STOCHASTIC DESIGN OPTIMIZATION OF MODULAR, RECONFIGURABLE, PERSISTENT SUPPORT PLATFORMS IN EARTH ORBIT

BY

JEFF PEREZ

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Adviser:

Assistant Professor Koki Ho

ABSTRACT

This thesis focuses on the design and optimization of modular, reconfigurable, hosted payload platforms operating in Earth orbit. Recent advancements in on-orbit servicing technologies and robotics are creating a market for hosted-payload platforms which can support multiple payloads with varying requirements. Such platforms can employ on-orbit servicing and robotic manipulation to repair or replace modules, enhance the platform's capabilities over time, and reconfigure modules to optimize performance. Traditional spacecraft design is often driven largely by payload requirements. For the case of persistent platforms, however, not all payloads will be known in the initial design phase. This presents a unique challenge to designers, who must account for the uncertainty of future payloads by trading off between the costs of adding more capability to the platform initially, which assumes the risk of wasted costs due to over-designing the platform, and the costs of utilizing an on-orbit servicer to add capability as needed. The hosted payload platforms considered in this thesis consist of platform modules and payload modules and uses a standardized interface for intermodular and customer payload connection. Each platform module contains a critical satellite subsystem that is necessary for on-orbit functionality. As payloads are added to the platform over time, their demands may exceed the current capability of the platform, at which point additional platform modules can be added to increase the platforms capabilities. This thesis proposes an approach using a multi-stage stochastic programming method to create an initial platform design that is robust and flexible enough to support a wide range of payloads and minimizes the expected costs of future platform additions. Probability distributions for future payload selections are created based on a survey of active satellites. These distributions are then used to create samples of payload selection scenarios. Using a simple cost model, the expected costs associated with the addition of new payloads and the required platform modules are computed for each scenario in the sample. A genetic algorithm is used to find an optimal initial platform size that minimizes the combined total of the initial cost of the platform and the expected on-orbit servicing costs associated with adding future payloads and platform modules for each scenario. Platform designs are compared for a range of on-orbit servicing costs to determine the cost at which the optimizer begins to utilize servicing over adding more capability initially. Finally, a sensitivity analysis is performed to assess the variations in platform design due to the randomly selected payload scenario samples. The results of this work are a first step towards a solving a unique challenge presented by an emerging and increasingly relevant mission concept.

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Professor Koki Ho, for bringing me into his research group and for his guidance and support over the past year. Working on SYNERGEO has taught me a lot and provided me with tremendous career opportunities. I am very grateful to have had him as an advisor. I would also like to thank my previous advisor, Dr. Alex Ghosh, for bringing me into the CubeSat group the summer before my first year of graduate school. That opportunity led me to discover my passion for space systems and systems engineering and has helped guide me to where I am today. I would also like to thank Neil Hejmanowski and Dr. David L. Carroll of CU Aerospace, LLC, as well as the SYNERGEO student team for a wonderful year. I am thankful to have been a part of such a brilliant and dedicated team. A final thanks go out to my family and my girlfriend, Jessica, whose love and support gave me the motivation to complete my degree.

This work has been sponsored by the DARPA Tactical Technology Office under SBIR contract D17PC00307. Dr. Jeremy Palmer and Todd Master served as the program manager, and Dr. Brook Sullivan served as the technical point of contact.

TABLE OF CONTENTS

LIST OF ACRONYMNS

ADCS	Attitude Determination and Control System
EELV	Evolved Expendable Launch Vehicle
ESPA	EELV Secondary Payload Adapter
GEO	Geostationary Earth Orbit
HPP	Hosted Payload Platform
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
OOS	On-Orbit Servicing
PODS	Payload Orbital Delivery System
SP	Stochastic Programming

1. INTRODUCTION

1.1 Motivation

In today's space industry, there has been an ever-increasing interest for sustainable spacecraft architectures that utilize on-orbit servicing (OOS). OOS addresses several issues with the process of designing and maintaining conventional monolithic satellites. It offers the opportunity to extend the operational lifetime of a satellite by resupplying propellant or repairing faulty components, reduce cost and complexity by relaxing the constraints on fault tolerance, correct a sub-optimal orbit insertion, or to remove nonoperational satellites from useable orbits [1]. In the past few years, architectures utilizing OOS have become even more relevant as significant progress has been made towards the development of OOS technologies. Several missions, such as DARPA's Robotic Servicing of Geostationary Satellites program and Northrop Grumman's Mission Extension Vehicle, seek to progress the capabilities of autonomous servicing spacecraft by demonstrating on-orbit servicing on active geostationary satellites in the near-term [2] [3]. A necessary aspect of OOS is the use of robotic manipulators capable of autonomously performing any tasks required to service or assemble satellites. Several robotic manipulators have already gained experience on-orbit [4] [5], or will soon be launched [6]. Additionally, NASA's Dragonfly project aims to develop a robotic manipulator which will enable robotic self-assembly of satellites in Earth orbit [7].

The developing technologies mentioned above lay the foundation for a new type of architecture known as a hosted payload platform (HPP). Hosted payload platforms are persistent satellite platforms which are designed to support payloads by providing all necessary subsystems of a satellite. Payloads can be added or removed from the platform as necessary using a robotic servicing spacecraft, henceforth referred to as a "servicer". Without the need to develop a dedicated satellite platform for a payload, universities or small businesses could instead focus solely on designing a modular payload and would not incur the costs and schedule delays normally associated with commissioning a full satellite mission [8] [9]. As a result, space would become more accessible to a wider range of customers. Hosted payload platforms would also help to alleviate the overcrowding of the space environment. Platforms hosting multiple payloads in GEO would help to maximize the number of remaining orbit locations available for commercial use, while platforms in LEO would help to reduce space debris as a servicer could deorbit the payloads it removes from the platform [10]. With the increased frequency of launch opportunities seen in recent years, hosted payload platforms would be able make use of the more rapid response times to repair or replace faulty components and restore full capabilities of the platform before functionality is lost.

1.2 Hosted Payload Platform Mission Architecture

The hosted payload platforms considered in this thesis consist of platform modules and payloads. Each platform module contains a critical satellite subsystem that is necessary for on-orbit functionality. This thesis considers four types of platform modules: Attitude Determination and Control Systems (ADCS), communications/command and data handling, power, and propulsion. The design of each platform module adheres to a specific form factor. The four form factors considered in this work are the Payload Orbital Delivery System (PODS) and PODS-Extended from Space Systems Loral [11] and the Evolved Expendable Launch Vehicle (EELV) Secondary Payload Adapter (ESPA) and ESPA Grande form factors from Moog CSA Engineering [12]. These form factors were chosen because they were designed to be launched as a secondary payload on larger launch vehicles, resulting in reduced launch costs and more frequent launch opportunities, and because their structural design facilitates modularity. The form factor specifications are described in more detail in Section 3.1.

It is assumed that a servicer will be responsible for the delivery of modules from their launch orbit to the platform, assembly of new modules to the platform, reconfiguration of the platform, and refueling of propulsion modules. The modules are assumed to be connected via a standardized interface mechanism that supports inter-module power and data connections. The modular design of the platform allows it to employ on-orbit servicing and robotic manipulation to repair or replace modules, enhance the platform's capabilities over time by adding platform modules, and reconfigure modules to optimize performance.

1.3 Stochastic Payload Selection

Traditional spacecraft design is often driven largely by payload requirements. For the case of hosted payload platforms, however, not all payloads will be known in the initial design phase. The uncertainty of future payload selection presents a unique challenge to designers. An HPP must be designed so that it can support almost any combination of unknown payloads with varying requirements. On-orbit satellite servicing is not an inexpensive service by any means, although the cost can be reduced in the long term through the advancement of servicing technologies and the development of space infrastructures to support servicing. Because of the high cost of servicing, a designer must tradeoff between adding more capability to the platform in the initial phase and adding more capability on an as-needed basis. In the latter case, a servicer will add more capability to the platform as payloads are determined. This would incur higher servicing costs but would prevent the platform from being given more capability than necessary. In the former case, the more-capable initial platform can be assembled before it is launched which reduces dependence on a servicer and therefore reduces cost. However, creating an initial platform design with higher capability introduces the risk of over-designing the platform. For example, if the platform is designed for the worst-case scenario in which it must support multiple payloads with high mass and power requirements, there is a chance that the actual payloads it will support require very little capability, resulting in wasted costs from launching unused mass to orbit. This thesis classifies the problem of uncertain payload selection for HPPs as a stochastic programming (SP) problem and attempts to solve the problem by applying SP methods found in the literature.

1.4 Thesis Overview

This thesis aims to apply stochastic programming methods to solve the problem of uncertain payload selection for HPPs by optimizing the initial platform design to find a solution that is feasible for most or all possible combinations of payloads while minimizing the total cost of launching and servicing the platform as it evolves within an on-orbit servicing framework.

Chapter 2 of this thesis develops a background in stochastic programming methods, problem formulations, decision-making processes, and describes how to interpret results. Chapter 3 describes the hosted payload platform mission architecture used in this work, along with the methodology used to apply stochastic programming methods to the problem of hosted payload platform design. Chapter 4 presents the results of the stochastic programming methods used and discusses the significance of these results. Finally, Chapter 5 summarizes the findings of this thesis and outlines several directions for future work on this topic.

2. STOCHASTIC PROGRAMMING

Stochastic programming is an optimization approach for modelling problems that involve some degree of uncertainty, or randomness. Unlike deterministic optimization problems, which are formulated using known data, stochastic optimization problems need to include parameters that are unknown at the time a design decision needs to be made. For many SP problems, including the problem of hosted payloads, a probability distribution for the uncertain parameters is known or can be estimated. The objective is then to create a decision-making framework that leads to a solution that is feasible for all or most realizations of the uncertain parameters, which are characterized by their probability distributions, and performs well on average [13]. Some fields where SP has been useful include capacity planning, production planning, transportation and logistics, and financial management [14].

2.1 Classes of Stochastic Programs

Mitra [15] categorizes SP problems into three main classes: distribution, chance constraint, and recourse problems. Distribution problems are solved by varying inputs to obtain a distribution of solutions or objective function outcomes to the SP. This type of problem is considered to be the equivalent of a sensitivity analysis in linear programs and is used to determine the robustness of the model. Chance constraint problems are formulated to ensure that the probability of satisfying a constraint is above a specified level. This formulation restricts the feasibility region which leads to a high level of confidence in the solution. Recourse problems are problems where a decision is made before a random event occurs followed by a recourse decision made after a random event is realized which is meant to make any corrections to the previous decision. The initial decisions are made using knowledge of the probability distributions associated with the random events while the recourse decisions are made with the new knowledge the realization of a random event. The problem of stochastic payload selection for hosted payload platforms falls into the recourse problem class of SPs and is described in further detail in the following sections.

