3-2005

An Approach for Optimizing the On-Orbit Servicing Architecture for the Space-Based Radar Constellation

Michael L. McConnell

Recommended Citation

McConnell, Michael L., "An Approach for Optimizing the On-Orbit Servicing Architecture for the Space-Based Radar Constellation" (2005). *Theses and Dissertations*. 3763.
https://scholar.afit.edu/etd/3763

AN APPROACH FOR OPTIMIZING THE ON-ORBIT SERVICING ARCHITECTURE FOR A GIVEN CLIENT SATELLITE CONSTELLATION

THESIS

Michael L. McConnell, Captain, USAF

AFIT/GLM/ENS/05-17

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GLM/ENS/05-17

AN APPROACH FOR OPTIMIZING THE ON-ORBIT SERVICING ARCHITECTURE FOR A GIVEN CLIENT SATELLITE CONSTELLATION

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Logistics Management

Michael L. McConnell, BS

Captain, USAF

March 2005

AFIT/GLM/ENS/05-17

AN APPROACH FOR OPTIMIZING THE ON-ORBIT SERVICING ARCHITECTURE FOR A GIVEN CLIENT CONSTELLATION

Michael L. McConnell, BS

Captain, USAF

Approved:

__//signed//________________________________ __21 March 2005
STANLEY E. GRIFFIS, Lt Col, USAF (Co-Chairman) Date

__//signed//________________________________ __21 March 2005
VICTOR D. WILEY, Maj, USAF (Co-Chairman) Date

AFIT/GLM/ENS/05-17

Abstract

Satellite systems, once operational, are essentially a consumable item with no capacity to maintain, repair, or upgrade them while on-orbit. In order to avoid having to replace costly space assets, the Defense Advanced Research Projects Agency (DARPA) and Air Force Space Command (AFSPC) are looking to developing programs to provide an on-orbit servicing capability for future satellite systems under development, such as the Space-Based Radar (SBR) system. DARPA and AFSPC are studying on-orbit servicing using the Orbital Express platform as part of an Analysis of Alternatives for the SBR program. Like their satellite clients, on-orbit servicing assets are expected to be resource intensive, and so proper management of these space logistics assets is essential.

This research provides a flexible planning tool to determine the optimal on-orbit servicing architecture for a given client satellite constellation and applies it to the proposed SBR constellation. The model uses a generalized network structure with side constraints to efficiently solve this large combinatorial optimization problem. The optimal number and type of servicing vehicles to use is found, along with the associated most efficient routing to meet client satellite demand for two commodities within multiple time windows.

Acknowledgments

I would like to express my sincere appreciation to all those without whose support this work could not have been possible, especially my committee members. I would also like to thank my sponsors for the support and latitude provided to me in this endeavor. Finally, I would like to acknowledge the hard work and determination of my wife who, in pursuing her own Master’s at the same time, somehow maintained the strength to put up with me throughout my research efforts as well.

Michael L. McConnell

Table of Contents

List of Figures

List of Tables

AN APPROACH FOR OPTIMIZING THE ON-ORBIT SERVICING ARCHITECTURE FOR A GIVEN CLIENT SATELLITE CONSTELLATION

I. Introduction

1.1 Background

The Air Force invests a great deal of resources in acquiring, launching, and operating satellite systems for a wide variety of tasks, from weather forecasting and communications, to guiding bombs onto targets. Every branch of the military depends on satellites to operate effectively across the globe. However, satellite systems are treated as essentially a consumable item. There is currently no ability to maintain, repair, or upgrade satellites while in orbit, and if a satellite fails, it must be replaced or the capability that satellite brought to the fight is lost.

Space is an extremely harsh environment in which to operate. Satellites in orbit around the Earth are subject to rapid temperature swings when moving in and out of the shadow of the Earth throughout an orbit, as well as the full spectrum of electromagnetic radiation from the Sun. X-rays, Gamma rays, Extreme-Ultraviolet radiation, and radio bursts are all hazards commonly encountered by the sensitive electronics onboard satellites (Air University, 2003). These hazards can cause individual components or even whole satellites to fail.

Researchers are looking at on-orbit servicing as an alternative to satellite replacement. On-orbit servicing can include anything from upgrade, repair, or cleaning solar panels to assembly of very large spacecraft (Waltz, 1993). Neither the technology nor the management policies are mature enough yet to make on-orbit servicing a reality at this time, though current development efforts are focused on making servicing a technological possibility by as early as 2010 (Tirpak, 2002). When on-orbit servicing becomes a realistic option, satellite systems will become much more flexible and responsive in their capabilities.

To help drive the development of technologies necessary for on-orbit servicing, Air Force Space Command (AFSPC) is including on-orbit servicing as one of its logistics alternatives in an Analysis of Alternatives (AoA) for "Operationally Responsive Spacelift" (McCormick, Hardy & Sundberg, 2003). One space system currently under development that is considering servicing in its design is Space Based Radar (SBR). SBR is planned to be an operationally responsive augmentation of the ground moving target indicator (GMTI) capability currently provided by the airborne platform, the E-8 Joint STARS. By basing a GMTI radar in space, it is hoped that battlefield commanders and intelligence personnel will have access to look far beyond their current reach, and make plans long before the already heavily-tasked E-8's can be brought into the area (Tirpak, 2002). A team led by Northrop-Grumman is developing the system, with the option of design for servicing being examined.

SBR has been resurrected from the Discoverer II program (Tirpak, 2002). The Discoverer II program was meant to be a technology demonstration platform. Conflict between the military services about the required capabilities and planned interfaces with war fighters, along with the perception that an operational platform was too far off in the future, caused Congress to cancel the program in 2000 (Tirpak, 2002). When the Bush administration made the Air Force the executive agency for military space, more coordinated planning for the SBR program developed. This, together with advancements in technology and more focused research and development efforts, made SBR a viable program, focusing on producing an operational system by 2010, not just another technology demonstrator (Tirpak, 2002).

Though the technology to make servicing a reality may be in place within the next decade, the policies of how to employ such technology need to be in place before fielding the new capability. Any on-orbit servicing system will be expensive, and the resources devoted to it should be used as efficiently as possible.

1.2 Problem Statement

Many studies have been conducted to determine the cost feasibility of on orbit servicing (Divinic, Chappie, Arkus & Greenberg, 1997; Hardy, 2003; Lamassoure & Hastings, 2001; Perez, Pires & Singleton, 2002). These studies examined the cost of servicing a satellite in terms of its commercial, civil, and military utility, direct servicing costs, and even the value of increased capability or flexibility. Most previous studies made assumptions regarding the servicing architecture already in place, and client constellations examined have been small, only 1 or 2 client satellites. Few studies have examined the different servicing architectures available. There is a need to examine the management alternatives for costly on-orbit servicing resources by looking at what architecture(s) would most efficiently utilize servicing assets. This research attempts to determine the optimum servicing architecture for the space-based radar (SBR) constellation as a way to integrate SBR as an operationally responsive space platform.

1.3 Investigative Questions

In order to determine what the optimum servicing architecture for SBR is, the following investigative questions must be answered.

1. What will be the demand for on-orbit servicing from the SBR constellation? This question is answered by answering these sub-questions:

    1-A. How many client satellites are in the SBR constellation?

    1-B. In what orbits are the client satellites operating?

    1-C. What is the nature of servicing demanded: maintenance, upgrade, etc.?

    1-D. How much servicing does each client require?

    1-E. Are there time restrictions on when servicing must occur?

2. Can the servicing system meet the demand of the SBR constellation?

    2-A. What is the nature of the servicing system?

    2-B. In what orbits can the servicing system operate?

    2-C. How many servicing system components (depot spacecraft and servicing vehicles) are available?

    2-D. What is the capacity of each servicing system component?

    2-E. What is the maneuvering capability of each servicing system component?

3. What servicing assets are available and how are they best utilized to satisfy SBR demands?

    3-A. Can the amount of the servicing system utilized be altered?

    3-B. Is there more than one feasible employment strategy for the servicing system?

3-C. Which employment strategy utilizes the minimum number of servicing system components?

3-D. What is the minimum number of servicing system components that meets SBR demand?

1.4 Scope

This research uses SBR as a client constellation to determine the best feasible servicing architecture. Examination of the dollar costs involved is not directly studied, though by optimizing servicing assets utilized it is assumed that budgetary resources are also optimized. Though SBR is used for this research, the methodology allows the substitution of any satellite constellation given data such as orbits used, mass demanded, etc. The research is not meant to provide the final, best answer for what servicing assets to employ, but rather a starting solution for analysts and planners to work from when adding real world complexity.

1.5 Document Overview

Chapter 2 reviews the published literature related to on-orbit servicing. It looks at the history of and the studies examining on-orbit servicing, as well as the status of on orbit servicing systems. Chapter 2 also briefly examines different research methodologies used in similar past studies and thus applicable to this research, and suggests the use of an optimization approach using a generalized network structure to solve the on-orbit servicing problem. Chapter 3 details the formulation of the on-orbit servicing problem in a manner similar to a facility location and vehicle routing problem with time windows, and Chapter 4 details the implementation of the model. Chapter 5 summarizes the results and conclusions and suggests areas for further research.

II. Literature Review

This chapter reviews and summarizes previous work related to on-orbit servicing and the problem of developing an optimal autonomous servicing architecture. The first section will briefly review examples of selected historical missions in which on-orbit servicing was a key objective. Section 2.2 lists some of the economic value of an operational servicing capability, as well as identifying past studies that have examined the economic feasibility of on-orbit servicing. This section continues by describing some of the non-financial benefits to servicing, highlighted by the possible applications to national security space systems and the Space Transportation System. Section 2.3 reviews the status of on-orbit servicing programs with a focus on Boeing's Orbital Express program, the servicing system alternative used as a baseline in this research. Section 2.4 examines areas in which further research is needed, including the requirement for a flexible but robust model for determining optimal servicing architectures. The chapter concludes with a characterization of the SBR on-orbit servicing problem and a discussion on methodologies applicable to solving it. The methods used by other researchers to solve similar problems are briefly noted, and use of a suggested model for solving the on-orbit servicing problem is presented.

2.1 On-Orbit Servicing Missions in History

There have been several manned missions to space where servicing was a major objective. This section briefly discusses a few high-profile missions.

The Solar Maximum Mission

In February of 1980, NASA launched the Solar Maximum Mission spacecraft to collect observations of solar flares, sunspots, magnetic fields, and the energy output of the sun (Waltz, 1993). After 10 months of operation, electronics malfunctions forced a decision to either replace the satellite or repair it. NASA chose to use the Space Shuttle to repair the Solar Maximum Mission thereby demonstrating that on-orbit servicing (in this case repair) was feasible (Waltz, 1993).

The servicing mission to the Solar Max spacecraft also showed that spacecraft could be upgradeable, as long as designs incorporated the ability to be serviced. The mission also showed that servicing of spacecraft on orbit could significantly increase satellite life expectancy.

Space Station Mir/Progress Missions

The Soviet, and later Russian, space station program began in 1977 with *Salyut 6* and served as a demonstration that operations necessary for servicing spacecraft in orbit were achievable. By docking with manned Soyuz and robotic *Progress* spacecraft, the Soviet space station demonstrated the feasibility of routine autonomous docking, refuel, and re-supply (Lamassoure & Hastings, 2001). While suffering only three first-dock attempt failures (only one docking had to be accomplished manually), more than 40 *Progress M* re-supply spacecraft autonomously delivered supplies to and returned waste from the *Mir* space station, the second generation of Russian manned orbiting facilities. Even with minor glitches, the stations proved that autonomous rendezvous, docking, and refueling operations were achievable (Lamassoure & Hastings, 2001).

The Space Shuttle Missions to the Hubble Space Telescope and International Space Station

The two most extensively serviced spacecraft to date are the Hubble Space Telescope and the International Space Station. When the Hubble Space Telescope first became operational, the fuzzy images returned showed evidence of a major malfunction. Because Hubble had been designed as a serviceable spacecraft, NASA deemed it cost effective to launch a Space Shuttle mission to repair the telescope's main reflector (Lamassoure & Hastings, 2001). Since that first repair mission, three other missions to Hubble have been accomplished, not just to repair mirrors and gyros, but also to upgrade electronic components and sensors. The second servicing mission to Hubble extended the telescopes sensing range from just visible light into near infrared wavelengths. The third and fourth servicing missions added further capability to Hubble, allowing it to see into ultraviolet wavelengths, and also replacing degraded solar arrays (NASA, 2004a). NASA estimated that each new instrument placed on Hubble increased its "scientific power by a factor of 10 or greater (NASA, 2004a)." Hubble served as a highly visible example of the capability increases possible in a spacecraft designed for regular servicing.

