Interplanetary Supply Chain Management and Logistics at NASA

Olivier de Weck, Jaemyung Ahn, Erica Gralla, Sarah Shull, David Simchi-Levi, Christine Taylor
Massachusetts Institute of Technology, Cambridge, MA
(deweck,jaemyung,egralla,sshull,dslevi,c_taylor@mit.edu)

Diego Klabjan
Northwestern University, Evanston, IL (d-klabjan@northwestern.edu)

Previous space exploration missions relied on simple logistics paradigms, such as self sufficient missions of Apollo type or continuous resupply as is the case with the International Space Station. Future sustainable Lunar and Mars explorations will use advanced technologies with many intertwined missions. As a result, the underlying logistics network will be complex, requiring intricate planning decisions. In close collaboration with NASA, a decision support system was developed for future supply chain management and logistics planning. The system contains a custom built database, a graphical user interface, an optimization component, and a simulation and visualization tool. This system provides NASA with a strategic planning tool to develop cost effective solutions and enable quick studies of high level decisions such as in space spares management, risk assessments, pre-positioning strategies, and high level architectural design.

Keywords: transportation: network; government: services; computers-computer science: System design-operation

Introduction

In January 2004, President George W. Bush announced a new plan to send humans back to the Moon and possibly to Mars. In response to this directive, NASA has developed a plan to step the next human on the Moon by 2020 and then extend human missions beyond the Moon. Immediately after this announcement, the Exploration Systems Architecture Study (ESAS) was carried out. The study identified key components for future explorations such as launch vehicles and crew exploration vehicles, and the underlying crucial technologies. The evolution outlined by the study recommends that initial sortie missions be followed by several pre-deploy missions and lead to the development of a lunar base with continuous resupply. Based on the best practices from the Apollo and International Space Station (ISS) programs, and part of the NASA's Constellation Program, the design and development of the underlying hardware components has already begun.

Unlike the Apollo missions, which were self-sustainable and independent of each other, future missions will be enabled in part by a coordinated and integrated mission plan which will require missions to be heavily intertwined. The underlying in-space logistics network will necessarily be more complex and require sophisticated tools to analyze it and the underlying core processes. This complexity also leads to several possible network designs, cost effective strategies, and the requirement of a software tool capable of building the underlying logistics network, evaluating trade-offs, risk assessments, and performing trade studies. In close collaboration with

NASA, a decision support system, called SpaceNet, was developed to provide the aforementioned functionalities, facilitating the agency with a high level strategic decision making capability. SpaceNet is a comprehensive system based on a carefully designed database, a sophisticated graphical user interface, employing state-of-the-art optimization algorithms and analysis tools, and including simulation and visualization components.

The basic user-defined input to SpaceNet is a demand scenario, which specifies time windows together with physical locations on planetary surfaces and equipment requirements for explorations. Mission planners, logisticians, and operations planners use the decision support system to directly obtain recommendations on:

- *launch planning*: when and what to launch with which launch vehicles,
- *network design*: what trajectories to use and when, together with the underlying architectural design of the vehicle (basic components called elements, amount of propellant, payload mass and capacity), and
- *manifesting*: what is the flow of all cargo including crew members.

By employing what-if analysis together with the underlying simulation, much deeper decision making is assisted:

- Is it beneficial to pre-position cargo or even to build an in space depot?
- What are the ramifications of equipment failures, scrubbed launches, and uncertain demand?
- What is the sensitivity with respect to the payload mass and volume of elements?

As NASA moves towards enterprise-like objectives and goals, the deployment of tools such as SpaceNet will be of vital importance. Decision making based on sound analytics is not new to the agency, however, investigations into space supply chain and logistics questions have only recently started to be tackled. SpaceNet represents a substantial step forward in this direction.

Interplanetary Supply Chain Management and Logistics

Past human space exploration programs have followed different types of logistics paradigms. Under the Apollo program, six missions to the lunar surface were conducted between 1969 and 1972. Each mission was self-contained and therefore did not require the existence of a space logistics network. On the other hand, the logistics strategy for the ISS program is based on regular resupply flights by various vehicles, including the American Space Shuttle and the Russian Progress. Figure 1 depicts the basic networks behind each of these logistics paradigms, as well as the growing network that may be needed to support the next-generation space exploration programs now in the planning stages. Relatively simple logistics strategies functioned well for the two major U.S. spaceflight programs that have been operated to date, but the next-generation network appears much more complex. This leads to the question: What is the best logistics paradigm for next-generation sustainable space missions? It is unclear exactly what combination of these strategies will provide the most affordable, sustainable, and robust supply chain for next-generation programs. Due to increasing complexity, decision making related to such questions within NASA should be computer based and it should rely on analytics and simulation. In close collaboration with NASA we have developed and delivered SpaceNet, a tool for interplanetary supply chain management and logistics based on analytics and simulation.

Due to the increasing complexity of future space operations, a holistic view at the systems level is required. Missions and campaigns are driven by exploration demands and therefore the entire process is demand driven. A demand scenario can, for example, specify that several crew members need to be on the Lunar surface during a given time period, which also specifies the required food and water supplies that must accompany them. In order to perform an exploration mission, some equipment and a crew habitat might be required during the same time period and location. In addition, the crew members must return back to Earth during a specified time period. For an extended campaign, many such demand scenarios are specified.

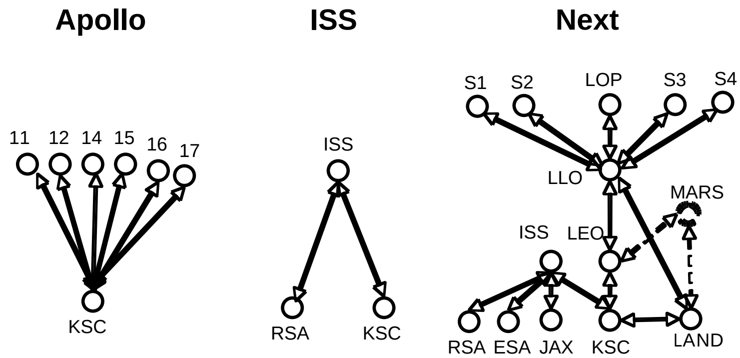

Figure 1: Space Logistics Supply Chains

The main objective of SpaceNet is to design the underlying logistics network to support the specified demand scenarios. Basic components, named elements, of space vehicles (called stacks) are specified. Clearly at specific in-space nodes cargo can be stored and reused at later points in time. As a result, a single stack can serve several demand scenarios. It is interesting to note that in a typical scenario the total cargo mass, including crew members, represents only a small percentage of the total launched mass. Element designs can (and should) be subjected to trade studies, evaluating their relative suitability for logistics.

One of the most important features of SpaceNet is its ability to have a holistic, system wide view of missions and campaigns. Logistics capabilities required for one demand scenario might be reused in a different demand scenario, an option that is difficult to capture in existing processes used by NASA. Such a global view also enables decision makers to easily assess via trade studies, the benefits and costs of pre-positioning, spare parts policies, uncrewed options, and risk pooling. Without a computer based system, it is difficult if not impossible to perform such an analysis.

Modeling interplanetary operations and dynamics poses unique challenges, which are outlined next.

Modeling Framework

One of the major challenges in the development of a space logistics model is defining the model components. Interplanetary logistics has not been previously modeled, so the scope of such a model must be defined. The basic elements of the model are: Movement (shipment of people, cargo, and stacks), Demand (by supply class), Information Architecture, Simulation and Optimization. Exploration architectures are modeled as a set of nodes (locations on a surface, in an orbit, or at a Lagrange point) and arcs (trajectories between these locations). Demand is generated at nodes; e.g., a mission at a lunar surface node would generate demand for crew provisions (food, water), science equipment, etc. Stacks traverse arcs carrying supplies to satisfy the demand. Users can either manually define the shipment paths through the network, or use the optimization tool to find the best solution given a particular demand scenario.

The challenge of integrating these components into a cohesive end-to-end logistics and operations model is discussed in the next subsections. First, we describe the basic building blocks of our modeling framework (nodes, elements, and supplies), along with two concepts which enable us to tie these together: the time-expanded network and processes for movement through the network. Collectively, this framework allows us to describe and model both the demand and the movement of items in the logistics scenario. Finally, we describe the remaining layers which enable the effective utilization of this modeling framework: the ability to simulate and evaluate various architectures, and, more importantly, to apply optimization techniques.

Nodes

Nodes are spatial locations in the solar system. The existence of a node does not necessarily indicate that a facility exists at that location or that a node is ever used or visited. A node is simply a way to refer to locations in space. Nodes can be of three basic types: Surface nodes, Orbital nodes, and Lagrange nodes. The nomenclature developed around nodes allows us to build up a potential transportation network and thus to formalize the description of logistics architectures.