2.2 Two-Stage Stochastic Programming

In two-stage stochastic programming, two decisions are made. The first-stage decision is made based on available data and does not depend on future observations. This decision is made to optimize the cost of the first-stage decision plus the expected cost of an optimal second-stage decision. The second-stage decision is another optimization problem to find the best recourse action to take following the realization of the uncertain data. Following the formulation of Shapiro and Philpott [13], the standard form for a two-stage SP is shown below in Equation (1)

$$\min_{x} g(x) = c^T x + E[Q(x, \xi)] \tag{1}$$

Where x is the first-stage decision vector, $c^T x$ is the cost of the initial decision, ξ is the vector containing the uncertain data, and $Q(x, \xi)$ is the optimal value of the second-stage problem

$$\begin{gathered} \min_{y} = q^T y \\ s.t. \;\; Tx + Wy \leq h \end{gathered} \tag{2}$$

In Equation (2), y represents the second stage decision vector, the term $q^T y$ represents the cost of the recourse decision, and $q, T, W,$ and h contain data from the second stage. In the constraint of Equation (2), the Wy term makes a correction for any inconsistencies in the system $Tx \leq h$. If the uncertain vector ξ has K finite realizations, also referred to as scenarios, with known probabilities p_k for $k = 1, \ldots, K$, then the expected value for the optimal second-stage decision can be discretized as

$$E[Q(x, \xi)] = \sum_{k=1}^{K} p_k Q(x_k, \xi_k) \tag{3}$$

Now, the two-stage problem can be reformulated as a single optimization problem as seen in Equation (4) below.

$$\begin{aligned} &\min_{x,y_1,\ldots,y_K} c^T x_k + \sum_{k=1}^{K} p_k q_k^T y_k \\ s.t.\ \ &T_k x_k + W_k y_k \le h_k\,, \qquad k = 1, \ldots, K, \\ &x_1 = x_2 = \cdots = x_K \end{aligned} \tag{4}$$

The last constraint of Equation (4) is known as the non-anticipativity constraint, which forces the initial decision to be identical for every scenario. Without the non-anticipativity constraint, the decision variables x_k would be allowed to depend on a realization of the uncertain data at the second stage which is not suitable for the two-stage decision model.

2.3 Multi-stage Stochastic Programming

Multi-stage SPs can be viewed as an extension of two-stage SPs [13]. This type of program is useful in problems where probabilistic data is realized sequentially over certain periods of time. Each time a random parameter or outcome is observed, that data becomes available to the decision-maker and a recourse decision is made. In multi-stage SPs, decisions made at each stage must account for previous decisions as well as the residual uncertainty at every future stage.

Scenarios in MSSPs are created from a sequence of realizations of a random variable. This sequence is represented by the vector $\xi = \xi^1, \ldots, \xi^N$, where N is the total number of stages. Scenario trees are used to represent the branching process made by the realizations of ξ. An example of a scenario tree is shown below in Figure 1.

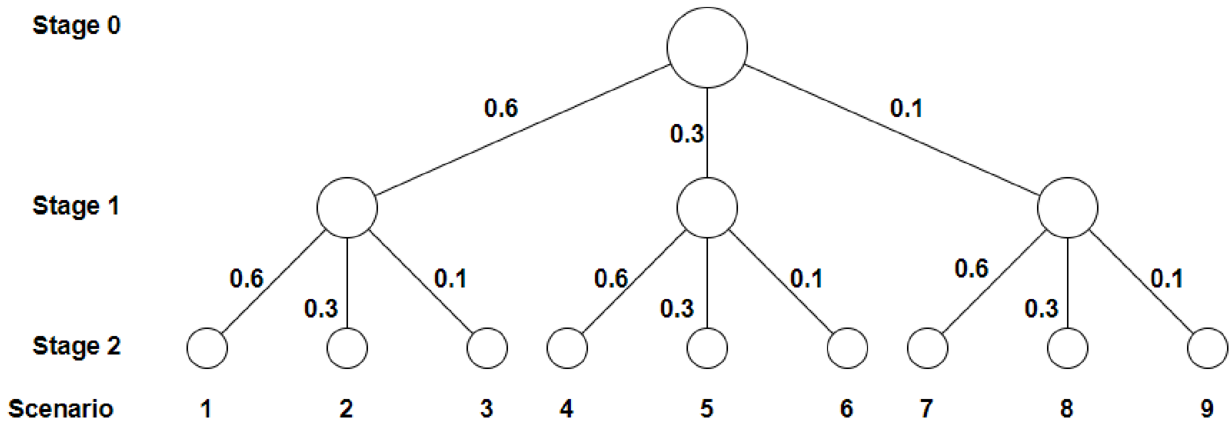

Figure 1: Scenario Tree Example

The circles represent nodes where decisions are made, and each node is connected by a branch. The first node is called the "root" node and all others are "children" of the previous nodes. The numbers associated with each branch represent the probability of that realization occurring. In the example presented above, there are two stages and three possible outcomes after stage, resulting in 3^2 possible scenarios. To obtain the probability of a specific scenario occurring, one would multiply the probabilities of every branch which lead to the scenario. For example, the probability of scenario 3 in Figure 1 occurring is $P = 0.6 * 0.1 = 0.06$.

Following a realization of ξ, a corresponding decision is made. The decision vector is represented by $x = x^0, x^1, \dots, x^N$, where x^n is the decision made at the n$^{\text{th}}$ stage. The general order of events is as follows:

1. Initial decision x^0 is made, accounting for the uncertainty $P(\xi^1, \dots, \xi^N)$at every future stage
2. Random variable ξ^1 is observed
3. Decision $x^1(x^0, \xi^1)$ is made, accounting for the uncertainty $P(\xi^2, \dots, \xi^N)$
4. Random variable ξ^2 is observed
5. Decision $x^2(x^0, x^1, \xi^1, \xi^2)$ is made, accounting for the uncertainty $P(\xi^3, \dots, \xi^N)$

This process continues until ξ^N is observed and the final recourse decision x^N is made.

As mentioned above, each decision can only depend on the previous decisions and the previous realizations of uncertain data. During the planning stage, decision makers can consider as many scenarios as desired, but when decisions are made, they cannot depend on realizations that have not yet been observed [14]. In order to force decisions to have no dependence on future realizations, the non-anticipativity constraint from the two-stage formulation can be extended to the multi-stage case. This constraint is formulated as

$$\begin{aligned} x_k^0 = x_j^0, \qquad & \forall k, j \in \{1, \dots, K\} \\ x_k^n = x_j^n, \qquad & when \;\; \xi_k^1, \dots, \xi_k^n \equiv \xi_j^1, \dots, \xi_j^n, n \neq 0 \end{aligned} \tag{5}$$

Where k represents the scenario and K represents the total number of possible scenarios. As an example, when applied to the scenario tree in Figure 1 with nine scenarios, the constraint yields

$$x_1^0 = x_2^0 = \cdots = x_9^0$$

$$x_1^1 = x_2^1 = x_3^1$$

$$x_4^1 = x_5^1 = x_6^1$$

$$x_7^1 = x_8^1 = x_9^1$$

Now, the optimization problem formulation for a multi-stage SP with a finite number of scenarios and discrete probabilities can be formulated as follows:

$$\begin{aligned} & \min_x \sum_{k=1}^{K} p_k \sum_{n=1}^{N} {c_k^n}^T x_k^n \\ & s.t. \;\; x_k^0 = x_j^0, \qquad \forall k, j \in \{1, \dots, K\} \\ & x_k^n = x_j^n, \qquad when \;\; \xi_k^1, \dots, \xi_k^n \equiv \xi_j^1, \dots, \xi_j^n, n \neq 0 \end{aligned} \tag{6}$$

3. METHODOLOGY

3.1 Hosted Payload Platform Mission Architecture

3.1.1 Platform Module Design

The hosted payload platform considered in this work consists of platform modules and payload modules, all of which are connected via a standardized interface mechanism that supports power and data transfer between modules. Each platform module contains a critical satellite subsystem that is necessary for on-orbit functionality of the platform. Although there are many ways to define the platform modules, this thesis considers four types. The different types of modules and their core functions are shown in Table 1 below. When the demand from the payload modules exceeds the platforms current capabilities, additional modules are added to meet the requirements of the payloads.

Table 1: HPP module types

Module	Functions
ADCS	Attitude determination and control for platform
Communications	Transmission of telemetry and payload data; command and control for station as a whole
Power	Power generation, storage, and distribution
Propulsion	Orbital station-keeping and momentum dumping

3.1.2 Module Form Factors

Four module form factors have been considered for this work. Two are based off the PODS and PODS-Extended designs from Space Systems Loral [11] and the remaining two are based off the ESPA and the ESPA Grande designs from Moog CSA Engineering [12]. These form factors were chosen primarily for their ability to be launched as a secondary payload, which leads to more frequent and less expensive launch opportunities. Also, the size ranges between the form factors allow for a wide range of potential payloads, as modules that are too small would not be able to support large payloads and modules that are too large would be inefficient. The physical characteristics and limitations of the form factors are presented in Table 2.

Table 2: Form factor physical characteristics [11] [12]

Form-Factor	Company	Mass limit (kg)	Length (m)	Width (m)	Height (m)	Volume (m^3)
PODS	SSL	75	1	0.5	0.4	0.20
PODS Extended	SSL	150	1	1	0.6	0.60
ESPA	Moog CSA Eng.	180	0.97	0.71	0.61	0.42
ESPA Grande	Moog CSA Eng.	320	1.42	1.17	1.07	1.78

In this work, the HPP is assumed to consist modules of a single form factor. A standardized form factor is ideal because it lowers the complexity of module design and better facilitates assembly and reconfiguration of modules on-orbit. However, varying form factors would lead to an interesting optimization problem in which the form factor would be included as a decision variable. This problem will to be left for future work, and the module form factor is left as a constant parameter input by the user.

3.1.3 Servicing Infrastructure and Assumptions

As the satellite servicer and servicing architecture are not the focus of this thesis, several assumptions are made about the servicer's capabilities. One assumption is that the servicer will have the ability to refuel depleted propellant tanks. Refueling a propellant tank is likely to be more cost effective than launching a completely new module to the platform. Additionally, the use of fuel depots in space is a major goal of future on-orbit servicing infrastructures, so it follows that an HPP should employ this capability. Another assumption is that the servicer will be capable of bringing multiple modules to the platform at a time, assuming that the modules have already been connected pre-launch. A third assumption is that there will be one dedicated servicer for an HPP, and the servicer is already on-orbit when the initial platform is launched. Having a dedicated servicer would result in shorter periods in which the recently launched modules are waiting to be collected by the servicer and brought to the platform. Finally, it is assumed that payloads will remain on the station for the entire lifetime of the platform. In reality, if a payload were to complete its objectives and no longer needed to be operated or if the payload failed irreparably, the servicer could remove the payload module from the platform to lower the mass and create more space for additional payloads.

3.1.4 Platform Design

Following the module design, the platform is designed according to a decision-making process. The first step of the process is to choose an initial number of platform modules to assemble and launch to the desired orbit. The method for designing an optimal initial platform is one of the main topics of this thesis and is discussed in Section 3.2. Following the launch of the initial platform, the first payload is selected. Then, the platform designer calculates the number of additional modules that are required for the platform to be self-sufficient for the user-defined lifetime. The payload and its additional required platform modules are then launched to orbit, where they are picked up by the servicer, brought to the platform, and integrated with the existing modules. After a user-defined time interval, a new payload is selected, and the platform module additions required for the new platform to be self-sufficient for the remainder of the lifetime are computed. At this stage, and all subsequent stages, fuel consumption for the platform is calculated and accounted for in the platform designer. If a propulsion module has expended all of its propellant, the servicer will refuel the propellant tank when it arrives with the new modules. This process continues for a user-defined number of payload addition stages.

3.2 Stochastic Programming

3.2.1 Optimization Problem Formulation

The problem being considered in this thesis is formulated as a single-objective, multi-stage programming problem. The objective is to find an optimal initial decision for the platform design that minimizes the average value for the cost of launching, maintaining, and servicing a hosted payload platform for a given set of input parameters, which include orbit, lifetime, module form factor, launch interval, and total number of payloads to be added to the platform over its lifetime. It is important to note that this formulation is slightly different from other multi-stage SPs. Typically, each decision made during the decision-making process is optimized based on the residual uncertainty of remaining future stages. In the problem considered in this work, only the initial decision is optimized, and all subsequent decisions are computed using the systems engineering HPP designer tool. These decisions are made to make corrections to the platform after the realization of an uncertain payload.