The International Space Station takes the on-orbit rendezvous and docking accomplishments of *Salyut* and *Mir* a step farther, relying on assembly of major components on-orbit for success. The station weighs over 400,000 pounds, is 146 feet long, 240 feet wide, and 90 feet tall (NASA, 2004b), clearly much too large to launch into orbit in one piece given today's launch capability. The major components of the station were assembled in several separate missions from 1998 to 2002. Over its lifetime, the International Space Station has seen 41 manned and robotic flights to assemble and maintain the spacecraft.

Although history has shown there is a value to servicing spacecraft on-orbit, fully autonomous servicing has yet to be proven feasible and economically viable, especially for spacecraft less expensive than Hubble or the ISS. The next section discusses research accomplished to answer the questions of "Can autonomous servicing be feasibly and economically accomplished?"

2.2 Previous Studies on the Feasibility and Value of On-Orbit Servicing

Advanced Robotics Enabling Autonomous Servicing Capabilities

In 2003, a group from the German Aerospace Center (DLR) studied the near- and long-term capabilities of tele-robotic operations on spacecraft. DLR designed the advanced tele-robotic arm, Canadarm 2, currently used on the International Space Station (Hirzinger, Landzettel, Brunner, Fischer, Preusche, Reintsema, Albu-Schäffer, Schreiber & Steinmetz, 2003), and is also experimenting with fully autonomous robotic designs for tasks such as routine inspection and cleaning on exterior surfaces of the Space Station. DLR is designing advanced lightweight robotic arms for use on vehicles like the Spacecraft Life Extension System (SLES), a servicing vehicle that will be discussed later in this chapter (Hirzinger et al., 2003).

Advanced robotics designs like those coming out of DLR, along with the historical evidence and planned demonstrators (see Section 2.3) are key steps being taken towards a mature autonomous on-orbit servicing capability. In addition to the technological obstacles to overcome, the economic value of autonomous servicing must also be examined.

The Economic Feasibility of Autonomous On-Orbit Servicing

Many studies have examined the trade-off between the cost “savings” of total spacecraft replacement, and the cost of building and employing a servicing capability in support of satellite missions. Perez, Pires, and Singleton’s (2002) study used a conceptual client satellite constellation of two satellites in low Earth orbit (LEO), with one spare satellite ready for launch as a replacement. They found that scheduled servicing every 3.5 years reduced design reliability requirements and could lead to potential cost savings of $37.4 million. While this at first appears to be a large cost savings, the cost of an operational servicing architecture was not calculated, but is expected to be significantly more than $37.4 million. The authors of this study did conclude that scheduled servicing alone would not be worth it, but additional revenue from unscheduled repair missions and increased client satellite capability could push savings over $125 million (Perez et al., 2002).

The value of servicing to a client satellite constellation depends heavily on the design of the client spacecraft, specifically, how much of the spacecraft is capable of being serviced. The Spacecraft Modular Architecture Design (SMARD) study (Divinic et al., 1997) categorized different levels of servicing in terms of “serviceability level.” The SMARD study found that servicing could be up to 38% less expensive than satellite replacement when at least 30% of the client satellite is designed for servicing. One key assumption of this architecture was that failed components were left on the client satellite and new components “bolted on”(Lamassoure & Hastings, 2001).

Leisman and Wallen (1999) also studied on orbit servicing, focusing on possible architectures for upgrading the Global Positioning System (GPS) constellation. Their study examined 30 different servicing architectures. Their designs choices were based on mass delivered by the servicing vehicle, the number of clients serviced, design life, and propulsion type (chemical, electrical, and solar thermal), along with variable time/space strategies for maintenance (Leisman & Wallen, 1999). On the basis of cost, Leisman and Wallen (1999) concluded there were six best alternatives, all including one servicing vehicle for each of the client constellation's orbital planes performing four servicing missions over 15 years. Using the 1996 NASA/Air Force Cost Model, the researchers found a total cost of $300 million, which, though higher than the baseline upgrade plan (time-phased satellite replacement) for GPS, had a higher value by reducing the time to repair or upgrade the constellation. The "best alternatives" also were an order of magnitude cheaper than another option of lump replacement of the entire constellation (Leisman & Wallen, 1999). This study was a very thorough examination of different architectures based on cost and value. However, it was limited by technological assumptions made by the researchers, and could only evaluate the alternatives the researchers chose. Leisman and Wallen's (1999) study was also a very specific answer for a very specific constellation, and it would be difficult and time-consuming to replicate for other possible client constellations. In addition to financial costs and benefits to servicing, the non-financial values of servicing must be considered.

Non-Financial Benefits of On-orbit Servicing

Beyond the potential cost savings in satellite replacement, autonomous on-orbit servicing has several non-financial benefits as well. Primary among these is the potential to reduce or eliminate risk to personnel. Current servicing missions must be completed by the Space Shuttle, which is costly and puts human lives at risk. On-orbit servicing can also allow certain missions (reconnaissance and intelligence satellites especially) the flexibility to maneuver throughout their life times, thus improving their capabilities in terms of responsiveness and the ability to spend increased time over specific areas of interest. A joint team from the National Reconnaissance Office (NRO), Air Force Space Command (AFSPC), and other government agencies examined the utility of servicing to national security satellites. The resultant list of benefits are discussed in the following sections.

Reduced Reliance on the Space Shuttle

Currently all on-orbit servicing missions to satellites must be completed by humans riding aboard the Space Shuttle. Although this allows virtually unlimited flexibility in repairs affected, it limits the number of satellites that can be serviced to those reachable by the Shuttle. It also makes servicing missions very expensive and puts personnel at risk of loss-of-life if a Shuttle is lost.

The Space Shuttle showed its utility in on-orbit servicing early in its life with the servicing of the Solar Maximum Mission satellite. Other missions between 1982 and 1986 showed further flexibility when shuttle astronauts retrieved two communication satellites that failed to reach proper orbit and repaired another (Columbia Accident Investigation Board (CAIB), 2003). The Space Shuttle also has great value in its capability to lift very heavy missions, 36,000 to 54,000 lbs (17,000 to 25,000 Kg), to orbit. However, the Shuttle can only reach altitudes of 115 to 690 miles (185 to 1,104 Km) and inclinations of 28.5° or 51.6° (Boeing, 2004). This means that satellites in higher inclinations (more polar orbits) or higher altitude (mid-Earth or geostationary) orbits cannot be reached. Shuttle missions also are very expensive endeavors as opposed to unmanned missions using expendable launch vehicles. The launch cost of the Atlas V and Delta IV series rockets ranges from $75 million to $160 million (Isakowitz, Hopkins & Hopkins, 2004). The Government Accounting Office (GAO)[1] calculated the average cost of a Space Shuttle mission to be $380 million (GAO, 2001). Because launch costs represent a large proportion of most satellite missions, it is economically feasible to service only the most expensive satellites, like Hubble, instead of simply replacing them after a failure.

In addition to the dollar-cost of a servicing mission, the Shuttle also puts human lives at risk. The tragic destruction of both the Space Shuttles *Challenger* and *Columbia* highlight the possible dangers involved in sending astronauts into space. Although some have come to see a trip on the Shuttle as a routine adventure, in the words of the Columbia Accident Investigation Board (2003), "Building and launching rockets is still a very dangerous business, and will continue to be so for the foreseeable future while we gain experience at it (CAIB, 2003)." Although a human can perform a wider variety of servicing tasks than an autonomous servicing vehicle, the benefit of not risking a human life should be considered in satellite mission planning and design. An autonomous servicing capability need not be looked at as a replacement for the Shuttle however.

---

[1] The GAO has changed its name from Government Accounting Office to Government Accountability Office. This work uses the name of the agency at the time of the referenced publication.

The *Columbia* accident, as terrible as it was, also raised a potential application of autonomous on-orbit servicing. Findings of the Columbia Accident Investigation, conducted in 2003 following the disaster, included areas where inspection and repair of the orbiter should be a goal for NASA. Some of these findings are listed here:

> - For missions to the International Space Station, develop a practicable capability to inspect and effect emergency repairs to the widest possible range of damage to the Thermal Protection System…
> - For non-Station missions, develop a comprehensive autonomous (independent of Station) inspection and repair capability…
> - Accomplish on-orbit Thermal Protection System inspection…early in all missions
> - The ultimate objective should be a fully autonomous capability for all missions to address the possibility that an International Space Station mission fails to achieve the correct orbit, fails to dock successfully, or is damaged during or after undocking
>
> (CAIB, 2003: p. 225)

While these recommendations use the word "autonomous" to mean independent of the ISS or NASA ground controllers, a fully robotic servicing vehicle could accomplish many of these tasks in addition to servicing satellites. Another potential application could be to deliver tailored spares kits directly to the Shuttle so that astronauts can repair damage suffered by the orbiter. While reducing the risk to personnel in space is a great non-financial benefit, it is by no means the only one.

Value of Flexibility

By looking at satellites as the commercial assets they are, their capabilities can be valued beyond just the platform cost. According to Lamassoure and Hastings (2001), the traditional methods of evaluating the value of satellites underestimate the influence that uncertainty plays. They believe that much of the value of a mission lies in flexibility brought on by servicing. By being able to repair and refuel a satellite, many of the "uncertain" events that can occur in a satellite mission (failure to achieve orbit, failure to pass checkout, failure to deploy solar arrays, etc.) may be mitigated by the possibility of repair or refueling. This would turn what is "unknown" into a calculable "risk" with a known cost and benefit (Lamassoure & Hastings, 2001). Some of the value of flexible missions also comes from the ability to re-task a satellite to a different orbit or inclination, a capability that is expensive today because of limited fuel reserves on satellites. A major user of satellite systems, the U.S. government, also studied this capability.

Joint Study on the Utility of On-Orbit Servicing

In early 2003, a team made up of members from the NRO, Headquarters AFSPC, the Defense Advanced Research Projects Agency (DARPA), and the Space and Missile Center (SMC), completed a joint study quantifying the potential utility of autonomous and tele-robotic on-orbit servicing for national security spacecraft. The study found that on-orbit servicing is a "significant potential enabler of national security space capabilities" for missions in low-Earth orbit (LEO) and geostationary orbit (GEO) (McCormick et al., 2003). Specifically, the study found that on-orbit servicing has potential utility for pre-planned product improvement, fueling, and assembly of very large spacecraft. The fueling capability is especially desirable in that larger satellites could be launched "dry" and fully fueled upon reaching orbit, or as "maneuver insurance" improving satellite capability (McCormick et al., 2003). However, before radical changes to satellite design and mission planning occur to incorporate servicing, an autonomous servicing capability must be demonstrated. Autonomous on-orbit servicing is still in its infancy, but advancements are being made in both the civilian and military arenas. The next section of this chapter discusses the status of autonomous on-orbit servicing.

2.3 Status of Autonomous On-Orbit Servicing

The dream of fully robotic satellite servicing is still several years away from becoming a reality. Despite the restrictions of our current technological capabilities, scientists in both civilian and military organizations are pushing the boundaries of what is possible. This section describes the advancements already made and projects planned in the near future.

Commercial Advancements

Several privately funded and government-assisted on-orbit servicing projects are underway in the commercial sector, both in the U.S. and abroad. Two companies working on such projects are Orbital Recovery and AeroAstro. Orbital Recovery is a European aerospace company involved in “satellite insurance services and risk management” which bills itself as “*the* European on orbit servicing development program (Orbital Recovery Corporation, 2004).” Working primarily with Dutch Space, Orbital Recovery has developed the ConeXpress Orbital Life Extension Vehicle (CX-OLEV) as a “space tug” to provide station-keeping services to communications satellites that have run out of propellant. The CX-OELV has made extensive use of current technology, modifying the current Ariane-5 payload adapter to accomplish its servicing tasks. The first planned flight is in 2007. Orbital Recovery also sees the CX-OELV spacecraft as a rescue vehicle for satellites that have become stranded in improper orbits (Orbital Recovery Corporation, 2004).

A Northern Virginia-based company, AeroAstro, is also taking on the challenge of on-orbit servicing with their Small Payload Orbit Transport (SPORT) and Escort Satellite projects. The SPORT project is a module that can be attached to small LEO satellites. The SPORT module uses aero-braking to transfer satellites from GEO orbits down to LEO orbits, allowing the small LEO satellites to be launched as a secondary payload on GEO launch missions (AeroAstro, 2004). The Escort satellite project is being developed as an add-on service to high-investment spacecraft missions. The Escort will fly "in formation" with a client satellite and provide visual and thermal imaging. Escort will also provide electronic and communications monitoring to quickly diagnose failures and allow preventive/corrective actions (AeroAstro, 2004). This capability reduces the required complexity of self-contained fault detection systems for already costly missions.