Supplies

Supplies are the items that move through the network, from node to node. Generally, supplies should include all the items needed at the planetary base, or during the journeys to and from the base. Examples include consumables, science equipment, surface vehicles, and spares. In order to track and to model the extraordinary variety of supplies that could be required, they must be classified into larger categories. We developed a set of 10 *functional* classes of supply, organized regardless of material or owner. The classes are therefore based on the essential functions of a planetary base, or the tasks that need to be accomplished, such as research, habitation, transportation, etc. These classes of supply form the basis for the modeling of supply items. Recall that the impetus behind the development of these supply classes was the need for a manageable modeling framework for supplies moving through a transportation network. With these ten supply classes we can model demand for various types of items at the supply class level.

Elements

Elements are defined as the indivisible physical objects that travel through the network and (in general) can hold or transport supplies. Several elements form a vehicle or stack, or what we usually think of as “rockets”. Typical elements are the crew exploration vehicle (CEV) Orion, propulsion stages, etc. At a given node, two inbound stacks can be completely disassembled or undocked, and reconfigured or docked into different stacks. Elements are characterized by a wide set of characteristics and abilities:

- hold supply items,
- be propulsive or non-propulsive,
- carry crew or not carry crew,
- be reused, refueled, disposed of (staged), pre-deployed, and
- be “docked” with other elements to form a (temporary) stack.

In general, an element has defined capacities for three types of items: crew, cargo, and propellant. These capacities determine what types of supplies can be assigned to that element for transport, and whether the element is propulsive. Thus, elements can transport supplies and crew between the various nodes of the transportation network.

With the preceding definitions of nodes, supplies, and elements, we have defined the basic building blocks of the modeling framework for space logistics.

The Role of NASA

As the primary customer of the modeling framework described above, several divisions within NASA have played a critical role in the guidance of the development of SpaceNet. While it is envisioned that this framework would be utilized by many organizations at NASA over the coming decades, the near-term usage is predicted to be centered within NASA's Constellation Program. Within this program, customer feedback about SpaceNet has primarily been received from their Modeling and Simulation (M&S) Group at Marshall Spaceflight Center and the Architecture, Trades and Analysis (ATA) Group at Johnson Space Center. These two divisions of NASA are seen as the primary near-term users of SpaceNet and as such their input has been very valuable.

The M&S Group is responsible for oversight and coordination of the majority of NASA's tools used for modeling and simulation. This daunting task includes attempting to ensure that new tools that are being developed are compliant with the appropriate NASA databases and existing tools. The M&S group will continue to be heavily involved in its development.

The ATA group spends much of their time performing trade studies for NASA's Constellation Program. As such this group uses a wide-range of modeling and simulation tools. During the summer of 2006, SpaceNet was utilized to support several ATA trade studies investigating the logistical impacts of various design decisions under consideration by NASA. As was mentioned in the introduction, the inspiration for this modeling framework was to assist NASA in applying terrestrial logistics methodologies to the interplanetary supply chain. Since this goal aligns well with the work performed in the ATA group, ATA will continue to be a key user of SpaceNet for many years.

The Decision Support System: SpaceNet

SpaceNet is a software tool for modeling the logistics of interplanetary space exploration. It includes a discrete event simulation at the individual mission level (e.g., sortie, pre-deploy, or resupply) or at the campaign level (i.e., set of missions). It also allows for the evaluation of manually generated exploration scenarios with respect to measures of effectiveness and feasibility, as well as the visualization of the flow of elements and supply items through the interplanetary supply chain. Finally, it includes an optimization capability and acts as a software tool to support trade studies and architecture analyses.

SpaceNet is built around the modeling framework described in Section "Modeling Framework". Missions are modeled on a network of nodes and arcs, with elements carrying supplies through the network. A mission is specified and demand is generated based on that mission; a simulation ensures that demands are met for a given scenario (undersupply situations are explicitly flagged as error conditions). SpaceNet thus allows for modeling demand-driven space logistics scenarios. Figure 2 provides an overview of the SpaceNet architecture and usage concept.

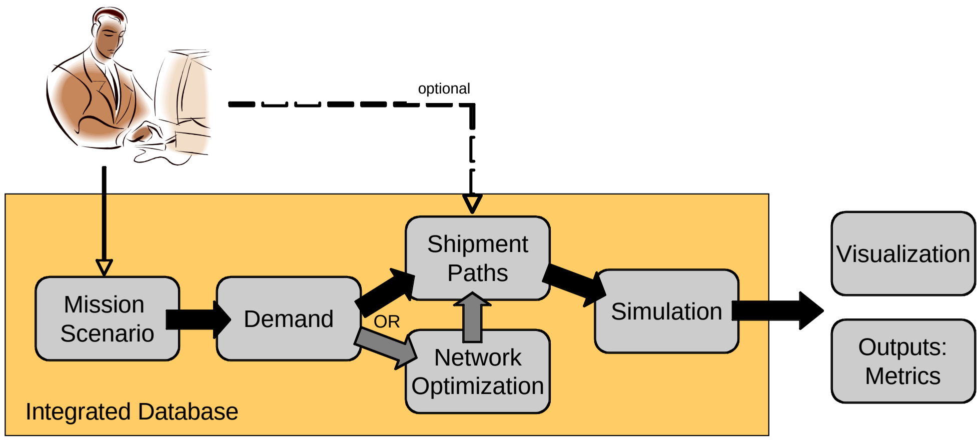

Figure 2: SpaceNet Architecture and Usage Concept

The user begins by defining a specific exploration mission, including the location (e.g., lunar surface), the number of crew, the length of stay, and required equipment and infrastructure. This mission scenario can include one sortie-type mission (to the lunar surface and back) or an entire campaign (made up of multiple sorties and resupply flights).

Demand is generated automatically based on the mission scenario. The required types and amounts of each of the relevant supply classes are estimated based on the surface mission and in-space transportation requirements.

The next step is to satisfy the demand generated in the previous step by defining how crew and supplies will move through the network to supply the mission. SpaceNet provides two options for carrying out this step. Optimization can be utilized so that the software suggests a set of transportation paths, or the user can enter the transportation paths manually (e.g., to evaluate previous architecture designs).

A discrete event simulation then runs through the logistics architecture and checks for feasibility, ensuring that all demands are indeed met by the transportation paths defined. It also generates a set of measures of effectiveness and produces time-series data tracking of all movements through the network, enabling two types of animated visualizations (discussed in Section "Visualization").

A graphical user interface is also provided, which allows describing complex space logistics architectures using the basic concepts of nodes, elements, and supplies. Built-in demand models, a unified database of nodes, vehicles, supplies, and orbital constraints, and an optimization capability assist the user in describing various types of supply chains. Such logistics scenarios can then be simulated and analyzed for feasibility and performance.

In SpaceNet 1.3, the majority of the modules are implemented in Matlab, while the optimization component utilizes C++. Data and visualizations can also be imported and exported in Excel. Next we provide more details on key components of SpaceNet. The block diagram is depicted in Figure 3. We start with the integrated database.

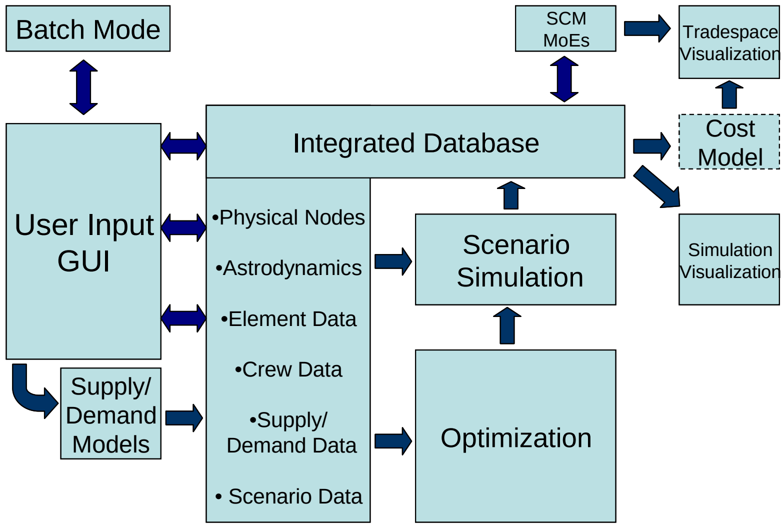

Figure 3: The Block Diagram of the Main Components of SpaceNet

Database Management

The backbone of SpaceNet is the integrated database. This relational database incorporates inventory management features as well as capabilities for manifesting, spares requirements planning, and mission planning. Information maintained in the relational database includes detailed node data, astrodynamics data, element data, supply data, and spares data. Table 1 lists some of the most important tables in the integrated database, along with a brief description of each and the intended usage of this information. It is important to note that much of the value added through the use of a relational database is in the fact that information need only be entered into the database once, thereby eliminating the need for duplication of effort and reducing the chance that an error is made.