There are two design variables in this problem. The first variable, ξ, is a stochastic variable defined as:

$$\boldsymbol{\xi} = [\xi_k^1, \xi_k^2, \dots, \xi_k^N] \tag{7}$$

In Equation (7), ξ_k^n is an integer corresponding to the payload that is randomly selected at the n^{th} stage of the k^{th} scenario based on a discrete probability distribution. The second design variable, $\boldsymbol{x}$, is the decision variable. It is a vector containing integer values for the quantity of each type of module added at the beginning of each stage and is defined as follows:

$$\boldsymbol{x}_k^n = [x_1, x_2, x_3, x_4, x_5]_k^n \tag{8}$$

In Equation (8), x_i for $i = \{1, \dots, 5\}$ is the number of ADCS modules, communication modules, power modules, propulsion modules, and propellant resupplies, respectively. The subscript k represents the scenario number.

Constraints on the decision variable $\boldsymbol{x}$ are fairly straightforward. The platform requires at least one of each type of platform module in order to be fully functional. Also, the number of propellant resupplies must be 0 for the initial decision and greater than or equal to 0 for all subsequent decisions. This yields the constraints shown in Equation (9) below.

$$\begin{gathered} {x_i}_k^n \geq 1, \qquad i \in \{1, 2, 3, 4\}, \forall k, n \\ {x_5}_k^0 = 0, \qquad \forall k \\ {x_5}_k^n \geq 0, \qquad n \in \{1, \dots, N\}, \forall k \end{gathered} \tag{9}$$

For multi-stage SPs, as mentioned in Section 2.3, the decision made at the previous stage must be constant for all future stages across all scenarios which branch from a common node. As a result, the non-anticipativity constraint is imposed on $\boldsymbol{x}$ and formulated as follows:

$$\begin{gathered} \boldsymbol{x}_k^0 = \boldsymbol{x}_j^0, \qquad \forall k, j \in \{1, \dots, K\} \\ \boldsymbol{x}_k^n = \boldsymbol{x}_j^n, \qquad when \;\; \xi_k^0, \dots, \xi_k^{n-1} \equiv \xi_j^0, \dots, \xi_j^{n-1}, n \neq 0 \end{gathered} \tag{10}$$

Following Equation (6), the final optimization problem is formulated as follows:

$$\begin{gathered} \min_{\boldsymbol{x}} \left[C(\boldsymbol{x}_k^0) + \sum_{k=1}^{K} p_k \sum_{n=1}^{N} C(\boldsymbol{x}_k^n) \right] \\ s.t. \quad \boldsymbol{x}_k^0 = \boldsymbol{x}_j^0, \qquad \forall k, j \in \{1, \dots, K\} \\ \boldsymbol{x}_k^n = \boldsymbol{x}_j^n, \qquad when \;\; \xi_k^0, \dots, \xi_k^{n-1} \equiv \xi_j^0, \dots, \xi_j^{n-1}, n \neq 0 \\ {x_i}_k^n \geq 1, \qquad i \in \{1, 2, 3, 4\}, \forall k, n \end{gathered} \tag{11}$$

$$x_{5}{}_{k}^{0} = 0, \qquad \forall k$$

$$x_{5}{}_{k}^{n} \geq 0, \qquad n \in \{1, \dots, N\}, \forall k$$

3.2.2 Payload Survey Probability Functions

To characterize the uncertainty of payload selection for the HPP, a probability distribution is required. To obtain a probability distribution for potential payloads, a satellite database, created and maintained by the Union of Concerned Scientists [16], was leveraged. The database contains data for almost 2,000 active satellites currently in orbit around Earth. In this thesis, it is assumed that the current satellites in orbit are an accurate representation of future payloads that an HPP might support. Realistically, this assumption may not be entirely accurate since the market is constantly changing, so predictions for the payload market in the future may also be required to obtain a more accurate probability distribution.

For each satellite in the database, there is data for a variety of categories, including the satellites orbit class, orbit type, payload purpose, mass, power, and many others. Table 3 and Table 4 below shows the breakdown of each class of orbit into its different types and different satellite purposes, along with the associated percentage of each respective class.

Table 3: Orbit class breakdown by orbit type

Orbit Class	Orbit Type	Quantity	% of Class
Elliptical	Cislunar	1	2.22%
	Deep Highly Elliptical	9	20.00%
	Molniya	17	37.78%
	Non-Polar Inclined	2	4.44%
	Other	16	35.56%
GEO	Geostationary	558	100.00%
LEO	Elliptical	8	0.65%
	Equatorial	20	1.63%
	Non-Polar Inclined	264	21.46%
	Polar	191	15.53%
	Sun-Synchronous	730	59.35%
	Other	17	1.38%
MEO	Equatorial	16	12.90%
	Non-Polar Inclined	90	72.58%
	Other	18	14.52%

Table 4: Orbit class breakdown by satellite purpose

Orbit Class	Purpose	Quantity	% of Class
Elliptical	Communications	10	22.22%
	Earth Observation	10	22.22%
	Navigation	2	4.44%
	Space Science	20	44.44%
	Technology Demo	3	6.67%
GEO	Communications	477	85.48%
	Earth Observation	42	7.53%
	Navigation	28	5.02%
	Space Science	5	0.90%
	Technology Demo	6	1.08%
LEO	Communications	273	22.20%
	Earth Observation	658	53.50%
	Earth Science	25	2.03%
	Space Science	60	4.88%
	Technology Demo	214	17.40%
MEO	Communications	17	13.71%
	Navigation	107	86.29%

From Table 3, it is observed that the majority of satellites in the elliptical orbit class are either government satellites in Molniya orbits or satellites with highly elliptic orbits, which typically have very strict and specific science requirements that require such an orbit. From Table

4, it can be observed that a large majority (86%) of MEO satellites are navigation satellites, with almost all of them being a part of a constellation. Therefore, it is assumed that satellites in MEO or elliptic orbits do not accurately represent the type of payloads that would benefit from a hosted payload platform and are excluded from the probability distribution. Thus, the focus for this work shifts to HPPs in both LEO and GEO. A case study for HPPs in both a sun-synchronous low Earth orbit and geostationary orbit will be used to test the HPP design tool and stochastic optimization methods.

The current probability distribution for satellite purposes in the two classes of orbits under consideration is not yet sufficient to characterize payloads. Within each satellite purpose category, there is a wide variety of satellites, ranging from 5 kg CubeSats to 10,000 kg national security satellites. In order to create more accurate representations of potential payloads, satellites in each "satellite purpose" category for LEO and GEO were broken down further into several mass ranges. Several simplifying assumptions have been made to narrow the scope of the payloads considered for the probability distribution, thus reducing the total number of possible scenarios and computational cost. Due to the size of the form factors under consideration in this study, satellites with a launch mass less than 25 kg and greater than 4000 kg were excluded from the data for the payload probability function. Satellites under 25 kg are mostly CubeSats of size 12u and smaller. Due to the development of low-cost commercial off-the-shelf CubeSat buses and subsystems along with the increasing number of ride-share opportunities [17], it was assumed that the cost of using a servicer to bring these small payloads to the platform would exceed the cost of developing a dedicated CubeSat platform to support the payload. Similarly, it was assumed that the massive payloads supported by satellites more than 4,000 kg would require an excessive amount of platform modules in order to have enough capability to support them, which would drive up the cost and add a considerable amount of parasitic mass to the platform. The remaining satellites were then categorized into the following mass ranges: 25-99 kg, 100-499 kg, 500-999 kg, 1000-1999 kg, 2000-2999 kg and 3000-3999 kg, as seen in Table 5 and Table 6 below.

Table 5: Geostationary satellite breakdown by satellite type

Satellite Type	Survey Data	500-999 (kg)	1000-1999 (kg)	2000-2999 (kg)	3000-3999 (kg)
Comms	Quantity	1	32	73	107
	Probability (%)	0.47%	15.02%	34.27%	50.23%
	Avg Power (W)	1500	1860	4655	5851
	Avg mass (kg)	950	1523	2481	3388
Earth Observation and Earth Science	Quantity	N/A	7	12	6
	Probability (%)		28.00%	48.00%	24.00%
	Avg Power (W)		550	1434	2420
	Avg mass (kg)		1454	2264	3413
Navigation	Quantity	2	8	3	2
	Probability (%)	13.33%	53.33%	20.00%	13.33%
	Avg Power (W)	1500	1623	2000	6800
	Avg mass (kg)	800	1426	2233	3800
Space Science	Quantity	4	N/A	N/A	1
	Probability (%)	80.00%			20.00%
	Avg Power (W)	600			1500
	Avg mass (kg)	700			3100
Technology Demo	Quantity	N/A	N/A	1	1
	Probability (%)			50.00%	50.00%
	Avg Power (W)			2100	4142.75
	Avg mass (kg)			2650	3800

Table 6: Low Earth orbit satellite breakdown by satellite type

Satellite Type	Survey Data	25-99 (kg)	100-499 (kg)	500-999 (kg)	1000 - 1999 (kg)	2000 - 2999 (kg)	3000 - 3999 (kg)
Comms	Quantity	31	69	117	N/A	N/A	N/A
	Probability (%)	14.29%	31.80%	53.92%			
	Avg Power (W)	160	482	397			
	Avg mass (kg)	43	263	790			
Earth Observation and Earth Science	Quantity	65	86	49	60	30	2
	Probability (%)	22.26%	29.45%	16.78%	20.55%	10.27%	0.68%
	Avg Power (W)	67	343	939	1493	3152	1950
	Avg mass (kg)	61	251	707	1382	2513	3525
Space Science	Quantity	3	9	4	2	1	N/A
	Probability (%)	15.79%	47.37%	21.05%	10.53%	5.26%	
	Avg Power (W)	39	188	703	940	750	
	Avg mass (kg)	73	237	658	1657	2500	
Technology Demo	Quantity	20	17	3	2	2	N/A
	Probability (%)	10.70%	9.09%	1.60%	1.07%	1.07%	
	Avg Power (W)	89	79	575	1000	1951	
	Avg mass (kg)	53	154	639	1360	2240	

As seen in Table 5 and Table 6 above, the probability that a satellite would fall into each of the mass ranges was found, along with the average mass and power of a satellite in that range. Only satellites with available mass data were used to find the probabilities. Some satellite types had no occurrences in certain mass ranges, which is indicated in the tables by "N/A". Also, it should be noted that due to the small sample size of Earth science satellites, the Earth observation and Earth science categories were combined to help reduce the scenario set. Each box under a mass range in Table 5 and Table 6 represents one possible choice for a payload selection, for a total of 15 choices for GEO and 19 choices for LEO.

Now that the satellite information for each payload choice has been obtained, the corresponding payload information can be estimated. With such a large list of satellites, it would not be efficient to look up specific payload information for each satellite. Instead, several mass and power fractions were obtained using historical satellite data from SMAD [18]. These fractions are shown in Table 7.

Table 7: Payload mass and power fractions

	LEO			GEO			
	Comms	Remote Sensing	Average for LEO	Comms	Remote Sensing	Navigation	Average for GEO
Satellite dry mass fraction of launch mass	0.9	0.96	0.91	0.83	0.96	0.94	0.91
Payload mass fraction of satellite dry mass	0.27	0.35	0.31	0.27	0.35	0.21	0.32
Payload power consumption fraction of satellite power	N/A		0.46	N/A			0.35

3.2.3 Scenario Generation

Even with the reduced payload set, there are still 15 potential payload selections for GEO and 19 potential selections for LEO. This results in 15^N or 19^N total scenarios for each case. As the number of payloads increases, the total number of scenarios quickly grows to a level that is computationally infeasible. Instead of trying to compute every possible scenario, a sample of scenarios is randomly selected using the probability distributions that were created from the payload survey. Every time a sample is taken, the payloads will differ slightly from previous samples. This can lead to slightly different results for various samples. Additionally, the number of scenarios used in the sample will also likely affect the results. As the number of scenarios used in the sample increases, the variance from sample to sample is expected to decrease. The sensitivity can be characterized using a Monte Carlo analysis, in which the same simulation is run for many samples of the same size to determine variations in the solutions. The same analysis can then be re-run for different sample sizes to further characterize the effect of sample size on the solution.