In addition to commercial ventures, the U.S. Government has also decided to invest in pursuing an on-orbit servicing capability.

Government/Military Advancements

DARPA is working to develop a fully robotic capability for on-orbit servicing, and hopes to significantly improve the capability of commercial and U.S. national security space programs. DARPA is working with a team led by Boeing to develop Orbital Express. The goal of Orbital Express is to "validate the technical feasibility of robotic, autonomous on-orbit refueling and reconfiguration of satellites… (DARPA, 2004)." Orbital Express' planned capabilities include refueling and component replacement in order to increase client satellite life, increase maneuverability, and increase launch margins. Component replacement will allow repair of failed components as well as upgrades to mitigate the risk of satellites becoming technologically obsolete (Potter, 2003). An operational system for Orbital Express is envisioned for 2010 with a demonstration system to be launched in 2006 on the Air Force Space Test Program mission (DARPA, 2004). The Orbital Express system is the most extensive servicing system currently under development, and is the system being considered in the AFSPC AoA relating to the SBR program.

Orbital Express Architecture

The Orbital Express operational architecture will consist of three main components: a servicing vehicle, an on-orbit spares and fuel depot, and a next generation client satellite designed for servicing. The servicing vehicle, Autonomous Space Transfer and Robotic Orbiter (ASTRO), will be able to completely autonomously rendezvous, dock with client satellites, and perform either fluid transfer, orbit replaceable unit (ORU) transfer, or both (Potter, 2003). The ASTRO vehicle will also be able to make repeated trips to and from the Commodities Spacecraft (CSC). The CSC will be an on-orbit "warehouse" spacecraft that will provide storage space for ORUs and cryogenic propellant. Some proposed architectures include the launch of "smart CSC's" that will launch directly to client satellites and serve as both CSC and ASTRO (Potter, 2003). The final facet of the Orbital Express architecture is the client satellite itself, termed NextSat. These "next generation" satellites will have to be designed for servicing using standard, non-proprietary interfaces so that ASTRO vehicles can service clients from multiple designers (Potter, 2003). For the purposes of this thesis, it is assumed that client satellites interested in servicing have been designed using these standard interfaces.

The demonstration flight in 2006 will feature one vehicle simulating both a CSC and a NextSat, and one ASTRO vehicle that will perform propellant transfer and will transfer an ORU containing a battery (Potter, 2003).

2.4 Questions Not Yet Answered by Current Research

Previous studies in on-orbit servicing have primarily followed an enumerative process. This is a labor-intensive process that not only takes time, but also limits the possible outcomes to be in terms of alternatives that the researcher must come up with on his or her own. Further, the number of servicing vehicles for a constellation has been static, not allowing the launch of different numbers and sizes of servicing vehicles at different times throughout a client satellites lifetime.

The criteria of life cycle cost and mission effectiveness have, so far, been the only response variables that mission planners have looked at (Divinic et al., 1997; McCormick et al., 2003; Perez et al., 2002; Potter, 2003). An additional criterion may be to minimize the number of launches required. Launches represent a significant variable cost to any multiple-vehicle space program, and reducing this cost would represent a significant cost savings. Another shortcoming of past studies has been the lack of optimal servicing path determination. What order satellites should be serviced in has not been examined given different servicing architectures. All of these variables should be taken into account, and a more flexible model to determine an optimal servicing architecture needs to be developed in order to respond to the anticipated future demand for on-orbit servicing.

2.5 Characterization of the Research Problem

Notional Description of the SBR Constellation

As currently envisioned, the SBR constellation will consist of 18 satellites distributed throughout 6 orbital planes (Hoy, 2004). Because the system is still under development, the exact pattern of the constellation has not yet been determined, and for the purposes of this research, the clients are assumed to be evenly spaced with a mean anomaly difference of 120 degrees. Figures 2.1 and 2.2 are generalized illustrations of the orbital planes and client satellite locations within each plane.

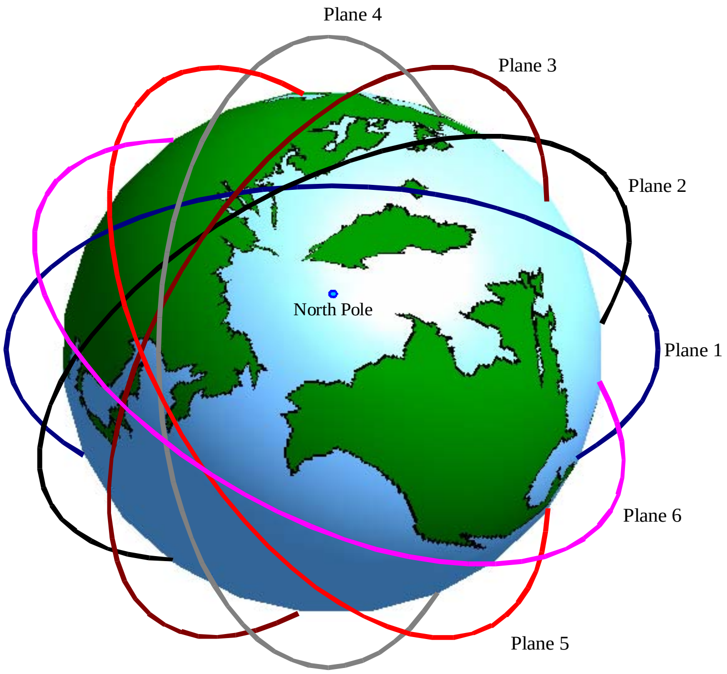


Figure 2.1 SBR orbital planes as seen from above the North Pole

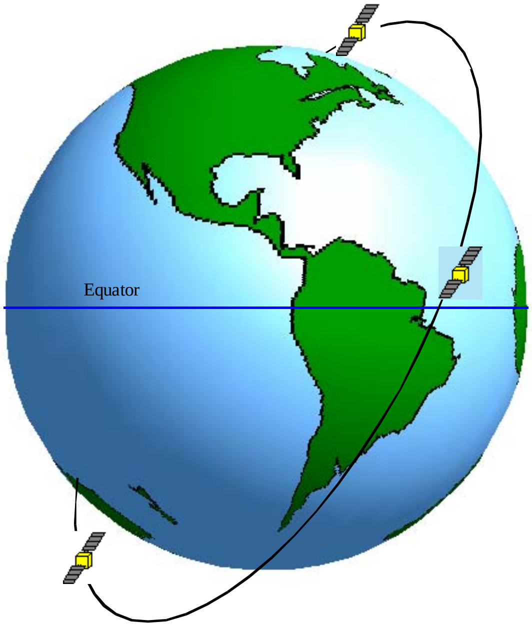


Figure 2.2 Notional relative locations of SBR satellites within orbital planes

In order to service the SBR satellites, servicing vehicles must travel to each of the client satellites. Servicing vehicles can travel from any client satellite to any other, as well as any of several depot spacecraft.

Given the nature of the servicing process, the design of an optimal architecture can be looked at as determining an optimal path on a network. In such a network, client satellites can be considered as nodes that must all be visited (serviced), within a specific time interval, else a satellite failure may occur. Servicing vehicle launch and disposal options can also be included as nodes, with connecting arcs representing the actual transfer of the servicing vehicle from one orbit/client satellite to another with costs equivalent to the amount of fuel and time needed to complete the maneuver. Each client satellite may have a different demand for servicing and the time required to move from one node to another will also vary with mass, fuel, and previous orbit. Given the types of variables, nodes, and restrictions involved, a complex vehicle routing problem appears to be a representative parallel to the SBR on-orbit servicing problem.

Vehicle Routing Problems

The classic vehicle routing problem (VRP) is a combinatorial optimization problem that minimizes the cost of routing a fleet of capacitated vehicles to a set of customers to meet specific demands (Crino, 2002; Papadimitriou & Steiglitz, 1998). This generalized VRP can be modified to more closely model real world problems by including constraints such as time windows, multiple depots, non-homogeneous vehicles, and variable vehicle capacities. The SBR on-orbit servicing problem has non-homogeneous vehicles and time windows, and has the added complexities of multiple depots and multiple servicing vehicle locations.

2.6 Methods Available for Solving Vehicle Routing Problems

Because the on-orbit servicing problem is relatively new to researchers, looking at past studies in areas outside of on-orbit servicing can be helpful in finding a methodology with which to solve the on-orbit servicing problem.

Simulation

Simulation attempts to imitate real world problems using computer models. Problems are described as *systems* of different *entities* (Law & Kelton, 2000). Simulation allows the user to examine how different alternatives perform given random occurrences, just like in the real world. For example, a manufacturing system can be simulated by modeling each process involved and the variable times it takes for each process to be completed. Although simulation is one method for evaluating the performance of a particular servicing architecture, it does not provide an optimal solution. There may be better alternatives in an on-orbit servicing architecture that are not readily apparent to the researcher. A pure simulation study places the burden upon the user to choose which alternative will be the "best" one to evaluate (Ballou, 1989). Without knowing what the best alternative in an on-orbit servicing architecture is, methods other than simulation will have to be considered.

Meta-heuristics

A technique to solve large combinatorial optimization problems that is growing in popularity is the use of meta-heuristics (Chiang & Russell, 1995; Crino, 2002; Dorigo & DiCaro, 1999; Gambardella, Taillard & Agazzi, 1999; Michalewicz, 1996; Tamashiro, Nakamura, Tamaki & Onaga, 2002). A heuristic is a method that uses specific algorithms to give a "good enough" solution. These algorithms cannot guarantee an optimal solution, but when combined with more than one solution technique, resulting in a meta-heuristic, significantly decrease the computational time necessary to find solutions. The relatively small improvement from these "good enough" solutions to optimal solutions often does not warrant the additional time and cost involved in solving real-world problems to optimality (Rayward-Smith, Osman, Reeves & Smith, 1996).

There are many meta-heuristic methods in use today, among them simulated annealing, genetic algorithms, ant colony optimization, and tabu search are well known (Rayward-Smith et al., 1996). This section briefly describes some commonly used meta-heuristic techniques.

Simulated Annealing

Simulated annealing models the behavior of individual molecules during the annealing process of an alloy during manufacturing (Rayward-Smith et al., 1996). In the annealing process, metal is heated to a specific temperature and then cooled according to a specific schedule in order to develop desired physical properties at the end. Researchers formulate problems according to a “cooling schedule” and different “temperatures” and run the algorithm to find solutions. A probability function in the simulated annealing algorithm allows moves away from local optima in search of a global optimum.

Genetic Algorithms

Genetic algorithms model the concept of natural selection to find the best solution to a problem from a “pool” of possible solutions. Selection occurs by “breeding” new solutions from the “pool” of possible solutions based on the perceived fitness of the current solutions (Michalewicz, 1996).

Ant Colony Optimization

Ant colony optimization (ACO) is a relatively new meta-heuristic technique that models the foraging behavior of insects. ACO uses artificial “ants” to lay down simulated pheromone trails along possible solutions (Dorigo & DiCaro, 1999). The ant then returns to the “nest,” and other ants follow the trail(s) left by others, laying down their own pheromone trails. The “pheromone” dissipates over time, so shorter trails (superior solutions) are favored over longer trails (inferior solutions). This heuristic technique also includes a probability that an “ant” will follow a previously unexplored path or an inferior one, allowing for escape from local optima. Dorigo and DiCaro (1999) provide a more in-depth explanation of ACO.

Tabu Search

The tabu search heuristics capitalizes on “adaptive memory” to escape local optima. By creating a list of solutions that are off-limits or “tabu” for a certain amount of time, the computer is forced to explore other solutions that may initially be inferior, but may eventually lead to a global optimum (Glover & Laguna, 1997).

Exact Techniques and Integer Linear Programming

The most obvious way to solve many problems is to look at all the options possible and choose the best one, assuming you have the necessary data and that you can fully examine all possible options. While this approach works for problems as simple as which item to order from a restaurant menu, it becomes increasingly difficult as the number of options increases. Even relatively small numbers of variables greatly increase the complexity of a problem, and thus the computation time necessary to reach a solution. Complete enumeration of all options is not a viable option for problems of any significant size.

Another option is integer linear programming. Linear programs use linear functions to represent the feasible region and value of solutions. Efficient algorithms exist for finding optimal solutions to these types of problems. Integer linear programming further restricts the feasible region to solutions comprised of integers. In general, efficient algorithms do not exist for these types of problems. Despite this, integer linear programs can be solved through optimization by structuring the problem efficiently and limiting the number of variables involved. The advantage to using linear programming is that it will find the optimal solution, if feasible, to the problem (given the constraints and assumptions made).