Table 1: Integrated Database Components

Table Name	Description	Usage
Supply Class	Attributes common to entire supply classes (i.e. Crew Provisions)	Mission Planning, Manifesting, Real-Time Tracking
Supply Item Type	Attributes common to a supply item type (i.e. Printer Paper)	
Supply Item	Attributes specific to one instance of an item (i.e. Printer Paper Ream X)	
Element Type	Attributes common to an element type (i.e. Space Shuttle)	
Element	Attributes specific to one instance of an element	
Crew	Attributes specific to one crew member	
Astro	Astrodynamics data	Mission Planning
Physical Nodes	List of nodes	
Arcs	List of allowable arcs between nodes	
Crew Provisions	Usage rate data for each supply item defined as crew provisions	Manifesting Real-time Tracking
Crew Operations	Usage rate data for each supply item defined as crew operations	
Parts Common	Common (i.e., regardless of application) data (e.g., function) for spare parts	Spares Requirements Planning, Manifesting
Parts Application	Application-specific data (e.g., mean time be-	

Table Name	Description	Usage
Specific	tween failures) for spare parts	
Maintenance Tasks	Specific maintenance task for each parts application	
Task Resource & Time	Resources required (crew, robotics, etc.) to perform a specific maintenance task	

Contained within the integrated database is detailed information regarding the supplies needed for human space exploration. Generally, the term “supplies” should include all the items needed at the planetary base and during the journeys to and from the base. Examples include consumables, science equipment, surface vehicles, and spares. In order to track and model the extraordinary variety of supplies that could be required, they must be classified into larger categories. A great deal of effort was spent on analyzing various ways to classify supplies. It was determined that the best method was to develop a set of functional classes of supply, organized regardless of material or owner. The classes are therefore based on the essential functions at a planetary base, or the tasks that need to be accomplished, such as research, habitation, transportation, etc. The final set of ten classes of supply is shown in Figure 4.

These classes of supply form the basis for the modeling of supply items. With these ten supply classes we can model demand for various types of items at the supply class level. In addition, we can more easily simulate and track the movement of these aggregate supply items through the transportation network, using the unified relational database described above.

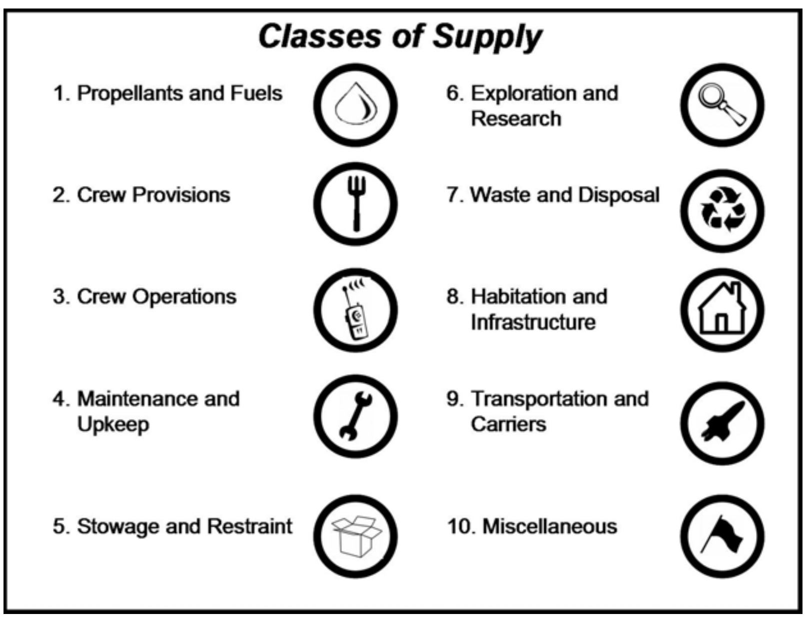

Figure 4: Functional Classes of Supply for Human Space Exploration

The Graphical User Interface

SpaceNet provides a Graphical User Interface (GUI) which allows the user to interact in a convenient fashion with the complex requirements for SpaceNet. By providing a set of pull-down menus that, in the required order, allow the user to specify the required information for SpaceNet to perform the outlined computations, an easy to implement analysis can be configured. First, the mission outline is defined, which includes the type of a mission to be performed, the location of the mission and other nodes to be considered in the network definition, and the primary features of this mission, such as the number of crew and duration of stay at particular locations. For this, and every other option outlined below, the user has the ability to select from a set of pre-specified options, or define their own. Given a specified mission, the nominal demand is outlined. As stated, the nominal demand represents the supplies generally required for a mission of this type, which can be augmented or reconfigured by the user. Finally, a list of available elements to transport the supplies are selected.

At this point, the GUI offers the option to further configure the scenario by defining the manual allocation of supplies and elements to routes in the network, subject to astrodynamic constraints and operational acceptability, or to allow the optimization to determine these options for the user. If the user chooses to manually input these details, another set of screens is produced to allow for an easy manipulation of these details. Alternatively, if the user elects for the optimization to produce these results, as described in the next section, the definition of the routes, the elements selected, and the timing is determined by the optimization.

Optimization

After specifying a scenario in the GUI, a user can either manually enter the underlying logistics decisions such as the usage of elements and supplies, or it can invoke the main optimization routine. Details about optimization are provided in Section "Modeling and Optimization". The distinguishing and unique feature of SpaceNet is its ability to recommend a logistics architecture through optimization.

The underlying decisions consist of first specifying launch times, launch stacks and their contents, the in-space routes of all elements, and the content of each element. The optimization module is written in C++ and it is interfaced with SpaceNet through an interface as shown in Figure 5. The optimization routes are written as a web service and thus the entire interfacing is performed via 'xml' like syntax.

The user can specify several options, some of them are mandatory. Among the required input parameters, the user must select a set of launch vehicles to be considered. Note that all other equipment data such as elements, supply, and demand points reside in the database and are part of the scenario. As a result, the user does not specify them in the optimization interface. The user can adjust optimization parameters such as the maximum number of optimization iterations and the maximum stay of crew.

Demand in the database, which is part of the scenario, is usually input at a more granular level than requested by optimization routines. For example, often it is specified on a per supply unit, however optimization deals with flows of supply and thus it should be aggregated at the supply class level. Therefore before beginning the optimization, the aggregated demand is computed.

The solution is read back by SpaceNet, analyzed and displayed, as is described in the next section.

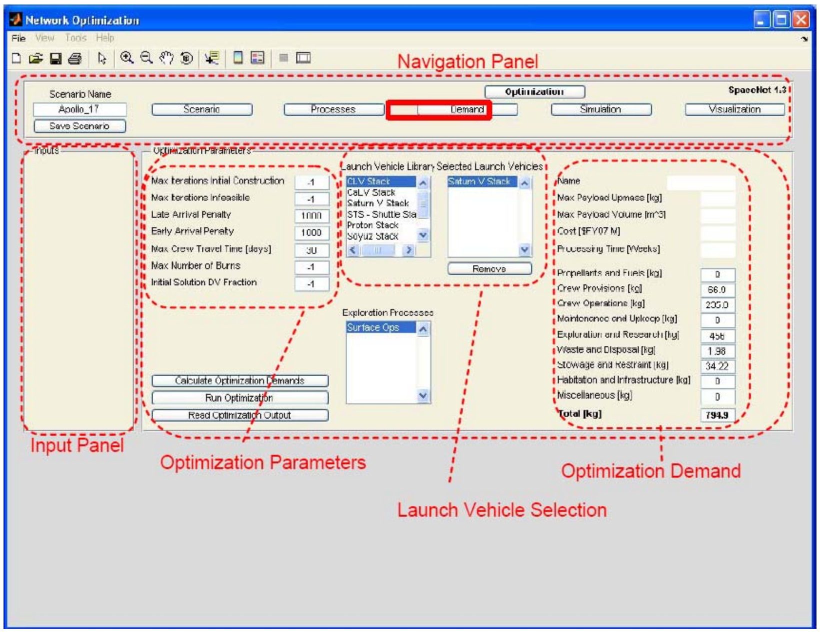

Figure 5: Interface to Optimization Routines

Visualization

After a logistics plan is obtained either through optimization or input manually, it is displayed in several ways. The basic display is a timeline-physical body two dimensional chart shown in the left display in Figure 6. The network arcs used within the selected network are displayed. Different colors represent various processes such as transport, transfer, and exploration.

The solution feasibility can be checked through the simulation tool, which is shown in the right figure in Figure 6. The simulation has several roles. First, it animates the entire mission or campaign by displaying stacks on each arc, their disposal, docking, etc. Second, during the actual animation, it checks for all feasibility requirements, e.g., enough fuel to perform the burns, compatibility of elements and supply classes, satisfying the exploration requirements, and many others. If a requirement is violated, based on its severity, the animation is either aborted or it proceeds. In either case the error is logged. Third, the animation also computes a comprehensive list of measures of effectiveness. Optimization currently captures only the direct cost, which includes the cost of using elements and fuel. Several other robustness and reliability measures were developed: the total launched mass, the number of crew surface days, the number of supply and crew transfers among elements, and several others, see, e.g., SpaceNet 1.3 User's Guide available at spacelogistics.mit.edu/publications.htm, for a complete list.

The solution can also be displayed in various formats within a spreadsheet. The user can export the spreadsheets for a later usage. Among others, the fuel consumption can be depicted and the supply consumption displayed (food, water are consumed while waste is accumulated). A graphical bat chart allows for a fairly intuitive visual representation of the flow of supplies and elements through the network, which can be easily inspected to determine if this architecture meets the requirements of the mission.