3.2.4 Hosted Payload Platform Designer

The HPP designer is a MATLAB-based code used to design the platform and make decisions on how the platform needs to change to accommodate new payloads that have been added. The designer begins by taking inputs from the user for the desired platform lifetime, desired orbit (LEO or GEO), module form factor, scenario sample size, new payload launch interval, and total number of payloads to be added to the platform. Next, the platform modules are designed, followed by the generation of a sample of randomly selected scenarios. Each scenario consists of the payloads that have been randomly selected, along with the corresponding payload data for each payload selection. The probability of each scenario occurring is also computed and stored with the data. After a sample of scenarios have been generated, the designer enters the optimization and decision-making loop. The optimization problem is defined by Equation (12) below.

$$\begin{gathered}
\min_{\boldsymbol{x}} \left[C(\boldsymbol{x}_k^0) + \sum_{k=1}^{K} p_k \sum_{n=1}^{N} C(\boldsymbol{x}_k^n) \right] \\
s.t. \;\; \boldsymbol{x}_k^0 = \boldsymbol{x}_j^0, \qquad \forall k, j \in \{1, \dots, K\} \\
\boldsymbol{x}_k^n = \boldsymbol{x}_j^n, \qquad when \;\; \xi_k^1, \dots, \xi_k^n \equiv \xi_j^1, \dots, \xi_j^n, n \neq 0 \\
{x_i}_k^n \geq 1, \qquad i \in \{1, 2, 3, 4\}, \forall k, n \\
{x_5}_k^0 = 0, \qquad \forall k \\
{x_5}_k^n \geq 0, \qquad n \in \{1, \dots, N\}, \forall k
\end{gathered} \tag{12}$$

In Equation (12), p_k represents the probability of each scenario occurring and $C(\boldsymbol{x}_k^n)$ represents the cost associated with making decision $\boldsymbol{x}$ at stage n for scenario k. Because the first non-anticipativity constraint is $\boldsymbol{x}_k^0 = \boldsymbol{x}_j^0$, the initial cost $C(\boldsymbol{x}_k^0)$ is independent of the scenario and can be rewritten as $C(\boldsymbol{x}^0)$. Writing out the second summation allows Equation (12) to be re-written as:

$$\min_{\boldsymbol{x}} \left[C(\boldsymbol{x}^0) + \sum_{k=1}^{K} p_k \left[C(\boldsymbol{x}_k^1) + \cdots + C(\boldsymbol{x}_k^N) \right] \right] \tag{13}$$

The expression $C(\boldsymbol{x}_k^1) + \cdots + C(\boldsymbol{x}_k^N)$ in Equation (13) is the total cost of every decision made for scenario k, or more simply the total cost of scenario k. This expression for the total cost of scenario k can be represented by the term C_k. Equation (13) can then be simplified to:

$$\min_{\boldsymbol{x}} \left[C(\boldsymbol{x}^0) + \sum_{k=1}^{K} p_k \, C_k \right] \tag{14}$$

After being given initial decision $\boldsymbol{x}^0$, the platform designer begins the process of computing the remaining decisions. The first payload from the first scenario, represented by ξ_1^1, is added to the platform and then the recourse decision $\boldsymbol{x}_1^1$ is computed. The process continues until each decision $\boldsymbol{x}_1^1, \dots, \boldsymbol{x}_1^N$ has been computed. After all decisions have been computed, they are input into the cost model which outputs the total cost of scenario 1, C_1. This process for computing the total cost is then repeated, one scenario at a time, for every scenario in the sample. It should be noted that by computing the decisions one scenario at a time, the second non-anticipativity constraint of (12) is automatically enforced. Now that that initial cost and total costs for all scenarios has been computed, the total expected cost, given by the expression inside the brackets in Equation (14), can be computed. Finally, MATLAB's genetic algorithm function is used to find the initial station configuration that minimizes the expected cost in an efficient manner. The optimal initial decision is denoted as $\boldsymbol{x}^{0^*}$.

3.2.5 Cost Model

The cost of on-orbit servicing is difficult to estimate and is highly dependent on the servicing infrastructure that has been put in place. Several works have explored the cost of on-orbit servicing infrastructures [1] [19]. Instead of using a specific cost model, it is an objective of this thesis to explore the effects that changes in a simple cost model have on the optimal platform design. This method will help obtain an initial estimate for what the cost of servicing likely needs to be reduced to in order to achieve the benefits of HPPs.

The cost model used in this work involves computing the cost associated with every decision that is made by the decision-maker. The decisions consist of an optimized initial decision, which chooses how many platform modules to use in the initial platform design, followed by calculated decisions for how many modules to add following the selection of each payload. The cost of the initial decision is assumed to be only the cost of launching the mass of the initial modules to the desired orbit. The cost-per-kg to GEO and LEO was found by averaging the costs for three launch vehicles: Atlas V, Delta IV, and Falcon 9 [20] [21]. These costs are shown in Table 8 below.

Table 8: Launch cost-per-kg for selected launch vehicles

	Cost per kg ($, FY2018)	
	LEO	**GTO**
Falcon 9	2,700	7,470
Atlas V	12,225	25,843
Delta IV	12,193	22,072
Avg	**9,039**	**18,462**

The cost of all subsequent decisions is computed as the cost of launching the mass of the new modules to the desired orbit plus the cost of servicing associated with adding the new modules. It is assumed that there is already a servicer on orbit that is dedicated to servicing the HPP, so the initial cost of developing and launching a servicer is not included in this model. The servicing cost for each decision is computed as the cost of launching the new modules using the prices in Table 8, plus the cost-per-kg of servicing the new modules. With this cost model, it is assumed that the cost of servicing only depends on the mass of the modules that need to be ferried to the platform. In reality, the cost will likely depend on several additional factors.

The cost model developed here is not meant to accurately portray the costs associated with an on-orbit servicing mission architecture. Instead, this cost model aims to enable testing of the stochastic optimization methods applied in this work and to obtain an initial estimate for a servicing cost at which servicing hosted payload platforms becomes cost-effective. A precise cost model will require further research and is left for future studies.

4. RESULTS AND ANALYSIS

4.1 Effect of Form Factor Selection on Module Design

To investigate the effect that form factor selection has on the HPP design, platform designs for all four form factors, PODS, PODS-Extended, ESPA, and ESPA Grande, were compared using a sample of 1000 scenarios and a mission lifetime of 15 years with new payload additions occurring every 2 years until the total number of payloads was reached. In this case, the initial decision has not yet been optimized. Instead, the initial platform design will consist of just one of each type of platform module so that modules are only added as needed, resulting in the smallest possible station size. Key platform characteristics for the GEO and LEO cases are shown in Table 9 through Table 12 below.

Table 9: HPP designs to support 7 payloads for a lifetime of 15 years at GEO

	Total Platform Mass (kg)			Total Platform Modules			Total Propellant Resupplies		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
PODS	4820	9033	6952	27	50	41.6	5	17	9.8
PODS Extended	4384	8295	6431	19	29	25.0	1	6	3.4
ESPA	4123	7435	5953	17	22	20.1	0	3	1.4
ESPA Grande	3917	7891	6067	14	18	16.8	0	2	0.2

Table 10: Number of modules by module type for platform designs at GEO

	Average Number of Modules				
	ADCS	Communications	Power	Propulsion	Propellant Resupplies
PODS	1	3	17.4	13.2	9.8
PODS Extended	1	3	8.2	5.8	3.4
ESPA	1	3	5.2	3.9	1.4
ESPA Grande	1	3	3.0	2.8	0.2

Table 11: HPP designs to support 7 payloads for a lifetime of 15 years at LEO

	Total Platform Mass (kg)			Total Platform Modules			Total Propellant Resupplies		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
PODS	772	5956	2526	15	36	21.7	0	13	3.4
PODS Extended	838	5283	2396	13	21	15.8	0	5	0.8
ESPA	722	4417	2196	12	17	14.0	0	2	0.2
ESPA Grande	999	5675	2348	12	15	12.7	0	1	0.0

Table 12: Number of modules by module type for platform designs at LEO

	Average Number of Modules				
	ADCS	Communications	Power	Propulsion	Propellant Resupplies
PODS	1	2	4.4	7.3	3.4
PODS Extended	1	2	2.3	3.5	0.8
ESPA	1	2	1.5	2.4	0.2
ESPA Grande	1	2	1.0	1.7	0.0

The total number of modules listed in Table 9 and Table 11 include all platform modules as well as all 7 payloads. Table 10 and Table 12 show the breakdown of the total modules into the specific module types, minus the payloads. One of the shortcomings of a modular design such as the HPPs considered in this work is the parasitic mass associated with the module structure and other components such as propellant tanks. Fewer modules used to make up the platform should result in a lower amount of dead weight. This effect is clearly seen in Table 9 and Table 11; for smaller form factors such as PODS and PODS-Extended, a larger number of modules are needed to support the payloads, resulting in a higher total platform mass. Additionally, it is observed that while the ESPA Grande form factor, which is the largest of the four, has the lowest average number of modules required to make up the platform, it does not have the lowest average total mass. This indicates that the platform may be “over-designed”, meaning that it has more capability than is required to support the payloads. These results indicate that the ESPA form factor may be the most efficient for the payload set chosen in this work.

4.2 Initial Platform Design Optimization

Several methods are used to test the platform designer's optimization of the initial platform design decision. First, an excessively high cost is assigned to servicer effort. This should result in an initial platform design that requires no platform module additions and no propellant resupplies for all scenarios considered. Then, no cost is assigned to servicer effort, which should result in an initial decision that allows for the maximum amount of servicer use to only add modules as needed, which would minimize the total mass of the platform. The platform designs and optimal initial decisions in this case would be very similar to the designs presented in Section 4.1, which assumed an initial decision of $\boldsymbol{x}^0 = [1,1,1,1,0]$. Once it is determined that the optimizer is working as expected, results can be obtained for a realistic cost for servicing at which HPPs would be most effective.

4.2.1 High Cost on Servicer Effort

The optimal initial decision and the resulting HPP designs for an excessively high cost on servicer effort is shown in the tables below.

Table 13: Optimal initial decision compared with final platform design for GEO HPP

	PODS		PODS Extended		ESPA		ESPA Grande	
	$\boldsymbol{x}^{0*}$	Average Total	$\boldsymbol{x}^{0*}$	Average Total	$\boldsymbol{x}^{0*}$	Average Total	$\boldsymbol{x}^{0*}$	Average Total
ADCS	1	1	1	1	1	1	1	1
Comms	3	3	3	3	4	4	3	3
Power	22	22	10	10	7	7	4	4
Propulsion	34	34	13	13	7	7	4	4
Propellant Resupplies	0	0	0	0	0	0	0	0

Table 14: Properties of optimal platform designs in GEO

	Total Number of Modules	Total Platform Mass (kg)		
		Min	Max	Avg
PODS	67	7371	9382	8466
PODS Extended	34	6045	8407	7333
ESPA	26	5293	7592	6481
ESPA Grande	19	5705	7590	6647

Table 15: Optimal initial decision compared with final platform design for LEO HPP

	PODS		PODS Extended		ESPA		ESPA Grande	
	x^{0*}	Average Total	x^{0*}	Average Total	x^{0*}	Average Total	x^{0*}	Average Total
ADCS	1	1	1	1	1	1	1	1
Comms	2	2	2	2	2	2	3	3
Power	9	9	4	4	2	2	1	1
Propulsion	26	26	10	10	5	5	3	3
Propellant Resupplies	0	0	0	0	0	0	0	0

Table 16: Properties of optimal platform designs in LEO

	Total Number of Modules	Total Platform Mass (kg)		
		Min	Max	Avg
PODS	45	3017	6238	4102
PODS Extended	24	2266	5541	3376
ESPA	17	1599	4236	2605
ESPA Grande	15	1797	3983	2750

In Table 13 and Table 15, the columns labeled x^{0*} show the optimal initial decision that was found. The "Average Total" columns show the total number of each type of module after the final stage, averaged over all scenarios in the sample. As expected, the two columns match exactly for each form factor, indicating the high cost on servicer effort forces the optimizer to choose an initial configuration that will support all payloads of every scenario for the entire lifetime without adding any modules or refueling.