2.7 Methodologies Previously Applied to Vehicle Routing Problems

An initial study to determine the optimum architecture for an on-orbit servicing system was conducted by The Boeing Company (2002). Boeing's researchers examined the possible servicing architectures available for two small intelligence, surveillance, and reconnaissance (ISR) satellites constellations. Boeing used complete enumeration to determine which architecture would be most efficient based on total mass delivered per total number of launches. As stated earlier, Leisman and Wallen (1999) also used an enumerative approach when evaluating different servicing architectures for the GPS constellation.

A large-scale linear optimization program was used by Baker, Morton, Rosenthal and Williams (2002) to optimize intercontinental airlift for cargo and passengers using a constrained, constant fleet of non-homogeneous vehicles.

McCarthy (1999) attempted to determine the optimum number of KC-130J tanker aircraft the U.S. Marine Corps should purchase using ARENA and Crystal Ball simulation software. The study looked at the impact of capacity failures (failures of servicing aircraft) on waiting times of client aircraft. The study also examined the trade-off between life cycle cost and fleet size (McCarthy, 1999). However, McCarthy (1999) did not fully enumerate the alternatives, thus the solutions found cannot be guaranteed to be optimal.

Schiffman (1993) performed a study using optimization to solve a servicing network problem. The study developed an aircraft carrier battle group refueling/re-supply model for the U.S. Navy. Schiffman (1993) developed a mixed optimization/heuristic model that was based on a modified traveling salesman problem (a specific instance of a VRP). He focused on optimization within a battle group itself, with only a set number (one or two) of servicing ships (Schiffman, 1993).

Michalewicz (1996) details that genetic algorithms can be used for many optimization problems, even vehicle routing problems with time windows (VRPTWs), if the proper modifications are made to algorithm coding and problem formulation.

Gambardella, Taillard & Agazzi (1999) developed an ant colony optimization meta-heuristic to solve traveling salesman and vehicle routing problems. Although they report that it is competitive with other methods for solving VRPTWs, ant colony optimization has not yet been applied to a multiple depot instance of the problem in the open literature.

Tabu search is a popular method for solving vehicle routing and scheduling problems. Osman and Said (1996) used tabu search to solve a vehicle fleet mix problem. The vehicle fleet mix (VFM) problem is similar to a vehicle routing problem, with the added complexity of a variable number of heterogeneous vehicles. The VFM is computationally harder than a vehicle routing problem, so exhaustive search techniques are impossible for problems of large size (Osman and Said, 1996: 132). Osman and Said (1996) found that tabu search obtains impressive results when solving large combinatorial optimization problems. They reviewed the published works related to solving the vehicle fleet mix problem, and reported on three different algorithms used, an interactive route perturbation procedure (PERT), a modified PERT, and a tabu search, in terms of computer usage times and best solution given to the vehicle fleet mix problem.

Osman and Said (1996) found the tabu search yielded 15 “best known solutions” to 20 test problems, and 10 of those solutions were new results (Osman & Said, 1996). Further, their tabu search produced the smallest average relative percentage deviation over the best-known solutions.

Glover and Laguna (1997) also examined tabu search applications to vehicle routing problems. Their work showed that tabu search experienced no difficulties when solving routing problems with time window constraints and customer sets of up to 100 customers. Another large-scale combinatorial optimization problem was solved by Cullenbine (2000). Cullenbine used tabu search in a nuclear weapons assignment problem with over 16 goals, 10,000 constraints, and 500,000 decision variables. The reported tabu search significantly reduced computation time over other methods and did not appear to be limited by problem size or formulation.

The bounds of tabu search applicability were further expanded by Crino (2002). Crino (2002) applied the mathematical group theory concept within the tabu search meta-heuristic to an advanced vehicle routing and scheduling problem. Crino’s problem involved assigning various logistics assets to optimize supply going to forward-deployed Army units. Crino reported, “Tabu search applications have provided the best solutions in the least amount of time for many instances of the vehicle routing and scheduling problem” (Crino, 2002:21). This modified tabu search methodology provided the first means to solve vehicle routing problems with multiple vehicle trips, multiple vehicle types, and a variable number of hubs over an extended period.

Wiley (2001) used group theoretic tabu search and object-oriented programming to both establish a high quality method for and to produce a "suite of excellent solutions to any instance of" the aerial refueling tanker assignment and routing (Wiley, 2001).

Although a powerful tool, tabu search involves intense computer programming that is often very time consuming. Analysts often have neither the time nor the computer programming skill necessary to create a new tabu search program for each real-world problem to which they must find a solution.

Table 2.1 summarizes some of the techniques applied to different VRPs in the past.

Table 2.1 Methods Used to Solve Vehicle Routing and Similar Problems

Methodology	Authors
Enumeration	(Boeing, 2002; Leisman & Wallen, 1999)
ILP/Optimization	(Baker et al., 2002; Mandl, 1979; Papadimitriou & Steiglitz, 1998; Schiffman, 1993; Smith, 1982)
Simulation	(McCarthy, 1999)
Genetic Algorithms	(Michalewicz, 1996)
Ant Colony	(Annaballi, 2002; Gambardella et al., 1999)
Tabu Search	(Chiang & Russell, 1995; Crino, 2002; Cullenbine, 2000; Glover & Laguna, 1997; Harder, 2000; Tamashiro et al., 2002; Wiley, 2001)

As summarized in table 2.1, tabu search is a widely applied method used to solve complex VRPs. However, the programming time and skill necessary to apply tabu search make it a challenge to use. An alternative is to use linear programming. By using the basic structure of a network problem, an optimal solution can be found while incorporating an inherent flexibility to allow for future modifications. Chapter III details the formulation of the problem.

III. Methodology

This chapter discusses the formulation of a model to solve the SBR on-orbit servicing problem. The problem of finding an optimal on-orbit servicing architecture is presented as an instance of a multiple-depot vehicle location and routing problem with time windows. Section 3.1 details the on-orbit servicing problem definition along with major assumptions made and constraints imposed upon the problem.

3.1 The SBR On-Orbit Servicing Problem Definition

The first step in solving the on-orbit servicing problem is defining the problem and what decisions can be made to work towards a solution. This section defines the SBR on-orbit servicing problem through five different factors: the sets of objects in the network, the parameters of the solution search, the decision variables, the objective function, and the constraints on the problem. Also discussed in this section are the assumptions made and the reasoning behind them.

Conceptual Model of the On-Orbit Servicing Network

In the on-orbit servicing network, the nodes represent client satellites, depot spacecraft, and servicing vehicles. The arcs represent the maneuvers between different orbits. Each arc will have costs (in terms of delta-V and time) associated with the chosen maneuver. Servicing vehicles are chosen by selecting their launch into the network, from the supply node on the far left of the network. It is possible to launch any number of each type of servicing vehicle either directly to a client or to a depot spacecraft. If a servicing vehicle is launched to a depot spacecraft, it is launched “dry” with only enough fuel to get it to the depot spacecraft where it will be fueled in preparation for its servicing mission. If a servicing vehicle is launched directly to a client satellite, it is launched “wet” with a full load of fuel. Servicing vehicles travel around the network visiting client satellites until their delta-V and/or ORU capacity is reached, at which time they must either exit the network (de-orbit) or visit a depot spacecraft to re-fuel.

Time throughout the network

In order to track the inventory of each servicing vehicle throughout its route, it is desirable to track the vehicle routes with respect to time. By discretizing the time into units equivalent to the total time to maneuver from one node to another on the network, it becomes possible to look at each servicing vehicle’s inventory at any given time. It also becomes possible to track the balance of ORU and delta-V propellant (i.e. demand) for each client at any given time. Incorporating time as a way to help structure the network results in vehicle routes being combined strings of binary choices, whether or not to choose specific arcs at specific instances in time. The continuous variables representing flow of delta-V and ORUs across arcs are also examined at specific instances in time. Figure 3.1 is a generalized diagram of the network and the arcs available for use at any given time period.

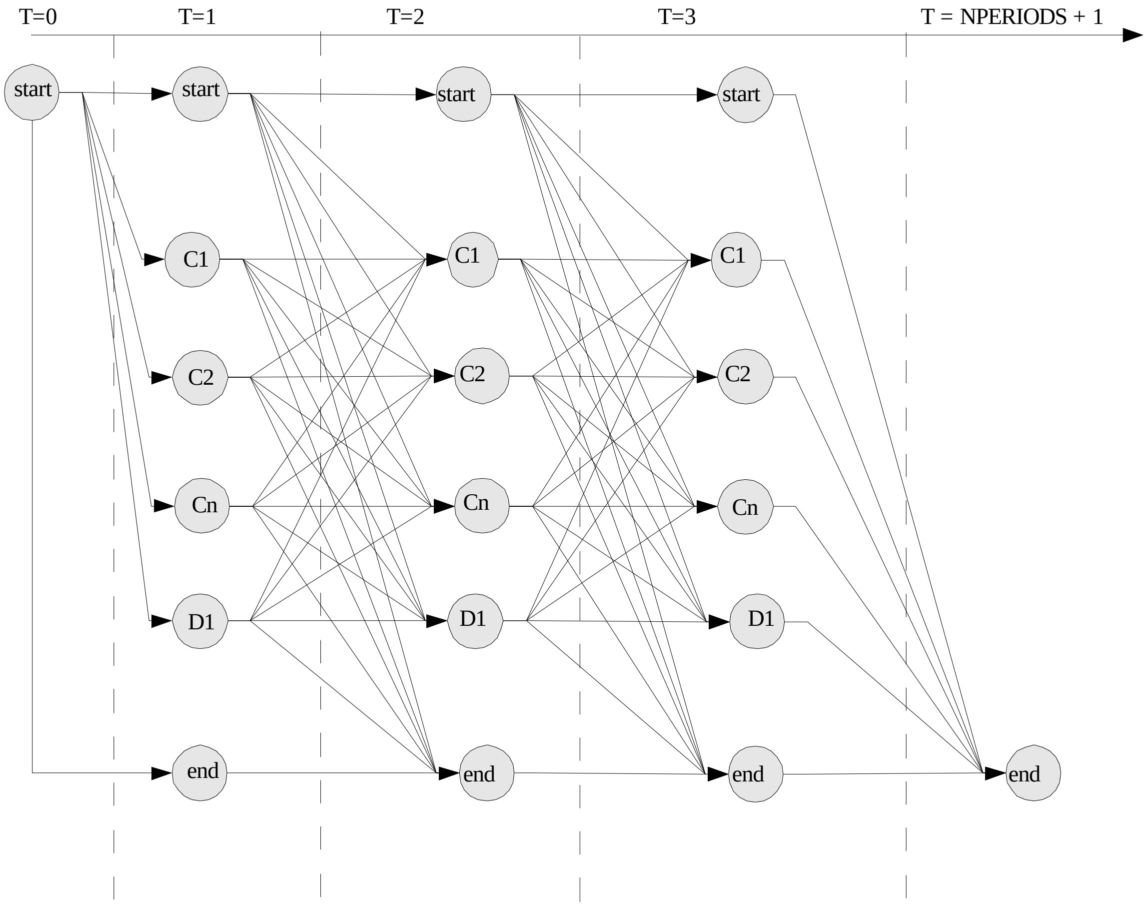


Figure 3.1 Simplified diagram of on-orbit servicing network

In the network diagram in Figure 3.1, the servicing vehicle can choose to use any arc at the beginning of the time period it is currently in. For example, at the beginning of period 0, the only arcs available to choose from leave the start node to any other node. At the beginning of the next period the arcs available depend on what choice was made at the beginning of the last period. The nodes labeled “C” represent client satellite nodes and “D” represents the depot spacecraft node. Once an arc out of the start node is chosen, it is not possible to go back to the start node. Likewise, once an arc is chosen into the exit node, it is not possible to leave that node for any other. Figure 3.2 shows a sample solution for one plane in the network.