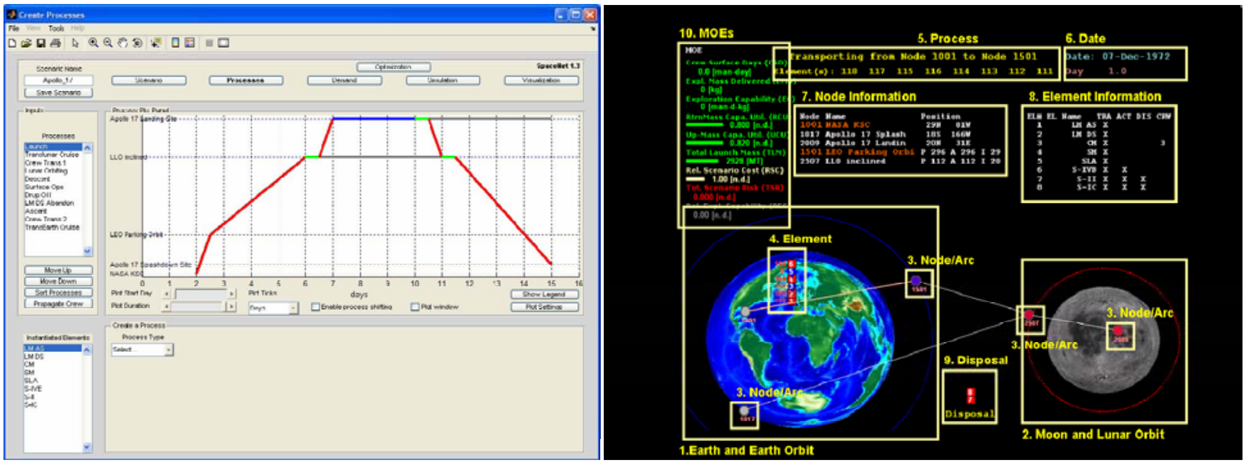

Figure 6: Visualization of a Solution

Modeling and Optimization

As missions become more complex and evolve over a period of time, defining a good, feasible architecture may become less obvious. Given the mission type, network definition, the required supplies and available elements, modeling and optimization for the interplanetary logistics problem focuses on determining low cost mission architectures that satisfy the mission requirements. The generated solution details the scheduling and assignment of supplies to vehicles for in-space transport and launch scheduling requirements.

Modeling Components

The remainder of this section provides a detailed description of the modeling development and solution procedure. Since NASA and the astronautic community use the term supply, SpaceNet is using the same term. For network modeling and algorithmic development, the term commodity is more frequently used. Thus in the reminder of the section we use commodities as the supply that needs to be transferred.

Networks

As described in previous sections, the framework for the analysis is based on a network description of interplanetary transfers. The physical network represents the set of physical locations, or nodes, and the connections, or arcs, between them. The physical nodes, or static nodes, represent the different physical destinations in space, including the origin and destination of all the commodities, as well as the possible locations for transshipment. The physical arcs, or static arcs, represent the physical connections between two nodes, that is, an element can physically traverse between these two nodes.

An example of an Earth-Moon static network is provided in Figure 7. In this figure, we can see the connections of the Earth surface nodes to the Earth orbit node, representing launches and returns. Similarly, the lunar surface nodes are connected to the lunar orbit node, representing descent and ascent trajectories. In addition, the orbit nodes, as well as the first Earth-Moon Lagrangian point are connected by in-space trajectories.

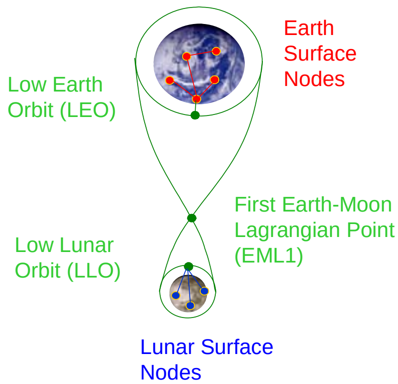

Figure 7: Depiction of an Earth-Moon Static Network

In order to analyze sequences of missions that evolve over an extended period of time, and to account for the time-varying properties that can arise in certain astrodynamic relationships, (Battin, 1999), we introduce time expanded networks. In the time expanded network, the time horizon is discretized into discrete time points. A copy of each static node is made for each of the time points and the nodes are connected by arcs according to the following rules:

- the arc must exist in the static network,
- the arc must create a connection that moves forward in time,
- the arc must represent a feasible transfer, with respect to the orbital dynamics.

Using the static network depicted in Figure 7, we can create the time expanded network in Figure 8. Here, the time expanded network is notional as not all arcs are represented, but how the trajectories evolve in time can be readily seen. Arc directions are omitted, but all arcs go forward in time.

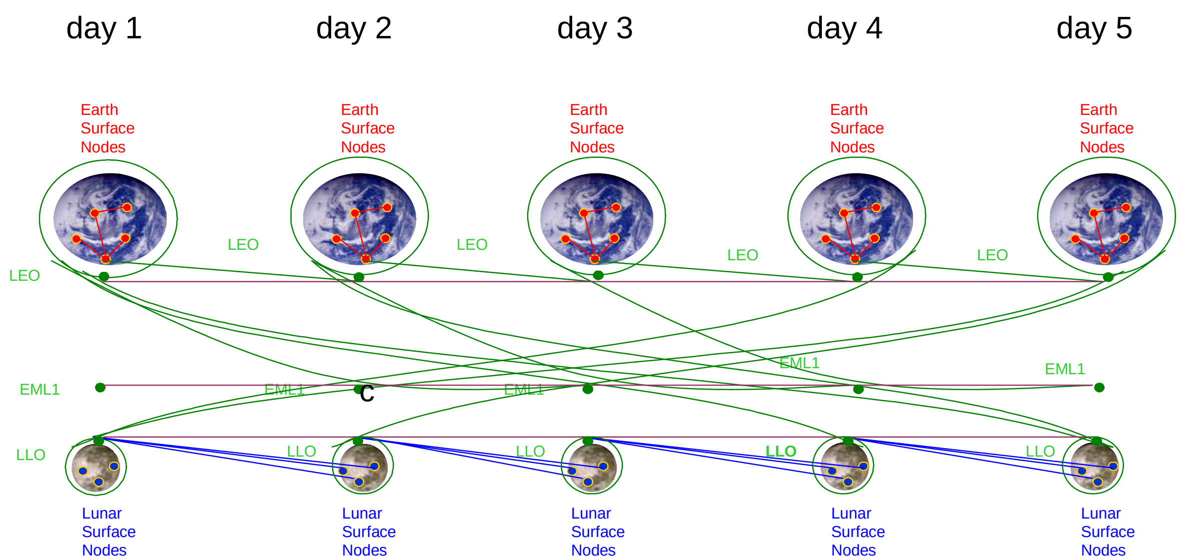

Figure 8: Depiction of an Earth-Moon Time Expanded Network

To account for the fact that on certain transfer arcs two burns occur, we slightly modify this time expanded network. Specifically, additional nodes are added on each arc, creating two distinct arcs with a single burn on each. In addition, arcs connecting the same static node at sequential time points are defined to permit waiting at a specific location for a specified amount of time. The waiting arcs do not have an associated burn for transfer.

Commodities

For the purpose of this logistics problem, a commodity is defined as a high-level aggregate of a type of supply, where the supplies are categorized as defined in previous sections. Specifically, each commodity is defined with the following parameters:

- the demand for each commodity,
- the origin node in the static network and the corresponding availability time window,
- the destination node in the static network and the corresponding delivery time window,
- any intermediate waiting segments that must be visited by the commodity, as defined by the corresponding static nodes and time periods,
- the mass and volume for a unit of the commodity.

By defining a waiting sequence as part of the commodity input, a number of wait arcs along the path can be specified, which allows on-route destinations to be designated, such as requiring a specified amount of time at a lunar location prior to return to Earth. It is important to note that in this model a crew member is treated as a commodity. In practice, crewed missions are treated differently during mission planning, however, for the purposes of this architectural design tool, crew can be considered a commodity with highly restrictive parameter values. By narrowing the availability and delivery windows for a crew commodity, the feasible shipment pathways are limited and reasonable architectures for crewed flights can be obtained.

Elements

Elements are the containment and propulsion units required to ship the supplies, or commodities, through the network. An element is classified by the amount of commodity capacity and propulsive capability it possesses. Therefore, elements are divided into two classes: non-propulsive and propulsive elements. A given element is defined by the following properties:

- the structural mass of the element,
- the mass and volume cargo carrying capacities of the element,
- the maximum fuel mass of the element, if it is propulsive,
- the cost of the element.

Decomposition

The execution of a space mission requires logistical decisions at every step. Logistical decisions are required to accumulate all of the required commodities for space missions, as well as procure and assemble all elements at the launch site. However, since at the time of launch, all of the items required to perform a space mission are co-located at the launch pad, the terrestrial logistics can be decoupled from the interplanetary logistics model. Therefore, the interplanetary logistics model encompasses all of the logistical decisions required between the launch pad and the in-space locations.