This case represents a worst-case situation in terms of total platform mass because all platform modules must be added in the initial stage instead of being added as-needed, resulting in higher propellant consumption for a longer period of time and thus requiring more propulsion modules. Despite the parasitic mass from the modular design, even for this worst-case scenario, HPPs can offer significant mass savings over developing a dedicated satellite to support each payload. Table 17 shows characteristics of the total combined mass of the seven satellites used to determine the payload set for each scenario.

Table 17: Combined mass of satellites required to support 7 payloads

	Minimum Mass	**Maximum Mass**	**Average Mass**
GEO	12,796 kg	24,128 kg	18,910 kg
LEO	783 kg	12,380 kg	4,690 kg

The average mass required for a dedicated satellite platform for each payload in GEO is significantly higher than even the worst-case average for hosted payload platforms, represented by the PODS form factor in GEO, with an average mass savings of about 55%. The average mass for individual satellite platforms in LEO is also higher than worst-case LEO average, with a mass savings of about 13%.

The ESPA form factor once again appears to be the optimal choice of the four form factors that have been analyzed. From this point on, the results presented will focus exclusively on the ESPA form factor.

4.2.2 No Cost on Servicing

Now, instead of placing an arbitrarily high cost on servicing, we remove the cost on servicing altogether to represent the other end of the spectrum. With no cost on servicing, the optimizer should choose to start with the minimal allowed initial station size, so that modules can be added as needed. Comparisons between the optimal initial platform size and the average final platform size of all scenarios in the sample is shown in Table 18 and Table 19 below.

Table 18: Optimal platform designs in GEO for ESPA form factor with no cost on service effort

	x^{0*}	Average Total	Average Final Platform Mass
ADCS	1	1	5,984 kg
Comms	1	3	
Power	1	5.3	
Propulsion	1	3.9	
Propellant Resupplies	0	1.5	

Table 19: Optimal platform designs in LEO for ESPA form factor with no cost on service effort

	x^{0*}	Average Total	Average Final Platform Mass
ADCS	1	1	2,178 kg
Comms	1	2	
Power	1	1.5	
Propulsion	1	2.4	
Propellant Resupplies	0	0.2	

As expected, the optimal decision is to start with the smallest platform allowed and use the servicer as much as possible to minimize the mass of the platform. When compared with the case of a high price on servicing, this case reduces the average mass of the platform after the final stage by over 497 kg for the GEO case and 427 kg for LEO.

4.2.3 Effect of Servicing Cost on Optimal Platform Design

Servicing is likely to be considerably costly, but the cost is highly dependent on the servicing infrastructure put in place. As an example, if there are multiple servicers already in orbit, then the closest servicer would be tasked with collecting the new modules and bringing them to the platform. Developing and operating multiple servicers at a time would have a high initial cost but would be more efficient in the long run than developing and employing a single dedicated servicer for one HPP. Additionally, fuel depots in space would lower the cost of refueling but would again have a high initial cost. As on-orbit satellite servicing technologies develop, and infrastructures begin to emerge, the servicing cost will begin to shift from the extreme with a very high cost of servicing towards the other extreme with a low cost of servicing. The concept of hosted payload platforms discussed in this work will become more beneficial as the cost of servicing decreases. Now that both extremes have been analyzed and the optimizer has been shown to be working properly, it is desired to examine how changes in the cost model affect the optimal decision for the initial platform design and at what costs of servicing are significant increases in servicer use observed. Figure 2 through Figure 5 below show various platform characteristics as the price of servicing changes for the GEO case using the ESPA form factor.

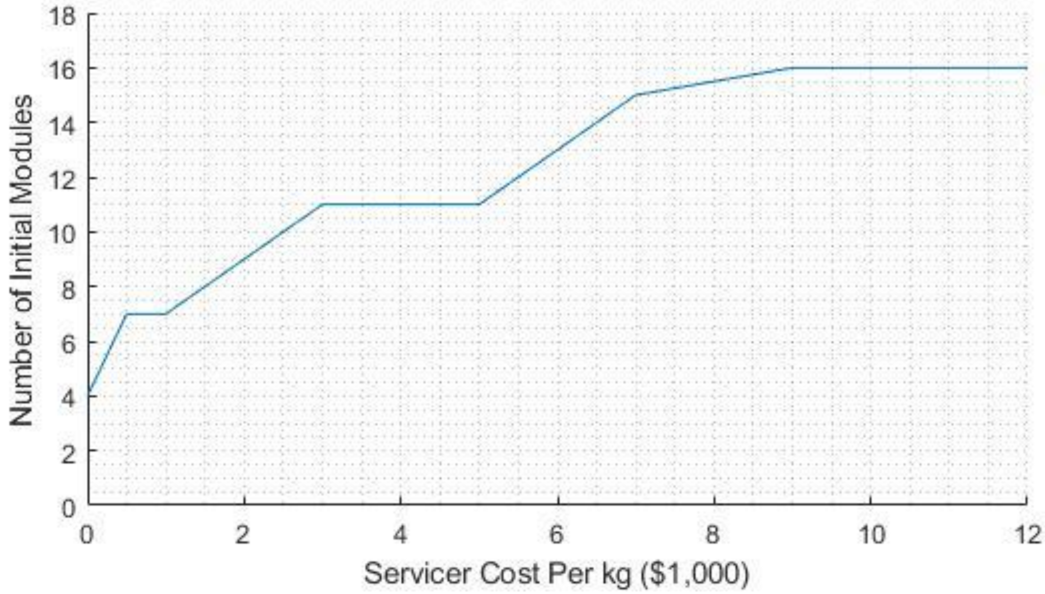

Figure 2: Size of initial platform compared to cost of servicing in GEO

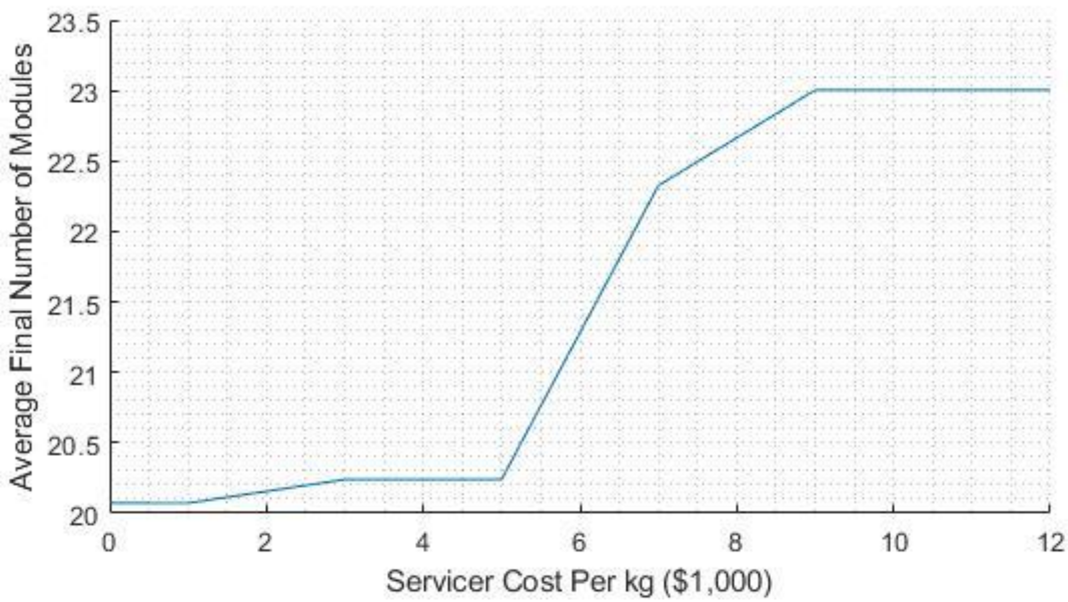

Figure 3: Average size of final platform compared to cost of servicing in GEO

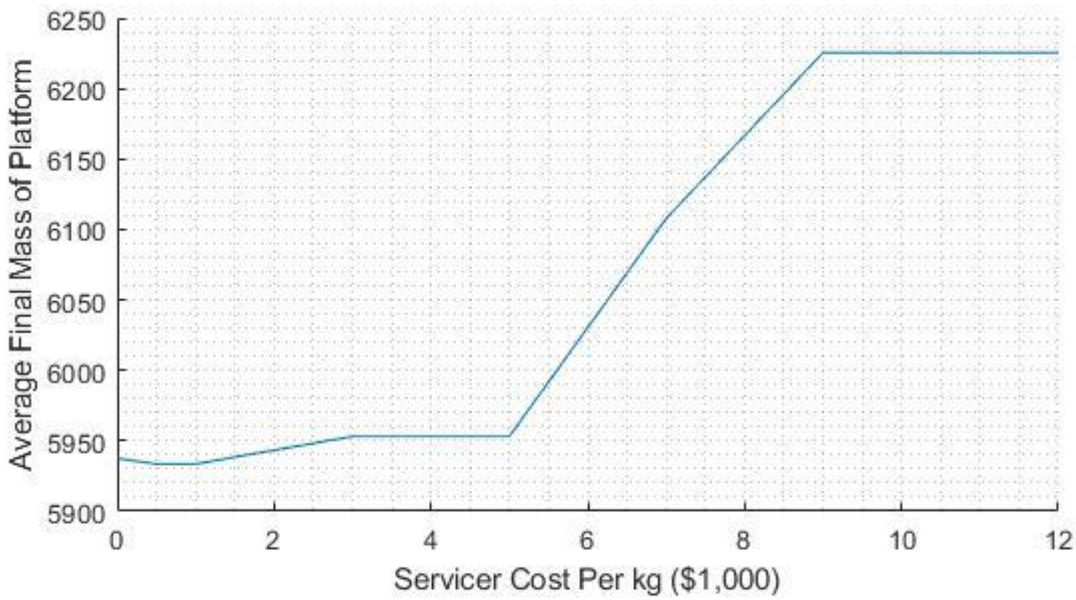

Figure 4: Average final mass of platform compared to cost of servicing in GEO

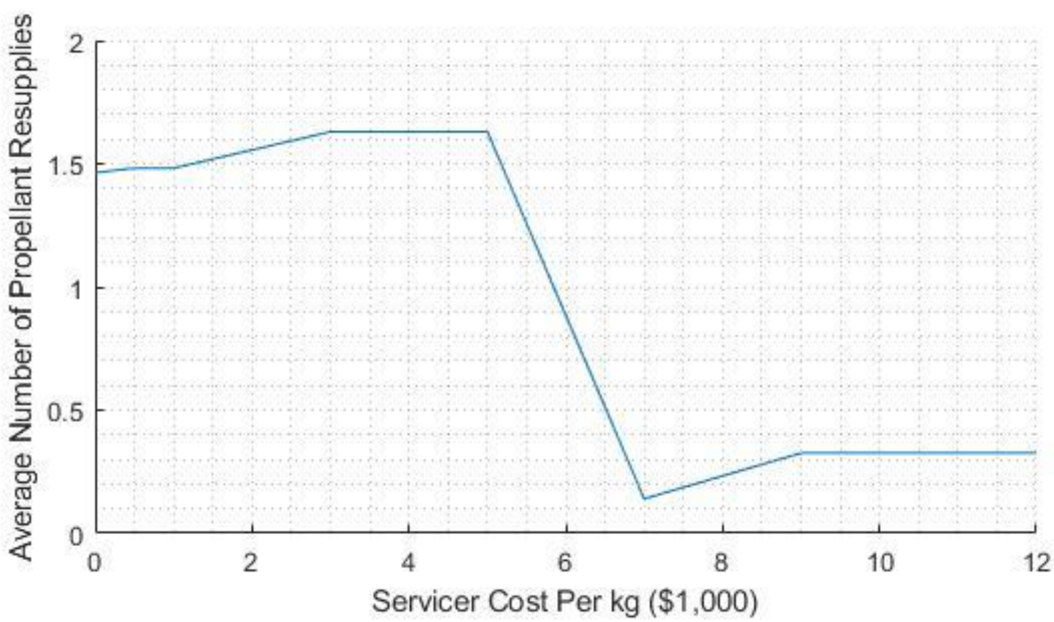

Figure 5: Average number of propellant resupplies compared to cost of servicing in GEO