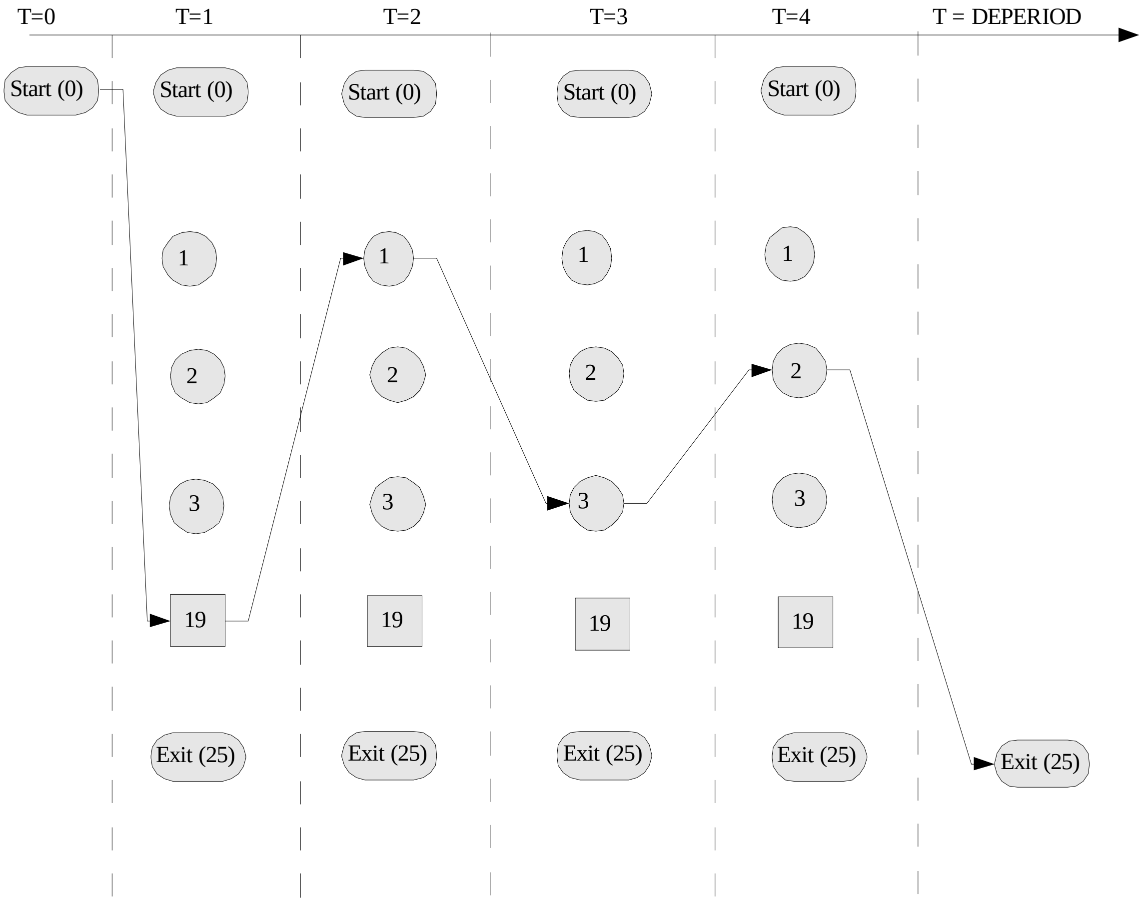


Figure 3.2 Sample solution for one client plane

In the sample solution shown, the servicing vehicle is launched to a depot spacecraft (represented by node 19) at the beginning of period 0. It then travels to client satellites 1, 3, and 2 before exiting the network. In the real world, choosing to go to the exit node is equivalent to staying at the last location visited, however, additional modifications could be made to the model to make the node represent a disposition strategy for the servicing vehicles (de-orbit or boost to hyper-synchronous orbit depending on the client satellite altitude).

3.2 Problem Mathematical Formulation

Sets

For the SBR on-orbit servicing problem the sets of objects in the network considered are client satellites, depot spacecraft, servicing vehicles, and the transfer orbits (arcs) along which servicing vehicles can travel. There are three types of servicing vehicles characterized as either small, medium, or large with their associated capacities varying accordingly. The variable sets are defined as follows:

NODES := {0, 1, 2, . . ., 25}	All nodes
CLIENTS (*C*) := {1, 2, 3, …, 18}	Subset of nodes where clients 1 through 3 are in the first orbital plane, and clients 4 through 6 are in the second orbital plane. There are 6 orbital planes with 3 client satellites in each (Hoy, 2004).
DEPOTS (*D*) := {19, 20, 21, …, 24}	Subset of nodes where one depot spacecraft is assigned per client orbital plane.
DSNODE := {0}	Subset of nodes, dummy start node
DENODE := {25}	Subset of nodes, dummy end node
STYPES := {1, 2, 3}	Servicing vehicle types
S := {1, 2, 3, …, 6}	Servicing vehicle index number (per servicing vehicle type)
V := {1,2}	Required servicing visits to each client satellite
PERIODS := {0, 1, 2, …, 13}	Time periods
DSPERIOD := {0)	Subset of periods
DEPERIOD := {14}	Subset of periods

$$Arcs(i,j) := \left\{\begin{matrix} (0,0) & (0,1) & (0,2) & \cdots & (0,25) \\ (1,1) & (1,2) & (1,3) & \cdots & (1,25) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (24,1) & (24,2) & (24,3) & \cdots & (24,25) \end{matrix}\right\} \qquad \begin{matrix} i \in NODES \\ j \in NODES \end{matrix}$$

Parameters

Client satellites requiring servicing are the “customers” or “demand” nodes. Each of the customers has a specific location given by its specific orbit. The client orbits are given as part of the problem definition, and are discussed in more detail in Chapter 4. Each client satellite has a specific demand in terms of orbit-replaceable unit (ORU) mass and propellant mass. There are also time windows to be considered, as early arrivals may result in too-frequent servicing and maintenance induced failures (Ebeling, 1997). Late arrivals may result in too-infrequent servicing and client satellite failures. However, only upgrade and refueling servicing are being considered in this research. Therefore, although late arrivals are not allowed, later arrivals are favored over earlier arrivals within the time windows for each client.

Depot spacecraft in orbit serve as “supply” nodes where a servicing vehicle can replenish its propellant and ORU stores. Like the client satellites, the orbits in which these depot spacecraft can be located are given as part of the problem definition.

An important factor in formulating a routing problem is the travel cost. Minimizing the cost to travel between nodes in the network is typically a major objective in finding the solution to the proposed problem. In a typical vehicle routing problem, the cost is defined as either time to travel to each customer or the distance between each customer. For the SBR on-orbit servicing problem, travel costs considered are time and delta-V.

In order for any body in orbit to maneuver, it requires a change in velocity, or delta-V. Space mission planners use delta-V as a proxy maneuver cost for propellant usage because it ignores the mass of the object (Kobel, 2004). By calculating the delta-V required for a specific maneuver, planners can consider the mass of the vehicle later in calculating how much propellant will be needed to achieve the desired delta-V.

The added complexity of where to initially locate servicing vehicles requires a definition of the cost to use specific locations. The cost of locating a servicing vehicle either "dry" at a depot or "wet" at a client is reflected in the launch cost, based on the generally accepted figure of $10,000 per pound (Air University, 2003).

The following are the specific parameters used in the SBR on-orbit servicing problem.

demDV_c := demand for delta-V propellant of client c

demORU_c := demand for ORU mass of client c

$\mathrm{timeearly}_{v,c}$:= early time allowed for visit v to client c by a servicer

$\mathrm{timelate}_{v,c}$:= late time allowed for visit v to client c by a servicer

$\mathrm{Txdelta}_{(j,k)}$:= delta-V required for a servicer to move from node j to node k

$\mathrm{Ttime}_{(j,k)}$:= time (quarters) required for a servicer to move from node j to node k

$\mathrm{DeltaCap}_{st}$:= delta-V capacity of servicer type st (for maneuver and delivery)

ORUCap_{st} := ORU carrying capacity of servicer type st (for delivery only)

$\mathrm{costwet}_{st}$:= cost to launch servicer type st fully fueled

$\mathrm{costdry}_{st}$:= cost to launch servicer type st unfueled

$\mathrm{arc}_{st,s,(j,k),t}$:= The decision to move servicing vehicle type st, number s from node j to node k at the beginning of period t is allowed/not allowed (See Appendix C for specific values)

$RHSBAL_{(n,t)}$:= Servicing vehicle balance for node *n* at the beginning of period *t*
(See Appendix C for specific values)

$DVBAL_{(n,t)}$:= Delta-V flow balance for node *n* at the beginning of period *t*
(See Appendix C for specific values)

$ORUBAL_{(n,t)}$:= ORU flow balance for node *n* at the beginning of period *t*
(See Appendix C for specific values)

The arc parameter helps speed calculation by allowing the computer to evaluate only arcs that actually exist in the network. The balance constraints help define the start and depot nodes as supply nodes and the exit node as a "sink" node for all servicing vehicles.

Decision Variables

$w_{st,s,(j,k),t}$:= 1 if servicer type *st* number *s* travels along arc (*j*,*k*) at the beginning of time period *t*
0 otherwise

$flowDV_{st,s,(j,k),t}$:= amount of delta-V transferred by servicer type *st* number *s* along arc (*j*,*k*) at the beginning of time period *t*
This is a continuous variable with a range anywhere from 0 to the delta-V capacity of the servicing vehicle type used.

$flowORU_{st,s,(j,k),t}$:= amount of ORU mass transferred by servicer type *st* number *s* along arc (*j*,*k*) at the beginning of time period *t*
This is a continuous variable with a range anywhere from 0 to the ORU capacity of the servicing vehicle type used.

Objective Function

The objective function of the SBR on-orbit servicing problem seeks to minimize the total launch costs for servicing vehicles while at the same time finding the least expensive (in terms of delta-V) path through the network visiting clients at the latest time possible. Mathematically, it is written as follows:

$$\begin{aligned}
\min & \sum_{st\in stypes}\sum_{s\in servicers}\sum_{c\in clients}\sum_{t\in periods}\left(costwet_{st}+\left(.01*s\right)\right)*arc_{st,s,(0,c),t}\,w_{st,s,(0,c),t}\ + \\
& \sum_{st\in stypes}\sum_{s\in servicers}\sum_{d\in depots}\sum_{t\in periods}\left(costdry_{st}+\left(.01*s\right)\right)*arc_{st,s,(0,d),t}\,w_{st,s,(0,d),t}\ + \\
& \sum_{st\in stypes}\sum_{s\in servicers}\sum_{(j,k)\in nodes}\sum_{t\in periods}(Txdelta_{(j,k)}*.01)*arc_{st,s,(j,k),t}*w_{st,s,(j,k),t}\ + \\
& \sum_{st\in stypes}\sum_{s\in servicers}\sum_{(j,k)\in nodes}\sum_{t\in periods}0.1*(NPERIODS-t)*arc_{st,s,(j,k),t}*w_{st,s,(j,k),t}
\end{aligned}$$

The first two terms of the function include the cost of launching a servicing vehicle either fully fueled (wet) to a client satellite, or un-fueled (dry) to a depot spacecraft. By multiplying the cost value by .01 time the servicing vehicle number drives the choice of smaller indexed servicing vehicles. This was done to differentiate otherwise equivalent solutions, thus speeding overall solution time.

The third term seeks the minimum total delta-V cost for the solution. This term is multiplied by .01 in order to scale down the importance of delta-V relative to launch costs. In future research extensions of this model, the delta-V term can be re-calculated in terms of a dollar cost by calculating the actual propellant used, and so match the units of this term to the rest of the objective function.

The final term drives servicing to as late in the time window for each client as possible, as it is generally better to delay maintenance actions as long as possible (Ebeling, 1997).

Constraints

The on-orbit servicing model includes a number of restrictions that limit the choices made. These constraints include maneuver range and cost for servicing vehicles, time windows for each visit to client satellites, and the capacity for flow of delta-V and ORUs along arcs between nodes. There are also balance constraints for the servicing vehicles, delta-V, and ORUs moving to and from the nodes.

Each servicing vehicle type has a different delta-V capacity. These capacities are derived from Boeing's Orbital Express demonstrator. The range of each servicing vehicle is determined by the maneuvers it makes, or in terms of a network problem, the arcs over which it travels. Each maneuver arc has a unique delta-V and time cost, calculated in a Microsoft Excel spreadsheet provided by the Aerospace Corporation (Kobel, 2004). The total maneuver costs for a servicing vehicle's route cannot exceed its delta-V capacity. Servicing vehicles have the option to replenish their ORU payload and re-fuel by visiting a depot spacecraft.

There is an additional constraint on the problem in the form of time windows. Each client satellite must be serviced between year 4 and 5. This requirement matches the optimal frequency of servicing determined by Waltz (1993). Waltz calculated several "breakpoints" at which servicing is better than satellite replacement. He determined that servicing should be favored over satellite replacement when all of the following occur:

> - ORUs cost less than or equal to 50% of total satellite replacement cost
> - Servicing equipment user charges are less than 50% of total satellite replacement cost
> - Servicing intervals are at least one-third of the time required to replace a satellite
> - Servicing intervals are at least 4 to 5 years
>
> (Waltz, 1993)

For the purposes of this research, it is assumed that the first three of Waltz's criteria will already have been met. The Space-Based Radar constellation will have a baseline (without any servicing) expected life span of 10 years (Hoy, 2004). With servicing every 4 years, the constellation can be upgraded at least twice during its expected life, with the possibility of extending that lifespan.