There are numerous decisions made during space missions. Although, from a system perspective, it would be desirable to make all of these decisions concurrently, due to computational limitations, this is not a tractable approach. Instead, the interplanetary logistics model is decomposed into three fundamental components: launch scheduling, manifesting, and in-space network optimization.

Launch is a highly constrained transportation activity, where besides traditional assignment and manifesting decisions, many additional constraints are necessary to model a feasible launch. For this reason the launch problem is decoupled at low Earth orbit (LEO), creating a boundary between launching and the in-space network optimization. This assumption is reasonable since for many mission architectures there exists a delay at LEO before proceeding to in-space destinations.

In-space network optimization, which is solved first, examines the entire mission design space of routing from LEO to all locations in-space. Due to the size of the time expanded network that is generated, this problem can become quite large, with hundreds of millions of variables and tens of thousands of constraints. The decision space of the in-space network optimization focuses concurrently on defining feasible routes for both commodities and elements and the assignment of elements to burns at minimum cost.

Element packing or manifesting is performed after all of the commodities and element routes have been determined. Given the assignment of commodities and elements to routes optimized in the in-space network optimization, individual commodity units are assigned to the selected elements. Constraints focusing on feasible assignments are considered while minimizing transfers between elements on route.

Network optimization and manifesting are followed by launching. The output of the network optimization and manifesting consists of a set of elements, 'filled' with specified commodities required at LEO at particular points in time. Launching focuses on selecting the appropriate elements to perform the launch (i.e., defining the launch stack that brings all the commodities and elements to LEO). It captures the payload requirements for launch, and scheduling requirements for consecutive launches.

The following sections discus all the components in the order they are applied in the overall algorithm. However, in order to understand the modeling definitions presented and for computational tractability, the following underlying assumptions are provided.

- We assume that an element that performs a burn, denoted as an active element, can only do so on consecutive burn arcs. A consecutive burn arc is defined as a series of burn arcs either directly connected or separated only by waiting arcs. Thus, once an element becomes active and performs a burn, it can either continue to be active on the next burn arc or no longer be active on any future burn arc. This assumption clearly does not affect feasibility, but only additionally constrains the feasible set in a manner that is consistent with practice.
- We assume that before every initial burn, the active element is filled to capacity with fuel and after the burns are completed, the remaining fuel is expelled. This assumption does not compromise feasibility, but it might create suboptimal solutions.
- We assume that any two elements can be docked and undocked. In addition, if any cost is associated with these operations, it is not explicitly captured. If some elements cannot be docked together, then this must be imposed in a separate post optimization analysis.

The first two assumptions eliminate the need to track the consumption of fuel by each element allocated within the network. Enforcing the final assumption eliminates the requirement of tracking the position of each element and the underlying commodities in the stack, as the stack can continually reconfigure.

Network Optimization

The in-space network model and optimization procedure is presented first, as it is the first step in the optimization process. The model is developed in three stages. First, the flow of commodities is defined and the constraints governing the commodity flows are discussed. Next, the element flows are modeled with the corresponding constraints. Finally, the constraints governing the capacity and capability, which represent the coupling constraints between the commodities and elements, are developed. The section is finished by describing the underlying model and algorithm.

Modeling Aspects

We start by first describing the model.

Commodity Flows and Constraints

In order to determine the optimal paths for each commodity to travel from origin to destination, it is first necessary to define the set of feasible paths. For a given commodity a path is feasible if it originates at the origin node during the availability time window and terminates at the destination node during the delivery time window, as defined by the commodity parameters. Furthermore, if a waiting period is defined, the path must include the corresponding waiting period. If additional restrictions, such as total travel duration are required, these can easily be incorporated into the business rules of feasible paths.

The commodity path variable determines how many units of a given commodity are on each of the commodity path. Such variable definition allows a simple way to determine if the demand of a commodity is met. Specifically, the total number of units of a commodity flowing on all commodity paths must equal the demand of that commodity.

Element Flows and Constraints

As already stated, elements can be classified as non-propulsive or propulsive, based on whether the element can carry fuel. This distinction allows for two sets of element variables to be defined. Non-propulsive element variables determine whether or not a given element travels on a given path. For propulsive element variables, it is necessary to also distinguish which set of arcs on the path, if any, does the given element provide propulsion for a stack. Clearly, for both propulsive and non-propulsive elements, all feasible paths begin at LEO. If additional requirements, such as a return to LEO or locations on the Earth's surface, are necessary, these constraints can be added as required.

As stated in the consecutive burn arc assumption, a propulsive element can be active on at most one sub-path of the path. Finally, it is also important to note that it is possible for a propulsive element to be utilized as a non-propulsive element, where the corresponding burn sequence is empty.

The element flow constraints govern the feasibility of element selections. First, any element can travel on at most one path. Additionally, for propulsive elements, we ensure that an element can provide propulsion on at most one sub-path of the path. Furthermore, it is necessary to constrain non-propulsive elements to only travel on burn-arcs if a propulsive element is collocated and providing the propulsion to traverse the given arc. A similar constraint for propulsive elements ensures that for every burn-arc in the path, exactly one propulsive element is providing the required propulsion.

Capacity Constraints

Capacity constraints ensure that the total amount of commodities (both mass and volume) present on each arc does not exceed the total capacity on the arc, which is given by the assigned elements. Since propulsive and non-propulsive elements are defined differently, it is necessary to account for elements of each type separately in the total arc capacity calculation.

Capability Constraints
The capability constraints determine if a given element has enough fuel to perform a burn, given the total mass on a burn arc. The constraints require that the total fuel of the active element performing the burn on the given sub-path be sufficient for carrying the total cumulative mass along every arc in the sub-path. Astrodynamic relationships, (Battin, 1999), are required to determine if the defined set of commodities and elements are feasible. (The rocket equation specifies that the ratio between propulsive fuel and the total mass to be transported on an arc is at least $1-\exp(-\Delta V_e / (I_m g_o))$. Here g_0 is the Earth's level gravity, I_m is the specific impulse of propulsive element m, and ΔV_e is the change in velocity on arc e.)

The Objective
Since the cost to route commodities is negligible, we include only the cost associated with elements. The objective function computes separately the total cost of both propulsive and non-propulsive elements, where the cost for fueling propulsive elements is assumed to be part of the cost coefficient.

The constraints defined do not represent an exhaustive list, but only the necessary constraints required to produce feasible solutions within the network model. Additional constraints governing the set of feasible arcs that can be traversed by a given element can be included to limit which elements travel on particular arcs (i.e., forbidding any element without a heat shield to return the Earth's atmosphere). Restrictions on commodity placement into elements can also be included for cases where feasibility is required, such as only housing crew in crew compatible elements.

Solution Methodology
The model just presented is complex and requires the implementation of a sophisticated algorithm in order to obtain good solutions. Due to the high number of variables and constraints, and the complexity of the model, a heuristic optimization method is employed. For the in-space network optimization, a series of heuristic optimization algorithms are employed to determine a complete solution to the routing and allocation problem. First, an overview of the heuristic optimization approach is presented, followed by a more detailed description of each component.

The optimization of the in-space network design problem has three components: commodity routing, element routing, and burn-arc assignment. The commodity routing is performed first, since the entire architecture is driven by the commodity demand. Next, given the commodity paths through the network, elements are assigned to paths, such that all capacity constraints are satisfied. Finally, since at this point the mass of the elements and commodities are known for each arc in the network, the propulsive element assignment can be performed. At several points within the algorithm, randomization is utilized to generate different outcomes and therefore this procedure is iterated many times to evaluate the different outcomes. Figure 9 shows the flow of the optimization algorithm.

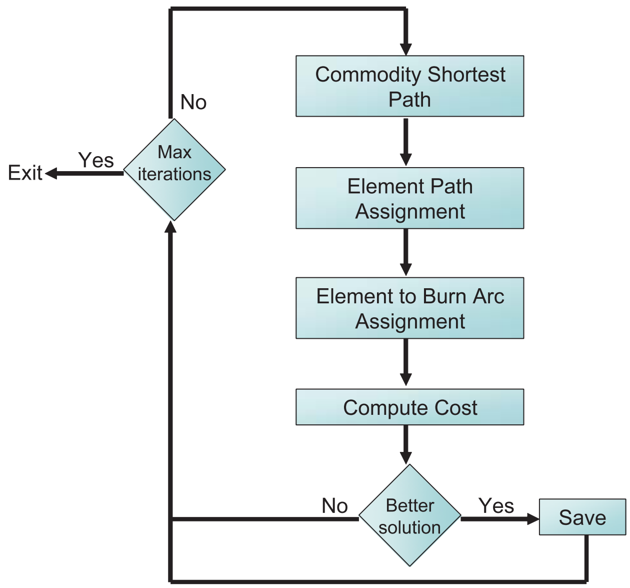

Figure 9: Flow Diagram of Heuristic Optimization

At each iteration, the heuristic determines a feasible set of commodity paths, element paths and burn-arc assignments, sequentially. If a feasible architecture is found, the cost of the architecture is computed. This cost is evaluated against the cost of the best architecture obtained thus far in the optimization process, and if a lower cost architecture is found, the solution is updated. This process is performed until the maximum number of iterations is reached. The remainder of this section provides a detailed explanation of the three components of the heuristic optimization performed in each iteration.