In the figures above, it is observed that the optimal initial platform design remains unchanged at costs exceeding roughly $9,000 per kg. Above this cost, the optimizer chooses not to utilize the servicer at all. Once the cost is reduced below that point, the optimizer begins to select fewer modules for the initial platform, indicating that it is more optimal to utilize a servicer than to add initial modules which are not immediately required. The number of initial modules continues to decrease until the cost reaches the edge case of $0 per kg, at which point the minimum number of required modules, one of each type of platform module for a total of four, is reached. Since the minimum platform size only reaches a minimum when there is no servicing cost, which is not a realistic scenario, it appears that the optimal solution at low costs is not always starting with the minimum platform size. Rather, it is a combination of servicer utilization and adding some extra capability initially.

Similarly, the average of all scenarios for the total number of modules after the final stage decreases in the same cost interval, although the difference in number of modules is not as drastic as with the initial platform size. As expected, as the total number of modules decreases, so does the total platform mass. Additionally, as the cost of servicing is reduced, the number of propellant resupplies begins to increase. Based on this simplified cost model and the scenario sample used to obtain the data, on-orbit servicers begin to become effective when the cost of bringing 1 kg of mass from its launch orbit to the platform drops below about $9,000. It should be noted that this cost is dependent on the scenario sample that was selected. In order to obtain a more reliable result, a sensitivity analysis is needed. Similar plots for the LEO case are shown in Figure 6 through Figure 9 below.

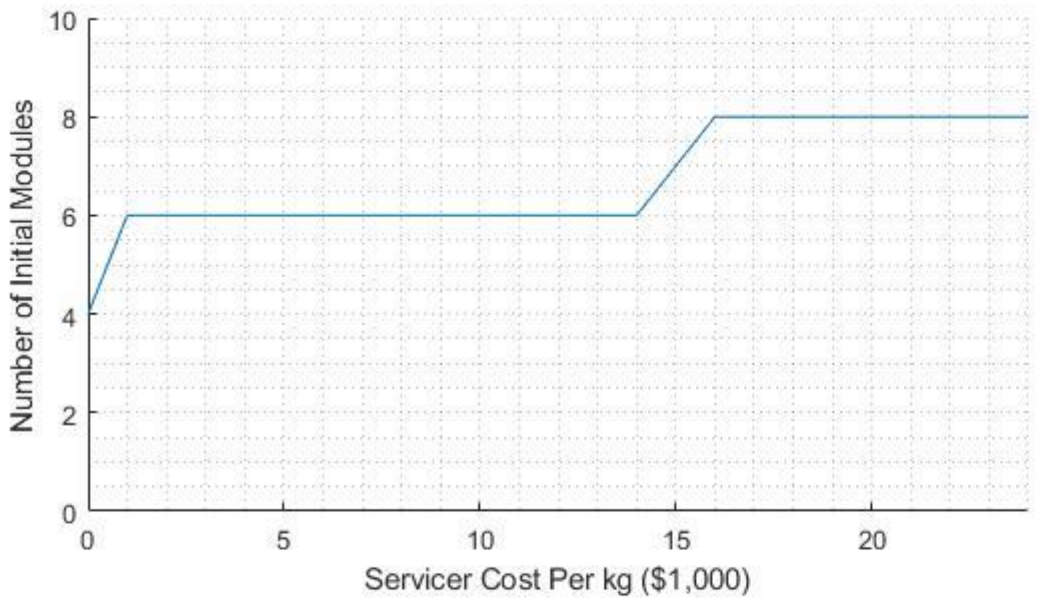

Figure 6: Size of initial platform compared to cost of servicing in LEO

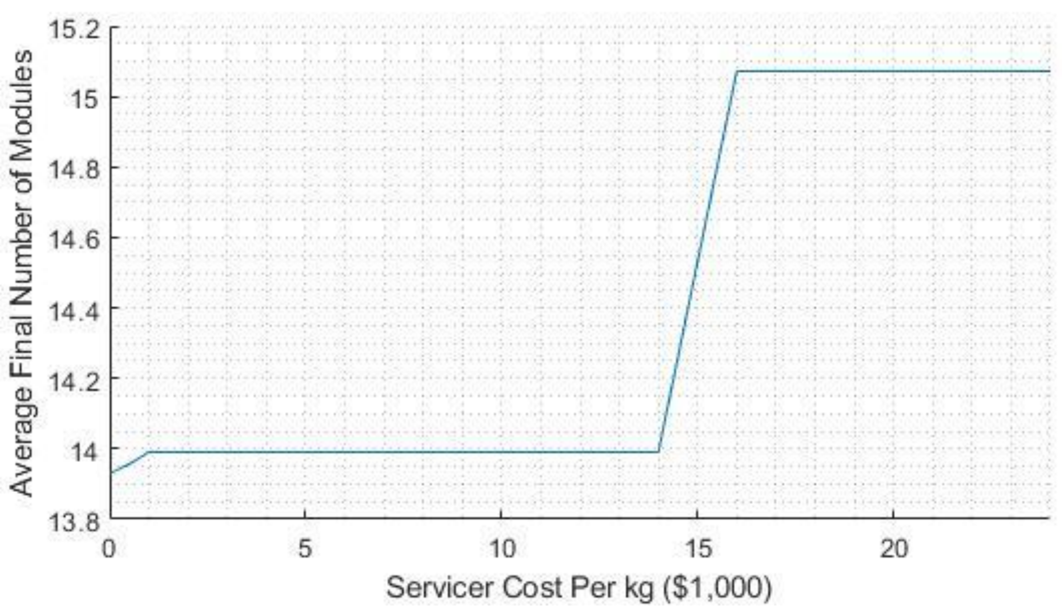

Figure 7: Average size of platform compared to cost of servicing in LEO

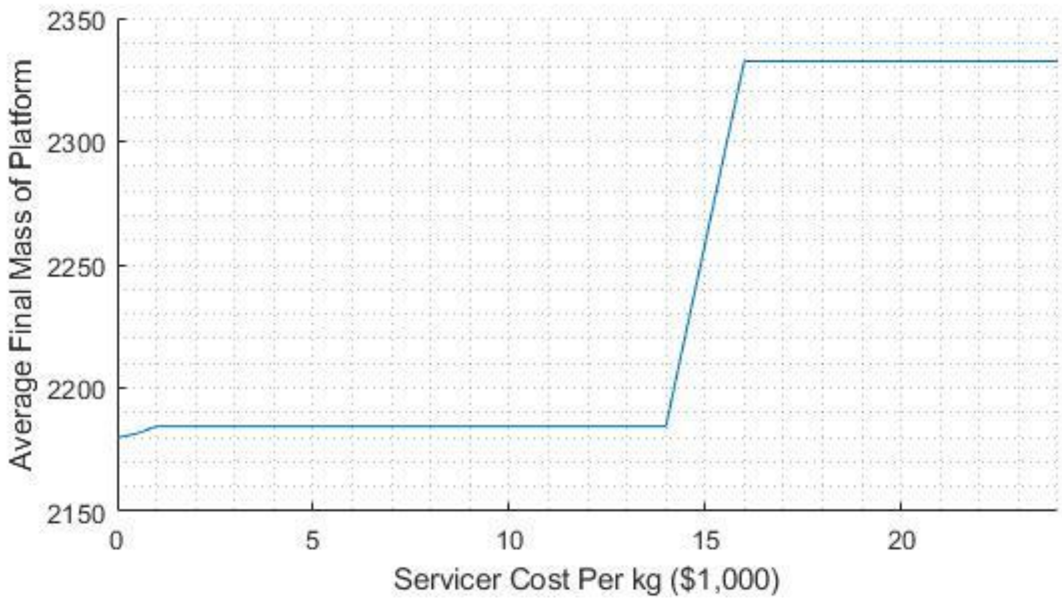

Figure 8: Average final mass of platform compared to cost of servicing in LEO

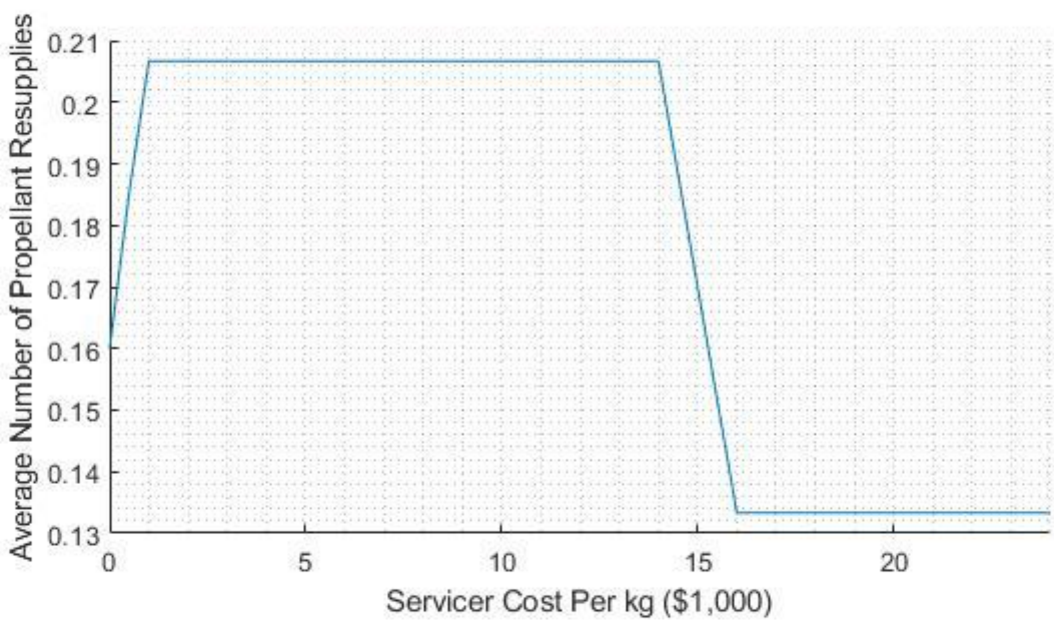

Figure 9: Average number of propellant refills compared to cost of servicing in LEO

Similar trends to those of the GEO case are observed for LEO. In this case, however, the optimizer begins to utilize servicing at a cost of about $16,000. As with the GEO case, the initial platform size only reaches the minimum at the edge case with no servicing cost. With a fully developed, more precise servicing cost model, the methods used here would be helpful for determining the level of infrastructure necessary for HPPs to be cost effective and could help motivate technological advancements to begin to put the infrastructure into place.