All of the constraints on the problem are formulated as follows:

The flow of delta-V propellant along arcs must be less than the capacity of the servicing vehicle type used.

$$flowDV_{st,s,(j,k),t} \le DeltaCap_{st} * arc_{st,s,(j,k),t} * w_{st,s,(j,k),t} \qquad (1)$$

$$\forall st \in stypes, s \in servicers, (j,k) \in arcs, t \in periods$$

The flow of ORU mass along arcs must be less than the capacity of the servicing vehicle type used.

$$flowORU_{st,s,(j,k),t} \le ORUCap_{st} * arc_{st,s,(j,k),t} * w_{st,s,(j,k),t} \qquad (2)$$

$$\forall st \in stypes, s \in servicers, (j,k) \in arcs, t \in periods$$

Servicing vehicle node and time period balance constraint

$$\sum_{k\in nodes}\ \sum_{t\in periods} arc_{st,s,(j,k),tp} * w_{st,s,(j,k),tp} - \sum_{k\in nodes}\ \sum_{t\in periods} arc_{st,s,(j,k),t} * w_{st,s,(j,k),t} = RHSBAL_{(j,t)} \qquad (3)$$

$$\forall\ st \in stypes,\ s \in servicers,\ (j,k) \in n$$

Delta-V node and time period balance constraint for clients

$$\sum_{j\in nodes}\ \sum_{tp\in periods:tp+Ttime(j,d)=t} arc_{st,s,(j,c),tp} * flowDV_{st,s,(j,c),tp} - \sum_{k\in nodes} arc_{st,s,(c,k),t} * flowDV_{st,s,(c,k),t} = \qquad (4)$$

$$demDV_{c,t} * \sum_{k\in nodes:c\neq k} arc_{st,s,(c,k),t} * w_{st,s,(c,k),t} + \sum_{k\in nodes} Txdelta_{(c,k)} * arc_{st,s,(c,k),t} * w_{st,s,(c,k),t}$$

$$\forall st \in stypes, s \in servicers, c \in clients, t \in periods$$

Delta-V node and time period balance constraint for depots

$$\sum_{j\in nodes}\ \sum_{tp\in periods:tp+Ttime(j,d)=t} arc_{st,s,(j,d),tp} * flowDV_{st,s,(j,d),tp} - \sum_{k\in nodes} arc_{st,s,(d,k),t} * flowDV_{st,s,(d,k),t} \geq \qquad (5)$$

$$DVBAL_{ct} * \sum_{k\in nodes:d\neq k} arc_{st,s,(d,k),t} * w_{st,s,(d,k),t} + \sum_{k\in nodes} Txdelta_{(d,k)} * arc_{st,s,(d,k),t} * w_{st,s,(d,k),t}$$

$$\forall st \in stypes, s \in servicers, d \in depots, t \in periods$$

Delta-V initial node and time period balance constraint

$$\sum_{j\in nodes:\, j\neq DSNODE}\ \sum_{tp\in periods:tp+Ttime=t} arc_{st,s,(j,DSNODE),tp} * flowDV_{st,s,(j,DSNODE),tp} -$$

$$\qquad (6)$$

$$\sum_{k\in nodes} arc_{st,s,(DSNODE,k),t} * flowDV_{st,s,(DSNODE,k),t} \geq$$

$$DVBAL_{dt} * \sum_{k\in nodes} arc_{st,s,(DSNODE,k),t} * w_{st,s,(DSNODE,k),t} \mid tp + Ttime_{(j,DSNODE)} = t \text{ and } j \neq DSNODE$$

$$\forall st \in stypes, s \in servicers, t \in periods$$

ORU node and time balance constraints for clients

$$\sum_{j\in nodes}\ \sum_{tp\in periods:tp+Ttime=t} arc_{st,s,(j,c),tp} * flowORU_{st,s,(j,c),tp} - \sum_{k\in nodes} arc_{st,s,(c,k),t} * flowORU_{st,s,(c,k),t} = \qquad (7)$$

$$demORU_{c,t} * \sum_{k\in nodes:c\neq k} arc_{st,s,(c,k),t} * w_{st,s,(c,k),t} \forall st\in stypes, s\in servicers, c\in clients, t\in periods$$

ORU node and time balance constraints for depots

$$\sum_{j\in nodes}\ \sum_{tp\in periods:tp+Ttime=t} arc_{st,s,(j,d),tp} * flowORU_{st,s,(j,d),tp} - \sum_{k\in nodes} arc_{st,s,(d,k),t} * flowORU_{st,s,(d,k),t} \geq \qquad (8)$$

$$-ORUCap_{st} * \sum_{k\in nodes:d\neq k} arc_{st,s,(d,k),t} * w_{st,s,(d,k),t} \forall st\in stypes, s\in servicers, d\in depots, t\in periods$$

ORU initial node and time balance constraint

$$\sum_{j\in nodes}\ \sum_{tp\in periods:tp+Ttime=t} arc_{st,s,(j,DSNODE),tp} * flowORU_{st,s,(j,DSNODE),tp} - \qquad (9)$$

$$\sum_{k\in nodes:DSNODE\neq j} arc_{st,s,(DSNODE,k),t} * flowORU_{st,s,(DSNODE,k),t} \geq$$

$$-ORUCap_{st} * \sum_{k\in nodes} arc_{st,s,(DSNODE,k),t} * w_{st,s,(DSNODE,k),t} \forall st\in stypes, s\in servicers, t\in periods$$

All clients must have a servicing vehicle arrive between the early time and the beginning of the late time for that visit.

$$\sum_{st\in stypes}\ \sum_{s\in servicers}\ \sum_{j\in nodes}\ \sum_{t\in periods} arc_{st,s,(j,c),t} * w_{st,s,(j,c),t} = 1 \qquad (10)$$

$$\forall c\in clients, v\in visits \mid timeearly_{v,c} \leq t + Ttime_{(j,c)} \leq timelate_{v,c} \text{ and } 0 < k \neq c$$

All clients must have a servicing vehicle leave between the early time and the beginning of the late time for that visit.

$$\sum_{st\in stypes}\ \sum_{s\in servicers}\ \sum_{k\in nodes}\ \sum_{t\in periods} arc_{st,s,(c,k),t} * w_{st,s,(c,k),t} = 1 \qquad (11)$$

$$\forall c\in clients, v\in visits \mid timeearly_{v,c} \leq t \leq timelate_{v,c} \text{ and } 0 < k \neq c$$

3.3 Assumptions

This section describes the assumptions made in the on-orbit servicing problem and the rationale behind each.

The first assumption is that the technology to make servicing possible is mature and in place. This technology includes, but is not limited to, robotics, cryogenic storage, spacecraft interfaces, etc. Although this technology is not yet mature, advanced technology demonstrators like Orbital Express show that it could be possible within a decade or two. This assumption is made because the purpose of this research is to determine how to best utilize the resources of a servicing architecture, not to determine what technologies need to be in place to allow servicing to occur.

The client constellation is also assumed to be fully operational and at day 0 of its active life. In real life, full constellations are considered "operational" only after a number of individual satellites are placed on orbit, tested, and maneuvered into their operational orbits. This process takes time, sometimes months or years. However, since this research focuses only on the servicing portion of mission planning, the complexity of considering individual satellite initial operational capability schedules is beyond the scope of this study. Further research into this area may consider the advanced time constraints of having different client satellites become operational at different times.

Another assumption made in this research is that the mass of the ORUs remain constant. The current concept of operations for the Orbital Express program has the servicing vehicle remove old components and replace them with the new ones (DARPA, 2004). Since satellites operate in an extremely harsh environment (radiation, temperature extremes, vibration, and more) all components must be shielded in order to function properly. The majority of the mass of an ORU would consist of shielding. The shielding requirements of an ORU will remain unchanged from year to year, and shielding technology is not expected to make radical advances in the near future. Therefore, the mass of any individual ORU is considered constant throughout the time horizon of the model.

This model assumes that servicing vehicles of the same type all have identical capabilities (mass delivery capacity, range, etc.). Once the investment is made to design, produce, and procure servicing vehicles, significant changes in servicing system design are not expected, nor can they be accurately anticipated and incorporated into the model.

The problem also assumes that each client satellite has an equal and stationary demand. Although propellant usage will vary from satellite to satellite, the primary purpose of servicing the SBR constellation is upgrade, specifically processor upgrade according to DARPA (2004), and the components on each satellite will not be significantly different. The possibility of variable or dynamic demand is beyond the scope of this research.

The time periods used are equal to 90-day increments. 90 days is a long enough length of time to efficiently use delta-V for orbit transfer maneuvers. Although some maneuvers may take less than 90 days to complete, the large time window accounts for the servicing time for each client or depot spacecraft as well as allowing flexibility for mission planners and a time buffer for real-world scheduling issues.

The actual time required to rendezvous with, dock with, and service a satellite is unknown. The closest examples available are Space Shuttle missions. Space Shuttles were examined as a possible high-end time limit for individual servicing actions. Shuttle missions often involve repeated servicing actions to a satellite, and involve a great deal of time for astronauts to prepare themselves to safely operate outside the spacecraft. Some of these missions can be extremely lengthy. For example, the longest Space Shuttle mission to-date was STS-80 in 1996, which took 17 days and serviced two different satellites along with performing seven on-board experiments and enduring a delayed landing due to poor weather (NASA, 2004c). Autonomous servicing will likely not be as involved. The client satellites in the SBR constellation will be designed to easily accommodate autonomous servicing by reducing the complexity of any servicing task when compared to human servicing of satellites. Even though autonomous servicing tasks are expected to be less complex than human servicing tasks, past human performance can be used as a baseline estimate for autonomous servicing times. The longest extravehicular activity (servicing of a satellite) ever accomplished by an astronaut was 9 hours (NASA, 2004c). Using this as a conservative estimate, the servicing is assumed to be completed within the same time period that the decision is made to depart from a client to another node. Since all of the time periods are in 90-day increments, this allows for more than enough time to complete servicing and maneuver to the next node in the network.

It is difficult to anticipate the mass of an orbit-replaceable unit. Each component of a satellite may have a different mass and different shielding requirement than any other. In addition, manufacturers may design similar components differently. Without knowing specifically what components of the SBR client satellites are likely candidates for servicing, the only recourse is to use a similar demand function as other on-orbit studies. Leisman and Wallen’s (1999) study used three different values for an ORU demand for the GPS constellation, 50 Kg, 150 Kg, and 300 Kg, as specifically requested by DARPA (the agency sponsoring the research). The median value of 150 was used in this model as a conservative approximation for ORU demand, although the flexibility of the model allows those values to be changed as the user desires.

The second commodity demanded by clients is delta-V. Although highly-maneuverable clients are not considered in this research, all satellites must expend some delta-V to keep within their assigned orbits. This station-keeping delta-V varies depending on the orbit of the satellite and many other factors such as solar activity, atmospheric drag, and orbital perturbations due to the oblateness of the Earth (Wertz, Collins, Dawson, Koenigsmann & Potterveld, 1997). Calculating the exact station-keeping delta-V required for the SBR clients is beyond the scope of this research, however, Wertz et al. (1997) offer an estimate of around 10 m/s per year for satellites with an altitude above 1,500 Km. Combining this value with servicing every 5 years gives a delta-V demand value of 50 per visit to each client.

IV. Model Implementation

4.1 Model Coding

Translation of equations into Mosel language

Using Xpress-MP optimization software, the model was translated into Mosel, a computer programming language that closely parallels mathematical expressions. Appendix C contains a copy of the code used. Mosel allows an almost direct translation of mathematical equations and is relatively easy for anyone with basic programming skills to understand. The Xpress-MP optimization software can also translate the Mosel code into C, C++, Java, and Visual Basic formats.

Full versus relaxed model

If all variables possible are considered, the on-orbit servicing problem becomes extremely large when formulated. There are different types of servicing vehicles, depot spacecraft, and different orbits where depot spacecraft can be located. There are also different routes which servicing vehicles can take, and different times at which they can move from one arc to another. Theoretically, there are in infinite number of orbits from which to choose for depot spacecraft locations. By limiting the number of depot locations to the same plane as the client satellites (though at variable different altitudes) and directly between any two client planes, most of the likely depot locations are considered while eliminating locations with only minor influence on possible solutions. This still results in a network with 86 nodes. Considering all of the choices available at any given time within the 10-year planned client lifetime, results in a problem with

47,926,080 variables. Solving a problem this large through integer linear programming is effectively intractable given current computing power, as well as time prohibitive.