Commodity Routing

Commodity routing is performed by implementing a shortest path algorithm that proceeds as follows. A commodity is selected at random and an auxiliary network is constructed for each commodity. The auxiliary network connects a single source node to the nodes where a feasible path can begin and a sink node is connected to the nodes where a feasible path can terminate. For commodities that do not have a specified waiting segment, a single auxiliary network is defined where a source node connects the nodes defined by the availability interval and a sink node connects the nodes defined by the delivery interval.

Given a commodity with specified waiting segments, an auxiliary network is constructed between each segment where nodes are specified. The first auxiliary network created connects a sink node to the delivery interval. The source node is connected to the nodes in the last waiting segment, allowing for the specified duration of stay at the waiting segment. For each subsequent auxiliary network, the sink node connects to the first node in the previously defined path segment and the source node is defined as above with respect to the currently considered waiting interval. In the final auxiliary network, the source node is connected to all nodes in the availability interval. Figure 10 depicts a simple example to clarify this construction.

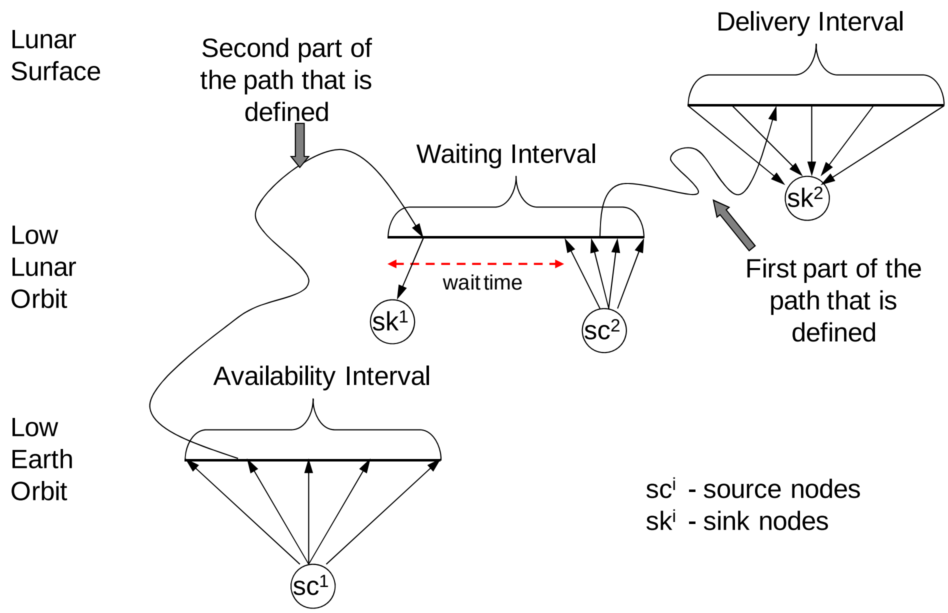

Figure 10: Auxiliary Network for a Commodity with a Single Waiting Segment

For each defined auxiliary network, a cost is assigned to every arc. For the first selected commodity, the arc cost represents the change in velocity required to travel on the specific arc. Since lower velocity change (ΔV) correlates to decreased fuel requirements, a shortest path algorithm is implemented to connect the source node to the sink node at lowest cost, or, in other words, lowest accumulated velocity change. For the remaining commodities, the arc cost is defined as a decreasing fraction of the velocity change, based on the number of times the particular arc has been chosen as an arc in another already selected commodity path. The reduction in cost for previously selected arcs reflects the desire to reuse previously selected arc.

The shortest path algorithm is run for each auxiliary network in the specified order of a given commodity until a feasible path is formed between nodes in the availability interval and in the destination interval. This process is then repeated for every commodity until all commodities have been assigned to paths.

Element to Path Assignment

After the commodity paths are determined, the element to path assignment is performed for commodity carrying elements. However, in order to perform this assignment, some preliminary manipulations are necessary. Since the network has arcs that only proceed forward in time, the nodes, and therefore arcs, can be arranged based on the topological order, see, e.g., (Magnanti, Ahuja, & Orlin, 1993). A topological order of the nodes and arcs is necessary to ensure that all assignments on downstream arcs are determined prior to the current arc assignment.

For each arc in the topological order, the following procedure is conducted to ensure that the elements assigned to the arcs for carrying commodities satisfy the mass and volume requirements on each arc. Given an arc in the topological ordering, the total mass and volume of all commodities on that arc is readily computed. To select elements to contain these commodities we first examine all already considered arcs, where these arcs are forward in time based on the topological order and their tail is the same as the head of the current arc, to determine if a previously assigned element can be reused to contain commodities on the current arc. This process is repeated until both the mass and volume capacity constraints are satisfied or until no existing elements can be utilized.

If additional capacity is required, a new element is selected by utilizing ideas from a generalized randomized adaptive search procedure (GRASP). A ‘good’ element for carrying commodities has low cost and large mass capacity. However, as the relative importance of these two properties for selecting a ‘good’ element is unknown, multiple score functions representing different balances of these properties are created. Specifically, six score functions are created and in each iteration of the heuristic, one of these score functions is selected uniformly at random and every element receives the corresponding score. For example, one score function is cost divided by mass capacity. Each score function must be increasing in cost and decreasing in mass capacity.

The probability of selecting a given element is defined as e^{-s}, where s is the element’s score, divided by the accumulated probability of all elements. Based on the score definition elements with low cost and high mass capacity are favored in the distribution. An element is then selected according to this probability distribution. The process of element selection is repeated until all the mass and volume requirements are satisfied for a given arc.

The element assignment process continues by working in backwards topological order until the mass and volume capacity constraints are satisfied on every arc. From this information, paths for each of the elements are constructed.

Element to Burn Arc Assignment

The final stage of the heuristic optimization is to assign elements to burn arcs. An element can be assigned to perform a burn if the amount of fuel available in an element is enough to satisfy the capability constraints, and can therefore provide the required velocity change, given the total mass on the arc, as defined by the rocket equation, (Battin, 1999). Since both the commodity paths and commodity-carrying element paths are known, the total mass on every arc is known.

Given an arc in the topological order, an element to burn arc assignment is performed as follows. First, forward connecting arcs are examined to determine if a previously allocated propulsive element that has already been utilized to perform a burn on a connecting burn arc can perform an additional burn. If an assignment has not been made based on this criterion, then a check of all elements on the current arc is performed to determine if a commodity carrying element could perform the burn. This second situation is distinguished from the first one because an element that has propulsive capabilities but is assigned to carry commodities is not automatically assumed to be fueled. Thus, selecting a commodity carrying element to perform the burn requires additional mass be added to the current arc and all previous arcs in the element's path, to account for the fuel of this element.

If the assignment has not yet been made for the given arc, a new element must be added to the architecture to perform the burn. A new element is selected by again employing GRASP, as described above, using the six score functions described above, but with the replacement of fuel mass capacity for mass capacity. This situation is iterated until a selected element satisfies the capability constraint for the given burn arc. Since this element is new to the architecture, it is necessary to immediately define the path of this propulsive element and update the payload mass on every arc in the path up to this current burn arc.

Manifesting

The manifesting problem is the next optimization component of SpaceNet. As the output of network optimization only determines the elements and commodities traveling on each arc in the time expanded network, and not the individual assignments of each commodity unit to specific elements, these decisions are handled in the manifesting problem. Thus, the goal of the manifesting problem is to appropriately assign individual commodity units to elements on each arc such that the capacity constraints of the elements are not violated. We stress that network optimization guarantees mass and volume restrictions at the aggregate level of commodities, but it does not provide an assignment at the individual commodity units. The manifesting problem also captures additional requirements such as not assigning particular pairs of commodities to the same element (e.g., crew together with hazardous material) or considering element capabilities for carrying certain commodities (e.g., crew can only go into a crew exploration vehicle-CEV). The objective is to keep a commodity unit in the same element as long as possible and therefore we minimize the number of transfers. The reminder of this section discusses the manifesting algorithm in greater detail.

The Manifesting Algorithm

The overall manifesting problem can be modeled as a large-scale integer program, however, we opted to develop a simpler algorithm, which is much easier to implement and is tractable. It uses integer programming but on a much smaller scale. The algorithm is iterative and it combines randomization and integer programming.

The first step in the manifesting algorithm is to assign the individual commodity units to paths in the network. For a given commodity, the network optimization provides the entire set of arcs that this commodity can travel on. Each arc in this subnetwork is assigned a capacity equal to the amount of flow of that commodity on each arc. To generate commodity unit paths of a given commodity, we start by generating a random path from the source to the destination. Since the network is acyclic, this can easily be done by randomly selecting an outgoing arc with positive capacity when building the path from the source to the sink. A unique commodity unit identifier is assigned to the underlying commodity unit and the path. After finding the first path, we adjust the capacities to reflect the fact that one unit is used on the arcs covered by the path. Now we select the second path in the similar fashion. This process terminates when all units of a given commodity have been routed, and is repeated for each commodity.

