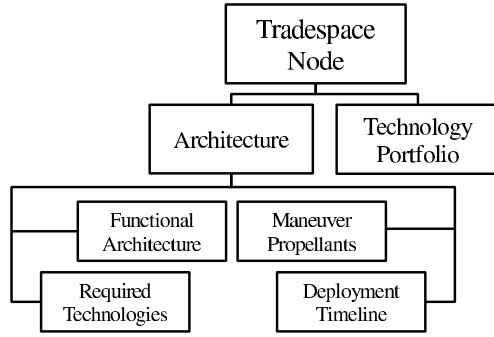


Figure 2: Hierarchy of components that make up a node in the in-space transportation infrastructure application.

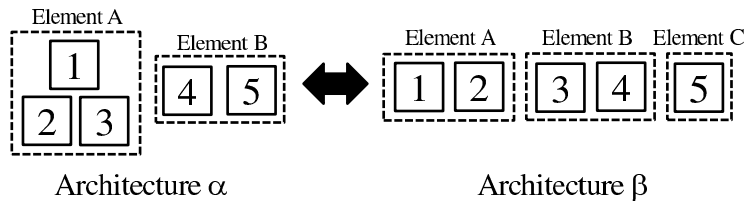


Figure 3: Two possible architectures α and β for five functions. Each solid box labeled with a number represents a function while each dashed box is an element of form, which is assigned one or more functions. To transform α to β Function 5 is split off to create Element C. Function 3 is then transferred to Element B from Element A. Since there are two operations the distance between α and β is two.

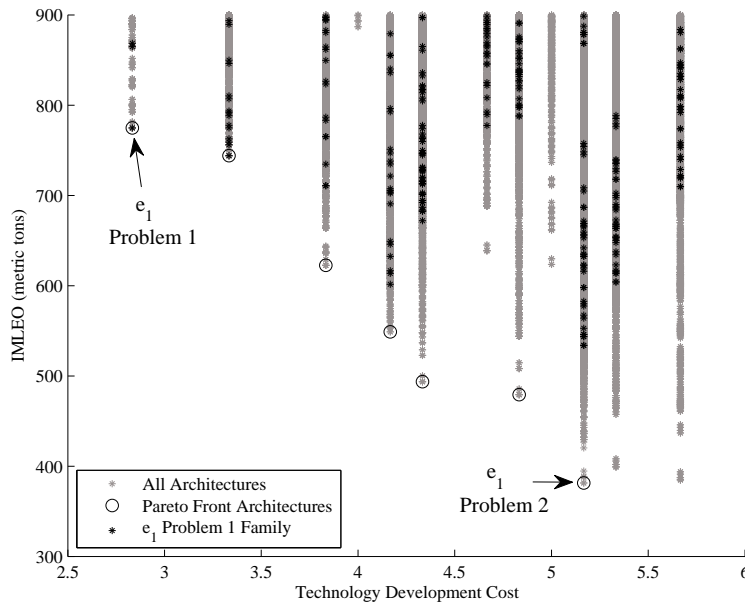


Figure 4: Overview of the enumerated tradespace of 75,351 architectures for a Mars exploration mission plotted according to architecture IMLEO and Technology Development Cost.

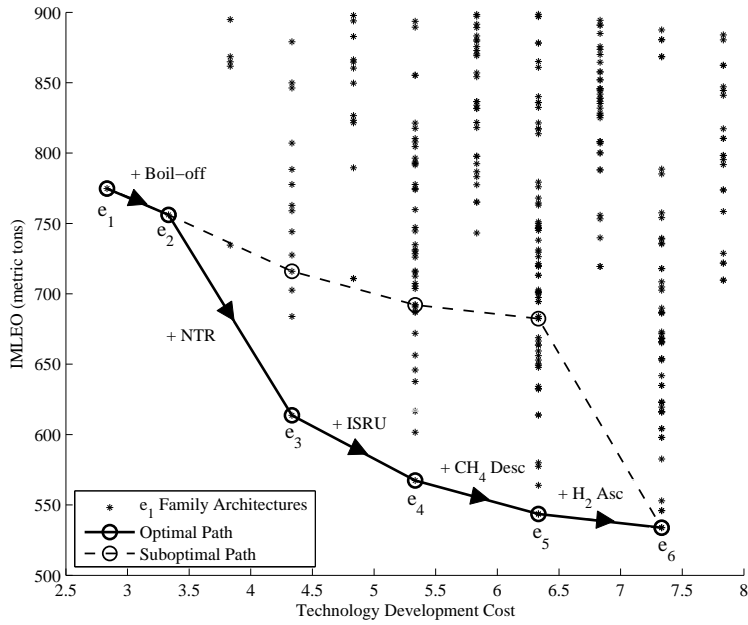


Figure 5: Evolution of architectures from initial to final showing only the 288 architectures with identical functional architecture as e_1 . There are 5,644 edges between these nodes in the tradespace graph. TDC adjusted for initial portfolio shown on the x-axis.

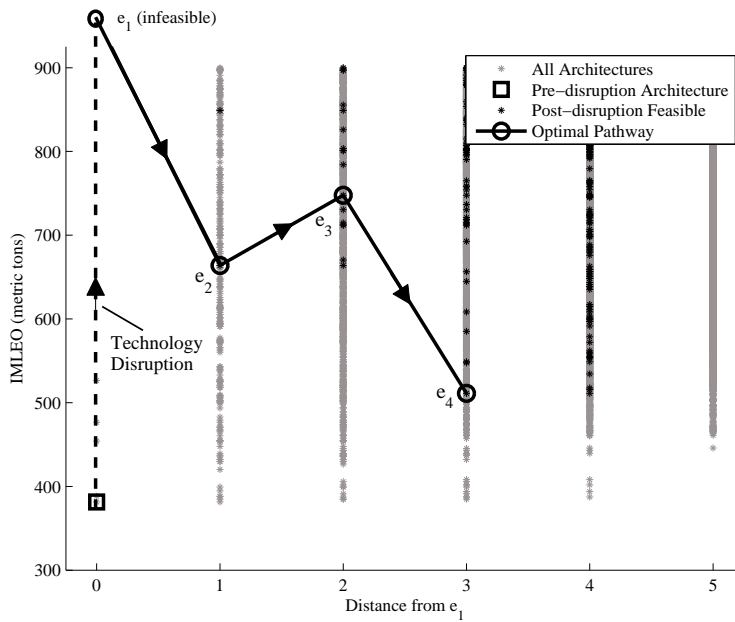


Figure 6: Architecture evolution after a technology disruption resulting in an infeasible initial architecture. There are 255 feasible architectures and 5,137 edges in the graph representation of the tradespace.