In order to solve the problem, reasonable reductions in size must be considered. The first reduction is in the number of servicing vehicles available to use. The number of available servicing vehicles is limited to no more than 6 of any type. For the SBR constellation, clients occupy six orbital planes with three satellites in each plane. Because the time window was set at four quarters for the first visit, and any maneuver takes one quarter to complete, a servicing vehicle can only visit four nodes within the first time window. This limits the number of clients that any servicing vehicle can visit within the first time window to four or fewer. Even the smallest servicing vehicle type has more than enough capacity to service all client satellites in any plane. Although it is feasible to use more than one servicing vehicle per client plane, if one servicing vehicle can meet the demand, using more than one is not a logical alternative, given the objective, parameters, and assumptions of this problem.

Limiting the number of depot spacecraft locations to one per client plane reduces the number of nodes in the network to 26 (one per client, one per depot, a start or "launch" node, and an exit node). Maneuvers from one orbital plane to another are very costly in terms of delta-V. By having a depot spacecraft located in each of the client planes, the need to travel to a different plane just to replenish a servicing vehicle is eliminated. The inactive time between servicing windows can be eliminated because it is not necessary to make any decision to move during this time, and so not necessary to model it. Further limiting the time from the beginning of the fourth year of the client's lifetime to the last year for servicing brings the number of time variables (quarters) from

40 down to 13. This brings the number of variables down to 474,552. Although still very large, a problem of this size can be successfully solved by integer linear programming.

4.2 Data

The data used for to construct the model were based off information from the Phase 1 findings of Boeing's Orbital Express program. Appendix D lists the specific servicing vehicle parameters (mass, delta-V capacity, ORU delivery capacity, etc.) used in the model.

The SBR constellation orbital parameters were derived from Hoy (2004). For a brief explanation of the description of satellite orbits, refer to Appendix A. The SBR constellation used in the model consists of 18 client satellites in six orbital planes. Each orbital plane is assumed circular at an altitude of 1,000 nautical miles, or 1,842 Km and an inclination of 50°. The specific pattern of the constellation is unavailable, since the program is still under development and these decisions have not yet been finalized. Therefore, a simple pattern was used for client satellites, spacing them evenly, with a mean anomaly difference of 120° between each client. Further, the Right Ascension of each client plane was evenly spaced with a 60° difference. Depot spacecraft locations were arbitrarily chosen to be between two client satellites in each plane. A simple illustration of the relative locations of the SBR planes is shown in Figure 4.1.

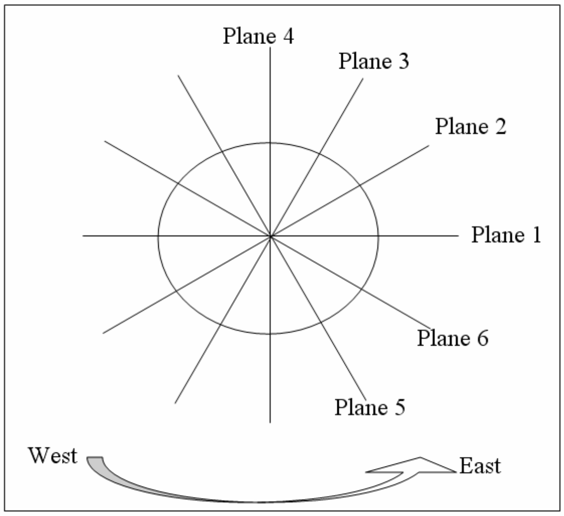


Figure 4.1 Relative positions of SBR constellation planes

Figure 4.2 illustrates the relative positions of the client satellites and depot spacecraft in the Plane 1. Nodes 1, 2, and 3 represent client satellite positions and node 19 represents the depot spacecraft position. All six planes follow an identical position scheme as shown.

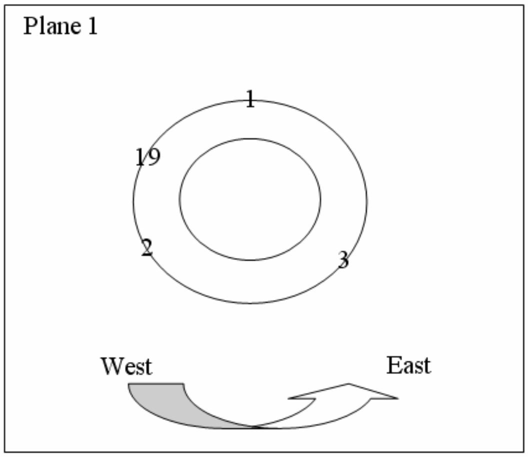


Figure 4.2 Relative positions of client satellite and depot spacecraft nodes within client plane

After determining the relative positions of each client satellite and depot spacecraft, the delta-V required for a servicing vehicle to move from any node to any other node was calculated using the spreadsheet provided by the Aerospace Corporation (Kobel, 2004). The delta-V required for any maneuver was calculated as the least amount of delta-V required to make the maneuver in 90 days. From these data, matrices were constructed and programmed into the model to give the costs for using arcs between nodes. For moves that were not allowed, from the exit node to any other node for example, the time cost was defined as the number of periods + 1, and the delta-V cost was defined as the maximum capacity of the servicing vehicle used + 1. Coding the cost function this way simplified coding of the constraints while at the same time eliminating forbidden arcs from possible solutions. It also reduced calculation time by allowing the optimization program to pre-solve parts of the problem and eliminate them. Appendix B lists the transfer delta-V and transfer time matrices used in the model.

4.3 Results

The model was run on a desktop personal computer with dual 2.8 GHz Intel Xeon processors and 3.0 GB of RAM. To reduce computing time, the model was run in two stages, one for each servicing visit time window. After solving for the optimal solution for the first servicing visit, the routing for each servicing vehicle was fixed as a starting solution for the second stage. However, the servicing vehicle type used in the second stage run was to variable. This allowed for the possibility of using a larger, more capable servicing vehicle for the first visit and needing to use fewer vehicles for the second visit.

Xpress-MP was able to use the ceded formulation of parameters and constraints to pre-solve the solution matrix from 170,388 variables (columns) and 151,596 constraints (rows) down to 40,014 variables and 29,520 constraints. Using the Newton-Barrier method, the LP relaxation took only 0.1 seconds and gave an objective function lower bound of 201.564. A branch-and-bound global search took 169.5 seconds, examined 10 nodes, and found the first integer solution optimal.

The first stage solution used one small servicing vehicle per client plane, launching the vehicles dry to a depot spacecraft, then visiting each of the three clients in that plane, and then returning to the depot. Table 4.1 lists the first stage solution.

Table 4.1 First stage solution

Servicing vehicle type, index #				Servicing vehicle type, index #			
Servicer 1,1	Travel to	Flow delta-V	Flow ORU	Servicer 1,4	Travel to	Flow delta-V	Flow ORU
Time 0	Depot 20	0	0	Time 0	Depot 22	0	0
Time 1	Client 5	154.2	450	Time 1	Client 11	154.2	450
Time 2	Client 6	102.5	300	Time 2	Client 12	102.5	300
Time 3	Client 4	50.8	150	Time 3	Client 10	50.8	150
Time 4	Depot 20	0	0	Time 4	Depot 22	0	0
Servicer 1,2	Travel to	Flow delta-V	Flow ORU	Servicer 1,5	Travel to	Flow delta-V	Flow ORU
Time 0	Depot 23	0	0	Time 0	Depot 21	0	0
Time 1	Client 13	154.2	998	Time 1	Client 8	154.2	450
Time 2	Client 15	102.5	848	Time 2	Client 9	102.5	300
Time 3	Client 14	50.8	698	Time 3	Client 7	50.8	150
Time 4	Depot 23	0	548	Time 4	Depot 21	0	0
Servicer 1,3	Travel to	Flow delta-V	Flow ORU	Servicer 1,6	Travel to	Flow delta-V	Flow ORU
Time 0	Depot 24	0	0	Time 0	Depot 19	0	0
Time 1	Client 17	154.2	450	Time 1	Client 1	154.2	450
Time 2	Client 18	102.5	300	Time 2	Client 3	102.5	300
Time 3	Client 16	50.8	150	Time 3	Client 2	50.8	150
Time 4	Depot 24	0	0	Time 4	Depot 19	0	0

Although the ORU flow values for servicing vehicles 1,2 and 1,5 are different, in terms of the clients, the values are equivalent as the same amount is being delivered in each case.

Following the first stage solution, the servicing vehicle routes were fixed and the model re-run looking at 13 periods to cover both servicing visits. Xpress-MP again pre-solved the matrix from 459,720 variables with 382,998 constraints down to 161,574 variables with 118,530 constraints. The LP relaxation (again using Newton-Barrier) took 0.9 seconds and found the lower bound for the objective function of 239.67. The branch-and-bound global search examined 82 nodes, with the second integer solution found being the optimal solution in 281.7 seconds. The second stage solution maintained the use of small servicing vehicles to complete the first stage fixed first visit routes and continued their use for the routes determined for the second servicing visit. Table 4.2 lists the solution to the second-stage model run.

Table 4.2 Second stage solution

Servicing vehicle type, index #				Servicing vehicle type, index #			
Servicer 1,1	Travel to	Flow DV	Flow ORU	Servicer 1,4	Travel to	Flow DV	Flow ORU
Time 0	Depot 20	0	0	Time 0	Depot 22	0	0
Time 1	Client 5	154.2	450	Time 1	Client 11	154.2	450
Time 2	Client 6	102.5	300	Time 2	Client 12	102.5	300
Time 3	Client 4	50.8	150	Time 3	Client 10	50.8	150
Time 4	Depot 20	0	0	Time 4	Depot 22	0	0
Time 5	Client 4	153.4	450	Time 6	Client 11	153.4	450
Time 7	Client 5	101.7	300	Time 8	Client 10	101.7	300
Time 8	Client 6	50	150	Time 11	Client 12	50	150
Time 12	exit	0	0	Time 12	exit	0	0
Servicer 1,2	Travel to	Flow DV	Flow ORU	Servicer 1,5	Travel to	Flow DV	Flow ORU
Time 0	Depot 23	0	0	Time 0	Depot 21	0	0
Time 1	Client 13	154.2	450	Time 1	Client 8	154.2	450
Time 2	Client 15	102.5	300	Time 2	Client 9	102.5	300
Time 3	Client 14	50.8	150	Time 3	Client 7	50.8	150
Time 4	Depot 23	0	0	Time 4	Depot 21	0	0
Time 9	Client 14	328	450	Time 6	Client 7	153.4	998
Time 10	Client 15	276.3	300	Time 9	Client 8	101.7	848
Time 11	Client 13	224.6	150	Time 10	Client 9	50	698
Time 12	exit	174.6	0	Time 12	exit	0	548
Servicer 1,3	Travel to	Flow DV	Flow ORU	Servicer 1,6	Travel to	Flow DV	Flow ORU
Time 0	Depot 24	0	0	Time 0	Depot 19	0	0
Time 1	Client 17	154.2	450	Time 1	Client 1	154.2	450
Time 2	Client 18	102.5	300	Time 2	Client 3	102.5	300
Time 3	Client 16	50.8	150	Time 3	Client 2	50.8	150
Time 4	Depot 24	0	0	Time 4	Depot 19	0	0
Time 6	Client 16	153.4	998	Time 6	Client 2	328	998
Time 8	Client 18	101.7	848	Time 8	Client 1	276.3	848
Time 10	Client 17	50	698	Time 11	Client 3	224.6	698
Time 12	exit	0	548	Time 12	exit	174.6	548

An attempt was made to run the full model without fixing the first stage route solution in order to verify that the solution given in the two-stage process was indeed optimal. The pre-solved matrix was reduced to 230,688 variables with 165,798 constraints and the LP relaxation using Newton Barrier gave the lower bound for the objective function at 239.67 in approximately 60 seconds. However, the global search did not return an integer solution after running the model for over 24 hours. Despite this, the fact that the solution obtained from the two-stage method achieved the lower bound found for the full model, supports the use of the two-stage method.

V. Conclusions and Areas for Future Study

5.1 Conclusions

This model gives the optimal solution for the given data set. It can be used as a tool to assist planners in the early stages of the acquisition process (Phase 1-2) of an on-orbit servicing system before the final numbers of servicing vehicles and depot spacecraft have been determined. The model is dependant on accurate information about the capabilities of servicing vehicles, and the make up and demand of the client satellite constellation. The solution provided by the model can be used to facilitate calculation of break-even points for the decision to design new satellite systems for on-orbit servicing.