Given all commodity unit paths, we assign each commodity unit to elements on a per arc basis. We first sort the nodes in the topological order and we scan the nodes based on this order. Assignments on incoming arcs allow us to capture the number of transfers for a given assignment on an outgoing arc. On such an arc the assignment problem is an integer programming problem, which is relatively easy to solve. The model is not presented here, but it can be found in (Taylor, Klabjan, & Mamani, 2008). These steps are repeated until all arcs are scanned.

The overall algorithm iterates these steps. In each iteration, different random paths are selected, which then yields a different number of transfers at the end of the iteration. The best solution after a desirable number of iterations is selected and returned.

Launching

Network optimization and manifesting deal with operations beyond LEO and therefore it is necessary to specify how to bring all required commodity units and elements to LEO. The manifesting algorithm defines the demand for each element and commodity unit and the corresponding suggested time at LEO. Launching specifies launch vehicles or stacks and the corresponding launch times in order to bring the elements to LEO.

In the launch scheduling problem, we are given an a-priori set of demand points. Each demand point is specified by the requested time period, and the element and commodity units inside it. This input data is provided as output from network optimization and manifesting. A solution is a schedule of launch events, i.e., a series of launches over time, where in each event only a single launch vehicle is launched, see Figure 11. Each event is characterized by the launch time, the launch space port (e.g., Kennedy Space Center, Russian launch ports), and the launch vehicle type. In addition, a full description of the launch vehicle's ‘cargo’, i.e., the elements within the launch vehicle, should be provided. A solution must satisfy all of the launch vehicle mass capacity, vehicle-element permissibility, and launch pad preparation and cleanup time constraints. On the other hand, it is necessary that we meet all of the demand points or at least come as close as possible with respect to the suggested time. Among all feasible solutions the problem is to find a solution that minimizes the launch cost and penalty for deviating from demand point times. We schematically represent launching in Figure 11.

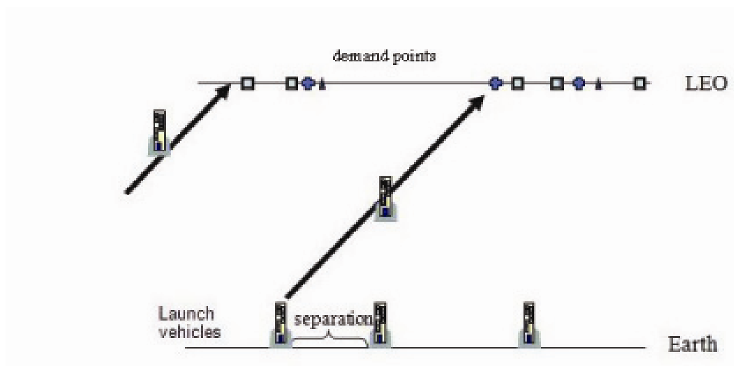

Figure 11: Launching

Due to the complexity of this problem, it is intractable to solve it to optimality. For this reason a heuristic methodology has been developed to circumvent this difficulty. We assign penalties for an element to reach LEO before or after its due date. There is a trade-off in early arrivals. They might be desirable due to robustness or undesirable due to perishability. This trade-off can be controlled by appropriately setting the penalty functions. On the other hand, late arrivals should be heavily penalized.

Launch Scheduling Formulation

To clearly define the problem, it is first necessary to obtain the set of requested ‘filled’ elements with their associated commodity units as defined by the network optimization and the manifesting algorithms. Next the set of all possible launch vehicles to be launched from Earth to LEO is determined. As is consistent with practice, we assume that the launch vehicles are disposable once having arrived at LEO and therefore cannot be reused. Finally, the locations on Earth (space ports) from which vehicles can be launched are defined as inputs to the problem.

The demand points described above are characterized by the total mass of filled elements required at LEO and the associated time they are needed. Additionally, for each element the following associated parameters are provided and are utilized to determine feasibility of the launch architecture. First, each element specifies a set of permissible vehicles that can launch that element, as is necessary when considering crew launches, or other restrictive supply. There is also an associated wait penalty per unit wait time for each element. These penalties differ by element and are specified separately for early and late arrival.

Further information required to model the launch problem characterizes the set of launch vehicles and space ports. Launch vehicles are characterized by their payload mass and volume capacities and their cost. Space ports are nodes corresponding to a launch pad at a launch site and are defined by the following properties:

- the set of permissible vehicles to be launched from the given space port,
- the time needed to prepare the pad before launching the vehicle,
- the time needed to clean-up the pad after launching a vehicle.

Launch Scheduling Optimization

The launching model defines the schedule of launches to meet the demand at LEO at lowest cost subject to constraints on the space port, and other feasibility constraints. The underlying heuristic is quite complex. The details can be found in (Taylor, Klabjan, & Mamani, 2008). Here we only outline the basic ideas.

Instead of defining all of these decisions concurrently, the main idea of the algorithm is to obtain them sequentially. The key is to first select the elements to be launched based on the due date. Next an appropriate low cost launch vehicle is selected to minimize the penalty, and at the end the launch time is computed. All these steps, which specify a single launch, are repeated until all demand points are satisfied.

Initially, filled elements are sorted so that those with closer due date are more desirable. Let D be the ordered set with respect to the due date of requested elements. In the heuristic we always launch elements in this order, e.g., if the third element in D is launched in a given solution, then also the first two elements in D have to be launched at the same time. In a single step of the heuristic, we first assign all of the elements in the list to launch vehicles in the following manner. Initially, the set of all vehicles V_{ji} that can hold the first j elements D_j in D is obtained for each j. These vehicles depend on the space port i. In forming these sets of vehicles we take into account the mass and volume restrictions of launch vehicles. Then for each j we identify a vehicle in V_{ji} with the lowest estimated cost contribution. The direct cost of launching a vehicle is combined in a multiplicative way with the sum of the clean-up and preparation times. After this step for each j we have the most promising vehicle v_{ji} that can transport the first j elements from space port i. It is intuitive that launches carrying large masses are more desirable. To offset this effect we divide the estimated cost contribution of each vehicle v_{ji} by the total mass that this launch vehicle carries. Note that this is readily available from the definition of D_j. Then, for each space port i, we choose among all vehicles $\{v_{ji}\}_j$ the best vehicle based on this adjusted cost. At this point, for each space port we have a single candidate vehicle. In the next step, by using the same cost approximation we greedily select the most promising space port, which gives us a unique vehicle and the corresponding elements in D_j. In the final step we find an optimal launch time by minimizing the total penalty of selected elements. At the end of a single iteration, the data is updated, e.g., the set of elements D yet to be transported is updated.

The algorithm then enters the next iteration and continues until all elements have been assigned for transport to LEO.

Representative Analysis

In response to President George W. Bush' announcement in 2004 NASA assembled a team of experts to conduct a system level study of future space exploration. ESAS (Stanley, Cook, & Connolly, 2005), the result of this initiative, includes a number of recommendations for accelerating the development of the CEV, implementing the new generation of spacecraft for human spaceflights, and methods for servicing the ISS. SpaceNet incorporates several lunar mission scenarios from ESAS and the Apollo campaign.

To validate and assess a potential study the proposed models were evaluated in SpaceNet on Apollo 17, which was the longest mission in the Apollo campaign, and a single lunar sortie mission from ESAS with duration of seven days. Figure 6 displays the Apollo 17 solution in SpaceNet based on the technological status in those years. Two technological requirements need to be explicitly specified: (1) any docking and undocking of elements must be crewed, and (2) only elements with a heat shield can reenter the Earth's atmosphere. Under such circumstances, the network design and system architecture recommendations based on optimization match those in Apollo 17. The situation is identical for the lunar sortie mission under the same two requirements.

We next demonstrate a possible assessment of a potential new technology by lifting the restriction on crewed dockings and undockings. Thus, in such scenarios crew is not needed at the low equatorial lunar orbit (LLO) to perform docking and undocking and to 'guard' any potential elements hovering at LLO. The results are compared in Figure 12. They clearly indicate large savings obtained by uncrewed options. The cost of an element is in millions of U.S. dollars.