4.3 Sensitivity Analysis

With such a large number of possible scenarios, samples of the scenario set must be taken to lower the computational costs to manageable levels. For small sample sizes, there is likely to be larger variation in platform design results because the optimizer has fewer scenarios to consider when searching for an optimal solution. As the sample size increases, the sample provides better estimates for the actual scenario set. With the optimizer considering a larger number of possible scenarios, the variance in the platform design should start to decrease. A sensitivity analysis was performed for the GEO case study by running the optimizer for 100 different samples using a fixed cost of $6,000 and a fixed sample size. This was done for sample sizes of 10, 50, and 100 scenarios. The results of this analysis are shown below.

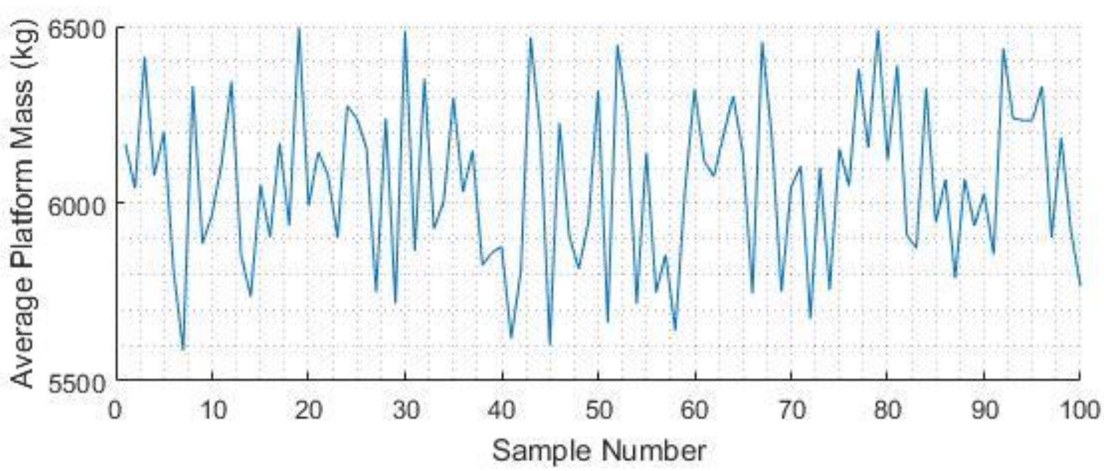

Figure 10: Variations in average platform mass – 10 scenarios per sample

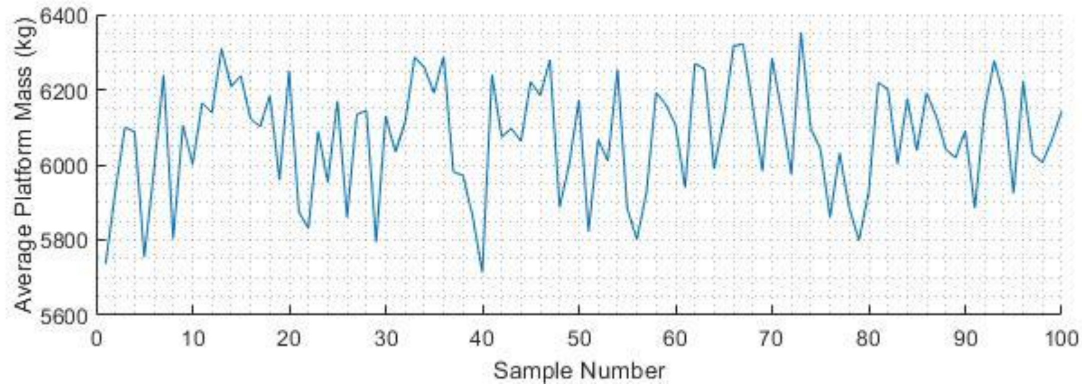

Figure 11: Variations in average platform mass – 50 scenarios per sample

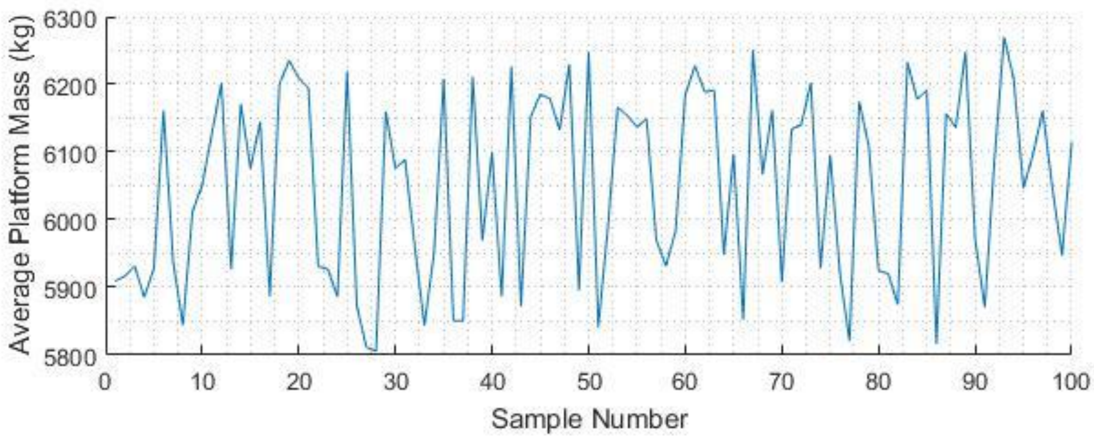

Figure 12: Variations in average platform mass – 100 scenarios per sample

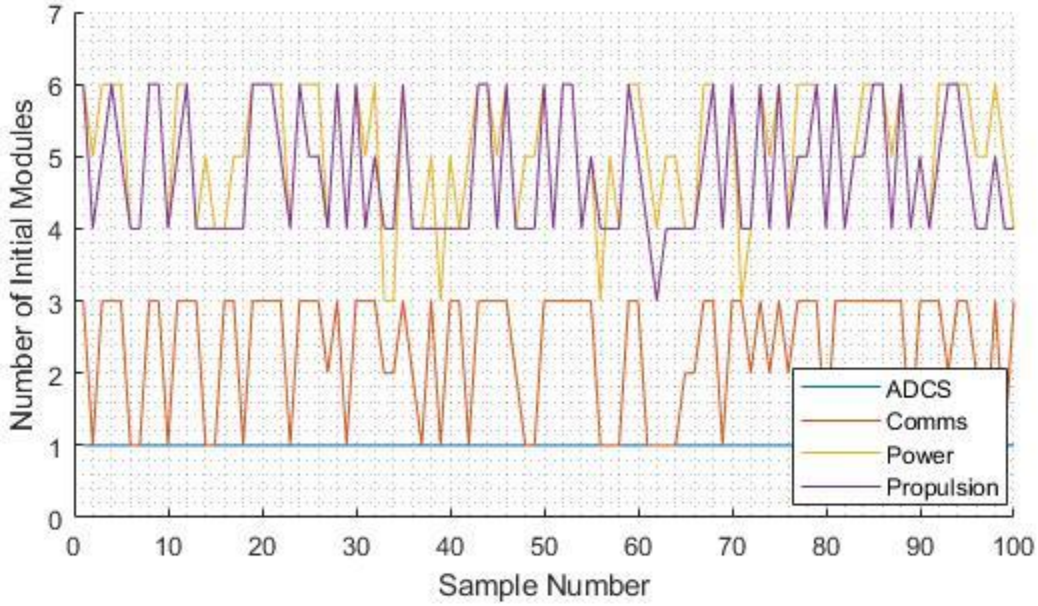

Figure 13: Variations in initial platform size - 10 scenarios per sample

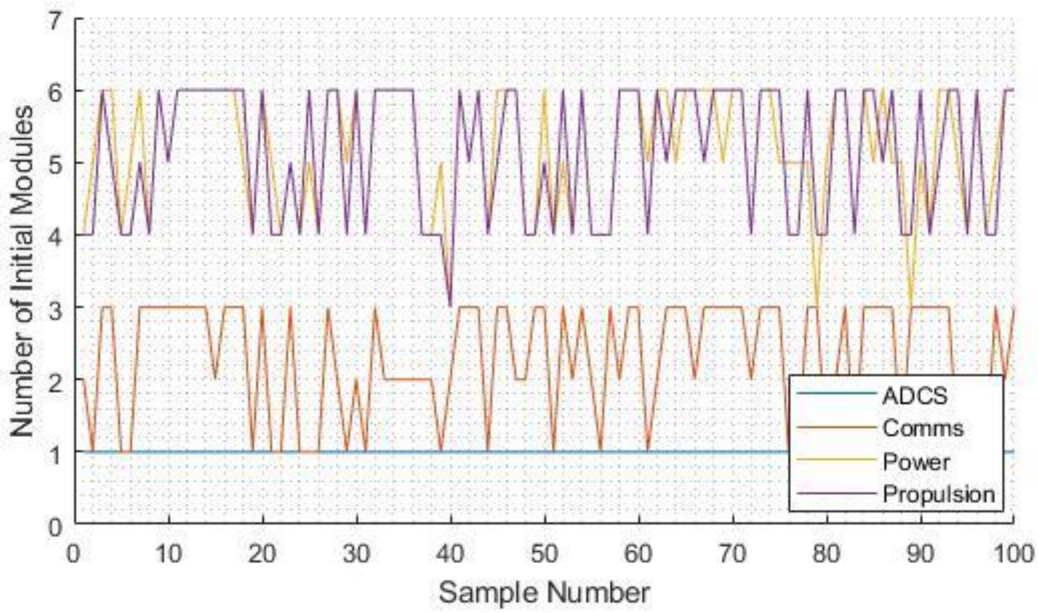

Figure 14: Variations in initial platform size - 50 scenarios per sample

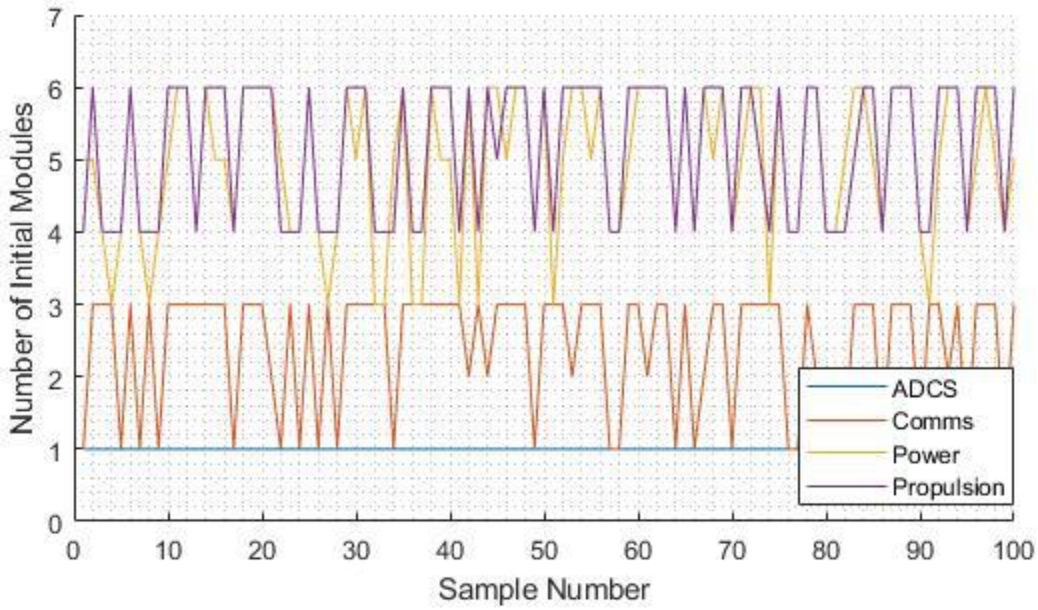

Figure 15: Variations in initial platform size - 100 scenarios per sample

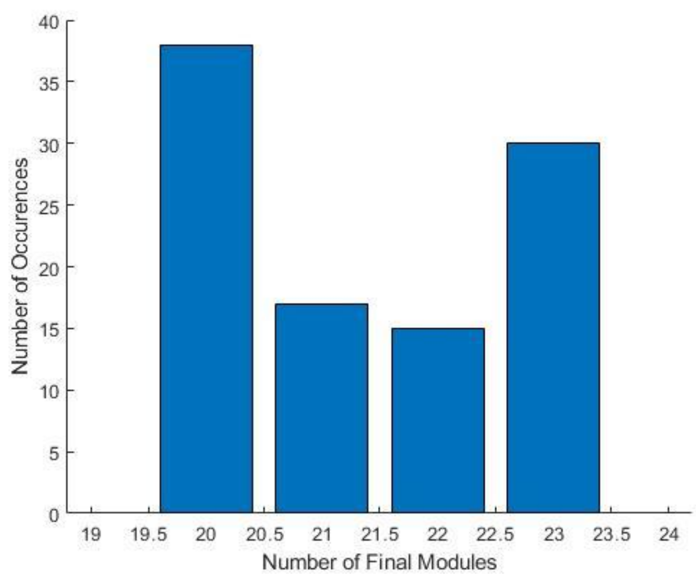

Figure 16: Variations in final platform size - 10 scenarios per sample

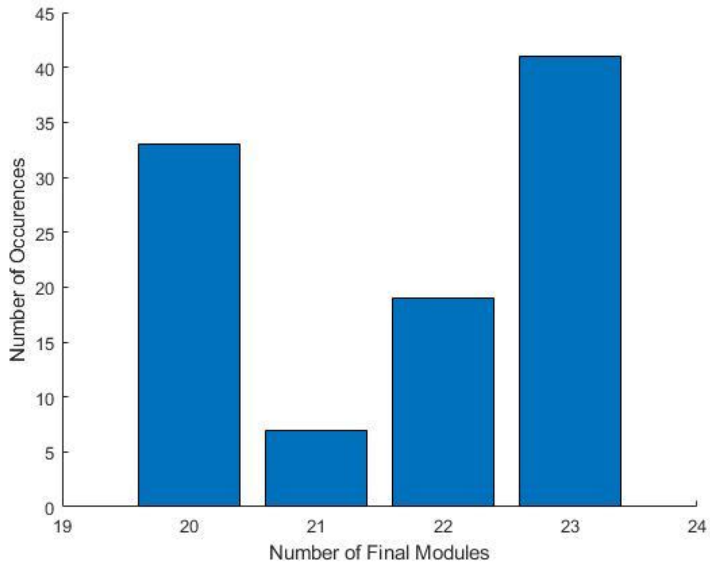

Figure 17: Variations in final platform size - 50 scenarios per sample

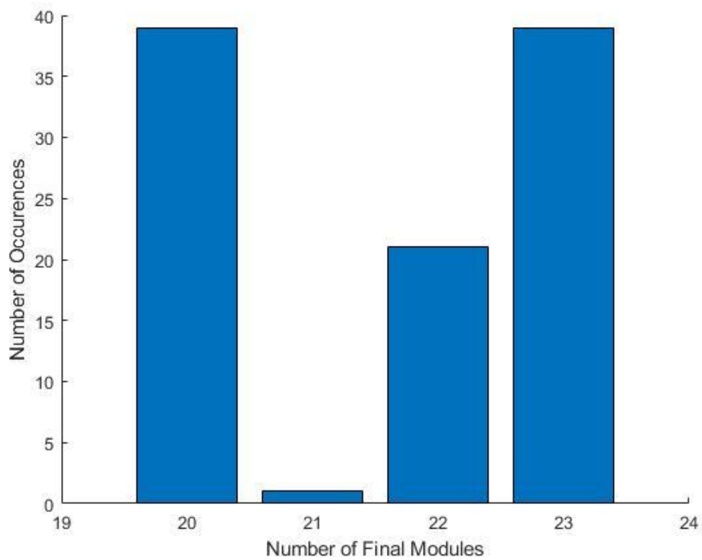

Figure 18: Variations in final platform size - 100 scenarios per sample

Figure 10 through Figure 12 show the variations in the average final platform mass for sample sizes of 10, 50, and 100 scenarios, respectively. It is observed that as the sample sizes increase, the variations in the average final platform mass start to decrease and fall within a smaller range. For the case with only 10 scenarios per sample, the masses fall within a 900 kg range.

Increasing the sample size to 100 scenarios per sample gives final masses that all fall within roughly 450 kg of each other. Figure 13 through Figure 15 show the variations in the number of each type of platform module in the optimal initial platform size. All three of these figures look very similar and do not immediately offer any insight into the trend as sample sizes increase. In each case, the number of power and propulsion modules stays mostly bounded between 4 and 6 modules, with a few occurrences of 3 modules. The number of communications modules stays bounded between 1 and 3, and the number of ADCS modules stays constant at 1 module. Figure 16 through Figure 18 show the average final size of the platform designs. In all cases, the total platform size stays bounded between 20 and 23 modules. A trend is observed in which the occurrences of final platform sizes of 20 and 23 modules increase as the sample size increases, while occurrences of final platform sizes of 21 and 22 modules. This could be explained by high probabilities on certain payloads which may have a larger impact on platform design than other payloads. As the number of scenarios per sample increases, the scenarios containing these high probability payloads also increase, causing the platform design to tend towards certain sizes as indicated in the figures. With the high computational cost on larger sample sizes, research into computational methods to make these methods more efficient is likely required to further characterize the quality of the solutions obtained.

5. CONCLUSIONS

This thesis proposed an approach using stochastic programming methods to solve the design challenge presented by uncertain payload selection for hosted payload platforms. In this work, an HPP architecture was formulated within an on-orbit servicing framework, probability distributions were created to characterize the uncertainty of payload selection and to generate scenarios, and a simple cost model was created to assess the effect of servicing cost on optimal platform designs. The platform design results showed the benefits of HPPs over dedicated satellites and established an estimated cost of servicing at which it is optimal to utilize servicing over adding extra initial capability before launch. A sensitivity analysis using Monte Carlo simulations showed a general trend towards the convergence of final platform sizes as the sample size increased, however no convergence trend was yet observed for the optimal initial platform size. Future research into computational methods to reduce the cost of simulations with large sample sizes is likely required to further assess the quality of solutions obtained using these methods.

The results presented here are an initial step towards addressing the complicated design problem presented by hosted payload platforms. It is this author's hope that the methods used in this thesis can be further developed so that a more precise estimate can be made as to how far on-orbit servicing development needs to go until HPPs become an efficient, cost effective solution. There are several logical next steps to build off the methods and results presented in this paper. One proposed next step is to replace the simple cost model used in this thesis with a well-developed servicing infrastructure cost model. This could provide much more realistic estimates for the cost at which the optimizer begins to utilize on-orbit servicing, which in turn would help determine whether the concept of hosted payload platforms is viable in the current market or, if not, how much more progress needs to be made towards developing OOS infrastructures in order to gain the full benefits of the concept. Another area of future work is to incorporate additional areas of uncertainty into the stochastic programming model. Logistical uncertainties, such as launch vehicle delays or servicer delays, may have an impact on the margin required for platform module additions and could affect the optimal platform design decisions. Uncertainties related to component or module failures could also impact the platform design. An additional area of improvement could be to add more design variables to the optimizer, such as launch vehicle and form factor selection, to further refine the optimal design.

6. REFERENCES

[1] "Developing On-Orbit Servicing Concepts, Technology Options, and Roadmap," International Space University, Beijing, 2007.

[2] J. Parrish, "Robotic Servicing of Geosynchronous Satellites (RSGS)," Defence Advanced Research Projects Agency, [Online]. Available: https://www.darpa.mil/program/robotic-servicing-of-geosynchronous-satellites. [Accessed 12 March 2019].

[3] Northrop Grumman Corporation, "Mission Extension Vehicle - A rendezvous Proximity Operation and Docking Bus for SpaceLogistics," 2018. [Online]. Available: https://www.northropgrumman.com/Capabilities/SpaceLogistics/Documents/MEV_Factsheet.pdf. [Accessed 12 March 2019].

[4] Canadian Space Agency, "Canadarm2, the Canadian robotic arm on the Space Station," 12 June 2018. [Online]. Available: http://www.asc-csa.gc.ca/eng/iss/canadarm2/default.asp. [Accessed 19 March 2019].

[5] Canadian Space Agency, "Dextre, Canada's Space Handyman," 19 June 2018. [Online]. Available: http://www.asc-csa.gc.ca/eng/iss/dextre/default.asp. [Accessed 19 March 2019].

[6] European Space Agency, "European Robotic Arm," 17 April 2014. [Online]. Available: http://www.esa.int/Our_Activities/Human_and_Robotic_Exploration/International_Space_Station/European_Robotic_Arm. [Accessed 19 March 2019].

[7] S. Ridinger, "NASA's Dragonfly Project Demonstrates Robotic Satellite Assembly Critical to Future Space Infrastructure Development," NASA, 13 September 2017. [Online]. Available: https://www.nasa.gov/mission_pages/tdm/irma/nasas-dragonfly-project-

demonstrates-robotic-satellite-assembly-critical-to-future-space.html. [Accessed 15 March 2019].

[8] B. Lal, et al. "Global Trends in Small Satellites," Institute for Defense Analyses, Alexandria, 2017.

[9] General Atomics, "General Atomics Awarded Air Force Space and Missile Systems Center Hosted Payload Solutions Satellite Delivery Order," General Atomics, 03 January 2019. [Online]. Available: http://www.ga.com/general-atomics-awarded-air-force-space-and-missile-systems-center-hosted-payload-solutions-satellite-delivery-order. [Accessed 12 March 2019].

[10] M. Andraschko, R. Baize, S. Horan, D. Neil, P. Rinsland and R. Zaiceva, "The Potential for Hosted Payloads at NASA," in *IEEE Aerospace Conference*, 2012.

[11] Space Systems Loral, "Payload Orbital Delivery System (PODS)," [Online]. Available: http://sslmda.com/html/access_to_space.php. [Accessed 20 March 2019].

[12] Moog CSA Engineering, "ESPA Overview," [Online]. Available: https://www.csaengineering.com/products-services/espa.html. [Accessed 20 March 2019].

[13] A. Shapiro and A. Philpott, "A Tutorial on Stochastic Programming," Georgia Institute of Technology, Atlanta, 2007.

[14] B. Defourny, D. Ernst and L. Wehenkel, "Multistage Stochastic Programming: A Scenario Tree Based Approach to Planning under Uncertainty," University of Liege, Liege, Belgium , 2010.

[15] S. Mitra, "A White Paper on Scenario Generation for Stochastic Programming," 2006.

[16] Union of Concerned Scientists, "UCS Satellite Database," 9 January 2019. [Online]. Available: https://www.ucsusa.org/nuclear-weapons/space-weapons/satellite-database#.XFcLhlxKjcs. [Accessed 19 March 2019].

[17] NASA, "State of the Art of Small Spacecraft Technology," NASA, 12 March 2019. [Online]. Available: https://sst-soa.arc.nasa.gov/02-integrated-spacecraft-platforms. [Accessed 22 March 2019].

[18] J. R. Wertz, D. F. Everett and J. J. Puschell, Space Mission Engineering: The New SMAD, Hawthorne: Microcosm Press, 2011.

[19] NASA Goddard Space Flight Center, "On-Orbit Satellite Servicing Study," NASA, Greenbelt, MD, 2010.

[20] Bryce Space and Technology, "The Annual Compendium of Commercial Space Transportation: 2018," Federal Aviation Administration, 2018.

[21] Government Accountability Office, "Surplus Missile Motors - Sale Price Drives Potential Effects on DOD and Commercial Launch Providers," United States Government Accountability Office, 2017.