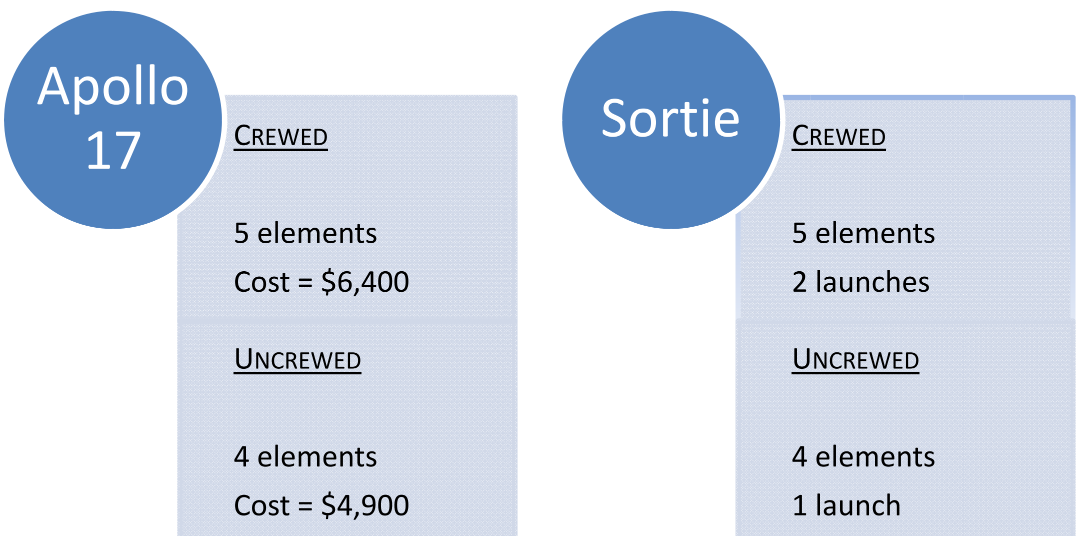

Figure 12: Crewed vs. Uncrewed Study

In addition to saving an entire element, in the sortie scenario an entire launch is also saved. NASA now easily sees the benefits of uncrewed missions and it can easier make decisions on future technological investments towards creating uncrewed dockings and undockings. Without SpaceNet NASA would have to go through the tedious task of manually recalculating the network design and system architecture.

In addition to the study of technological requirements, measures of mission effectiveness can also be obtained through SpaceNet, as displayed inFigure 13. The horizontal axis shows the total launched mass for a particular mission, while the vertical axis displays the exploration capability, which is the amount of time the crew spends performing exploration and research at all exploration nodes in the mission, multiplied by the amount of total exploration mass (the total exploration mass includes only supply that is directly needed for exploration, e.g., it excludes structural mass of all elements and fuel). The three constellation scenarios are from ESAS, where the sortie missions have already been previously introduced. The constellation campaign consists of four sortie missions that are intertwined by potentially pre-positioning supplies one mission for future missions. The lunar outpost scenario spans three years and nine missions.

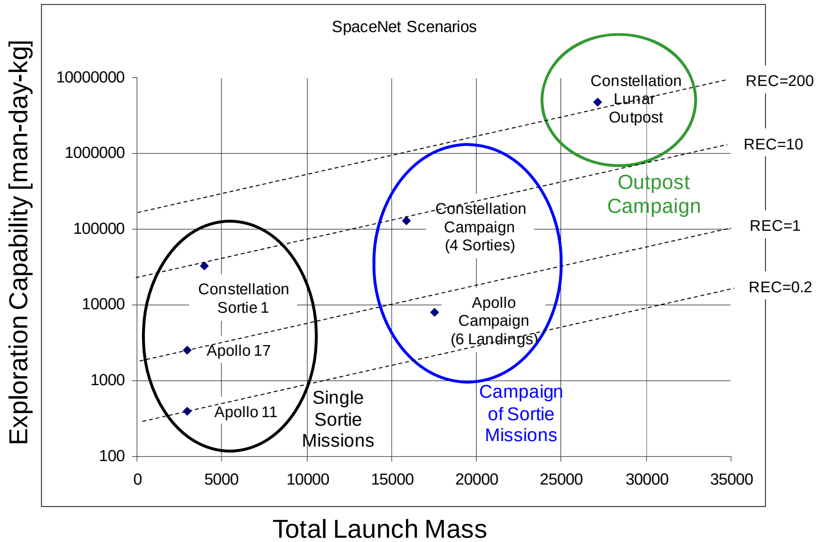

Figure 13: A Trade Study

The relative exploration capability (REC) is a normalized measure of exploration logistics efficiency. It measures the amount of productive exploration that can be done for each kilogram of mass launched from the Earth's surface, relative to Apollo 17. Apollo 17 is used as the reference case because it can be argued that of all the Apollo lunar surface missions, Apollo 17 was the most productive in terms of exploration and science and also the one that came closest to approaching the constraints imposed by flight hardware elements and operational capabilities at that time. Thus, a relative exploration capability of REC=2.5 would indicate that 2.5 times the amount of exploration capability per kilogram of mass launched can be achieved, relative to Apollo 17.

Figure 13 is augmented by showing the lines of constant relative exploration capability. The REC lines are parallel lines in this plot with iso-REC lines of higher efficiency providing more exploration capability for the same total launched mass. Single sortie missions are shown in the lower left corner, campaigns of (disconnected) sortie missions are in the middle and outpost missions are in the upper right corner. Interesting insights can be obtained from this chart. First, a set of sortie missions that all use the same transportation architecture, elements and technology should always fall into the same iso-REC line. Next, the constellation sortie missions are about 10 times more efficient than the Apollo sortie missions. This is mainly due to advances in technologies (e.g. propulsion). Another order of magnitude in logistics efficiency can be gained by reusing previously delivered exploration mass in subsequent missions, as is done in the lunar outpost scenario (REC of approximately 200). Thus, logistics strategies such as pre-deployment can also improve logistics efficiency.

Conclusions

While it is not uncommon for corporations to use network design tools, optimizing the architectural network design and logistic operations of future space explorations is an untapped application of analytics and modern decision support systems. There are several parallels between terrestrial and space logistics, but there are also important distinctions. These distinctions do not only pertain to the underlying decision support system, but they have a significant role in solution methodologies.

- To start with, the sheer size of the problem is frightening. The planning horizon spans five to ten years, which is discretized by day. This clearly leads to many time periods and large underlying networks.
- In terrestrial logistics strategic planning of fuel plays only a secondary role. When a truck leaves the origin, it can be refueled on route to its destination. In space logistics we must assure that a propulsive element has enough fuel to perform the entire burn.
- In terrestrial logistics, the travel time between two locations by a given mode of transportation is independent of the actual date. Due to the relative motion between some of the in-space nodes, the travel time between these nodes can depend on the day of the year and the actual year.
- One of the most challenging differences is the concept of gains during transportation. If a pallet is shipped in terrestrial logistics, then the same pallet arrives at the destination. In space logistics, food is consumed and thus the mass leaving a node does not equal the mass at the destination.

SpaceNet bridges the gaps between terrestrial and space logistics. It assists NASA in strategic planning of future missions and campaigns. It is a valuable tool that opens the door to many other enhancements and integrations. There are plans to completely redesign the tool and turn it into a web platform, which would clearly facilitate its usage and spread its adoption. On the other hand, there are initiatives to integrate SpaceNet with other planning tools within NASA. In particular, the terrestrial logistics operations before a vehicle is assembled at a lunch pad could be connected with SpaceNet to provide an end-to-end logistics and decision making tool. NASA is also designing an internal ontology and plans are to make SpaceNet more compliant with this ontology.

Regarding the optimization module, there is room for improvement and enhancement on several fronts. Instead of a heuristic approach to in-space network design, a more global algorithm such as branch-and-price can be implemented. The underlying mathematical formulation is presented in (Taylor, Klabjan, & Mamani, 2008) and the first version of the implementation is forthcoming. The launching heuristic can also be made more robust by directly incorporating stochastic aspects such as delayed launches.

Acknowledgments

This work was completed as part of the Interplanetary Supply Chain Management & Logistics Architectures project financially supported by NASA under contract NNK05OA50C. Prof. Olivier de Weck and Prof. David Simchi-Levi, Massachusetts Institute of Technology, serve as the principal investigators, with Dr. Martin Steele from NASA's Kennedy Space Center as Contract Officer Technical Representative. Co-investigators are Dr. Robert Shishko (JPL) and Mr. Joe Parrish (Payload Systems Inc.).

This work would not be possible without extensive contributions of Dr. Gene Lee, the main architect and developer of the graphical user interface, and Dr. Robert Shishko, who designed the database and measures of effectiveness. Both of them work at the NASA Jet Propulsion Laboratory, Pasadena, CA.

Hamed Mamani and Miao Song are acknowledged for their contributions related to optimization. Both of them are graduate students at the Massachusetts Institute of Technology. Mr Mamani spent numerous hours in implementing the launching optimization component, while Mrs. Song added several practical constraints to the network design component.

Bibliography

Battin, R. H. (1999). *An Introduction to the Mathematics and Methods of Astrodynamics.* AIAA Education Series.

Magnanti, T. L., Ahuja, R. K., & Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall.

Stanley, D., Cook, S., & Connolly, J. (2005, September 19). *The Exploration Systems Architecture Study.* Retrieved from http://www.nasa.gov/mission_pages/exploration/news/ESAS_report.html

Taylor, C., Klabjan, D., & Mamani, H. (2008). Advanced Optimization for Interplanetary Logistics. In O. de Weck, & R. Shichko, *Space Logistics, Supply Chain Management, and Operations: Opening New Frontiers.* Reston, VA: AIAA Progress in Aeronautics and Astronautics Series, American Institute of Aeronautics and Astronautics.