The code allows modifications to client demands and time windows as well as the servicing assets available. Changes to the client satellite constellation pattern can be modeled with the appropriate changes in the transfer delta-V cost matrix. Changes in client satellite demand or servicing vehicle capabilities can also be addressed. Appendix C lists the full code as written in Mosel for the Xpress-MP optimization software.

While there are many real-world complexities that could still be added into the model, increasing the complexity may require the use of other methods to obtain solutions in a reasonable amount of time. The model could be used as part of a meta-heuristic technique to solve more complex problems.

5.2 Limitations

The limitations to this research stem from the developing nature of on-orbit servicing and the complexity of the problem. On-orbit servicing is not mature enough to be commonly incorporated into satellite designs, therefore the model is applicable only to future satellite constellations. The model as built only examines servicing architectures for delivery of two commodities, delta-V propellant and orbit-replaceable units. The formulation must be modified to incorporate other servicing tasks or additional commodities. The model is also not capable of automatically determining the impact of a change in the client constellation configuration or demands. The effect of adding or deleting depot spacecraft and locations on the solution also cannot be predicted. The model must be run with new data in order to perform such sensitivity analyses.

The model only provides the optimal servicing architecture and routing to meet the demands of the client constellation and does not evaluate the impact of servicing on the client satellites. Stochastic demands by the client satellites are not considered in this work, neither is the possibility of unsuccessful servicing visits. The cost feasibility of the determined solution is not examined, though the solution is the lowest cost with the given data. Other costs such as depot spacecraft re-supply missions, launch processing time, and possible maintenance-induced failures are not evaluated.

5.3 Areas for Future Research

Autonomous on-orbit servicing is a developing field of study. On-orbit servicing, and space logistics in general, is an area that needs further study as the war-fighters increase their reliance on sustainable, flexible, and effective space assets. This model is a first step to determining the best way to manage resources in the unique operating environment of space.

A next logical step is increasing the complexity and realism of the model. Additions can include varying the number, types, and locations of depot spacecraft, calculating actual propellant usage, or increasing the granularity of the time steps. Other techniques may be applied to track the inventory of the depot spacecraft and servicing vehicles instead of modeling them as flows through the network. Including stochastic or unequal demand from clients, the launch costs for depot spacecraft and commodity re-supply missions will further improve the realism and validity of the model. As different launch vehicles become available, the cost to launch may also change, and the servicing vehicles' capabilities may improve over time. Optimizing the launch vehicles and launch sites used could also provide significant benefit to mission planners, allowing launch from Vandenberg Air Force Base, California, Patrick Air Force Base, Florida, or other launch sites that may become available in the future.

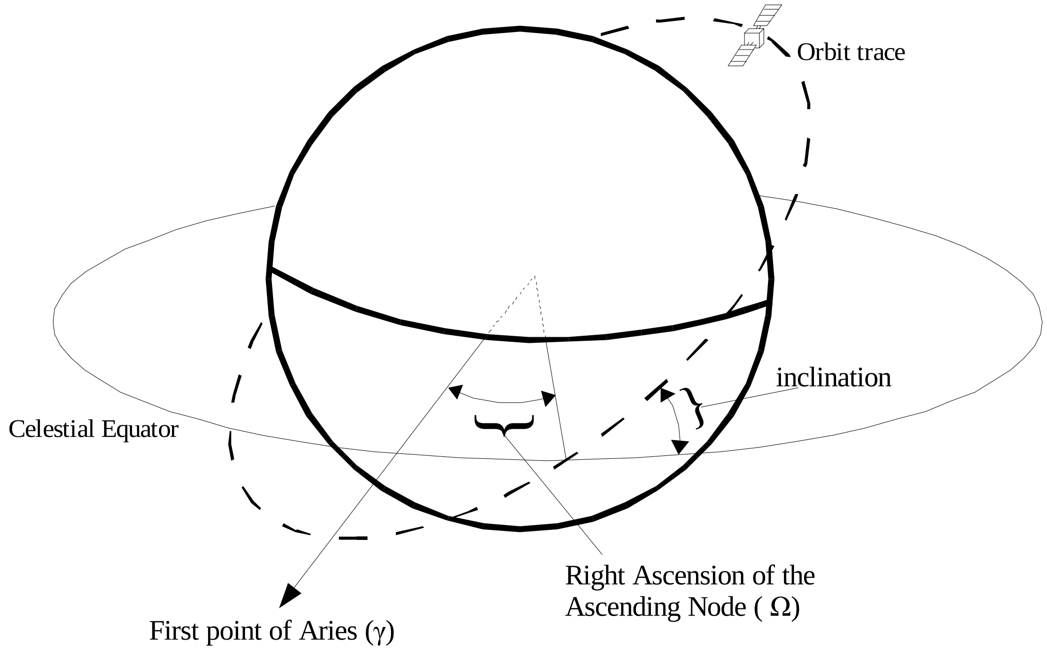
The possibility of servicing failures should also be incorporated along with the impact of on-orbit servicing to overall client satellite mission capability. Future researchers should also consider applying the model to different satellite systems. An interesting study would be to apply the model to a constellation of highly maneuverable satellites that would have a high and variable demand for delta-V. Contributions to space logistics research could also be made by looking at other servicing tasks such as assembly, inspection, and re-boosting satellites from degraded or improper orbits.

5.4 Summary

This research effort provides a first step in solving the complex and difficult problem of finding optimal the on-orbit servicing architecture for a client satellite constellation. A brief background on the current and future efforts of on-orbit servicing was provided along with a discussion of methods available for solving problems like the on-orbit servicing problem. By defining the problem as a minimum cost flow network, it was possible to apply integer linear programming and find the optimal solution within a reasonable amount of time using Xpress-MP optimization software. Following a discussion of the results, areas for improvement and future research were presented.

Appendix A. Description of Satellite Orbits

Because satellites are constantly moving, their locations are described by their specific orbits. Satellite orbits are described in terms of their locations in space relative to accepted fixed reference points (Air University, 2003). The orbits used in this study can be described based on their altitude above the Earth's surface, their eccentricity, and their inclination. An orbit's inclination is the angle between the plane of the orbit and the celestial equator, as illustrated in Figure A.1.

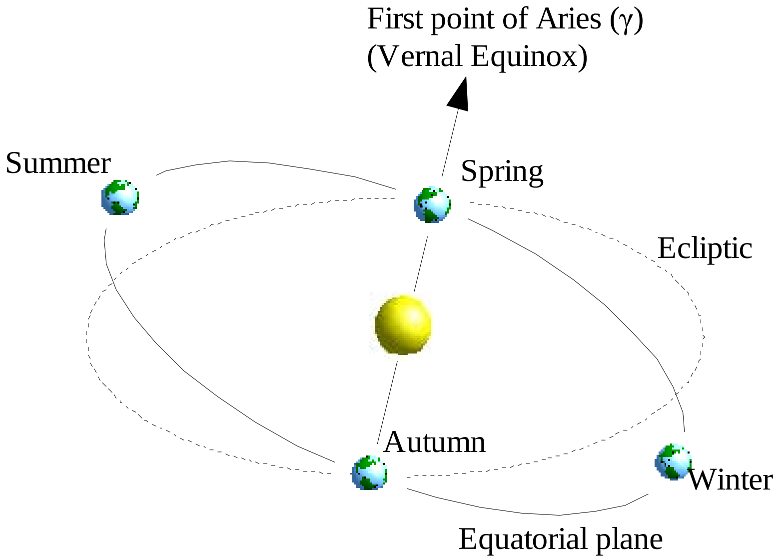


Adapted from Air University Space Primer (2003) http://space.au.af.mil/primer/

Figure A.1 Inclination and Right Ascension of satellite orbits

The eccentricity of an orbit describes the shape of the orbit. An eccentricity of 0 means the orbit is circular. An eccentricity between 0 and 1 means the orbit is elliptical in shape (Air University, 2003). Eccentricities of 1 or greater, or less than 0 refer to parabolic and hyperbolic shapes, and thus are not relevant to Earth-orbiting satellites.

The other basic description of a satellite's orbit is the Right Ascension of the Ascending Node. This refers to the angle between the point at which the satellite crosses the celestial equator while moving in a South-to-North direction and the first point of Aries (see Figure A.1). The first point of Aries is the direction towards the Vernal Equinox along a line drawn through the intersection of the Earth's equatorial plane and the ecliptic (the path along which the Earth travels around the Sun). Figure A.2 illustrates the direction of the Vernal Equinox.



Adapted from Air University Space Primer (2003) http://space.au.af.mil/primer

Figure A.2 Determination of the first point of Aries

A satellite's position in space is further described by its relative position along its orbit trace. The angle between a satellite's current position along its orbit trace and the orbit's Right Ascension is called its mean anomaly. A mean anomaly of 90° would mean that the satellite is 90° from the equator.

Appendix B. Transfer Delta-V and Time Matrices

Table B.1 Transfer delta-V required for node transfers

Transfer Delta-V Required												
	To											
From	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1.7	1.7	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8
2	1.7	0	1.7	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8
3	1.7	1.7	0	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8
4	287.4	287.4	287.4	0	1.7	1.7	239	239	239	443.9	443.9	443.9
5	287.4	287.4	287.4	1.7	0	1.7	239	239	239	443.9	443.9	443.9
6	287.4	287.4	287.4	1.7	1.7	0	239	239	239	443.9	443.9	443.9
7	594.9	594.9	594.9	287.4	287.4	287.4	0	1.7	1.7	239	239	239
8	594.9	594.9	594.9	287.4	287.4	287.4	1.7	0	1.7	239	239	239
9	594.9	594.9	594.9	287.4	287.4	287.4	1.7	1.7	0	239	239	239
10	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4	0	1.7	1.7
11	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4	1.7	0	1.7
12	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4	1.7	1.7	0
13	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4
14	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4
15	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4
16	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9
17	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9
18	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9
19	1.7	1.7	2.5	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8
20	287.4	287.4	287.4	1.7	1.7	2.5	239	239	239	443.9	443.9	443.9
21	594.9	594.9	594.9	287.4	287.4	287.4	1.7	1.7	2.5	239	239	239
22	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4	1.7	1.7	2.5
23	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9	287.4	287.4	287.4
24	239	239	239	443.9	443.9	443.9	633.8	633.8	633.8	594.9	594.9	594.9

Transfer Delta-V Required												
	To											
From	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
1	594.9	594.9	594.9	287.4	287.4	287.4	0.8	239	443.9	633.8	594.9	287.4
2	594.9	594.9	594.9	287.4	287.4	287.4	0.8	239	443.9	633.8	594.9	287.4
3	594.9	594.9	594.9	287.4	287.4	287.4	2.5	239	443.9	633.8	594.9	287.4
4	633.8	633.8	633.8	594.9	594.9	594.9	287.4	0.8	239	443.9	633.8	594.9
5	633.8	633.8	633.8	594.9	594.9	594.9	287.4	0.8	239	443.9	633.8	594.9
6	633.8	633.8	633.8	594.9	594.9	594.9	287.4	2.5	239	443.9	633.8	594.9
7	443.9	443.9	443.9	633.8	633.8	633.8	594.9	287.4	0.8	239	443.9	633.8
8	443.9	443.9	443.9	633.8	633.8	633.8	594.9	287.4	0.8	239	443.9	633.8
9	443.9	443.9	443.9	633.8	633.8	633.8	594.9	287.4	2.5	239	443.9	633.8
10	239	239	239	443.9	443.9	443.9	633.8	594.9	287.4	0.8	239	443.9
11	239	239	239	443.9	443.9	443.9	633.8	594.9	287.4	0.8	239	443.9
12	239	239	239	443.9	443.9	443.9	633.8	594.9	287.4	2.5	239	443.9
13	0	1.7	1.7	239	239	239	443.9	633.8	594.9	287.4	0.8	239
14	1.7	0	1.7	239	239	239	443.9	633.8	594.9	287.4	0.8	239
15	1.7	1.7	0	239	239	239	443.9	633.8	594.9	287.4	2.5	239
16	287.4	287.4	287.4	0	1.7	1.7	239	443.9	633.8	594.9	287.4	0.8
17	287.4	287.4	287.4	1.7	0	1.7	239	443.9	633.8	594.9	287.4	0.8
18	287.4	287.4	287.4	1.7	1.7	0	239	443.9	633.8	594.9	287.4	2.5
19	594.9	594.9	594.9	287.4	287.4	287.4	0	239	443.9	633.8	594.9	287.4
20	633.8	633.8	633.8	594.9	594.9	594.9	287.4	0	239	443.9	633.8	594.9
21	443.9	443.9	443.9	633.8	633.8	633.8	594.9	287.4	0	239	443.9	633.8
22	239	239	239	443.9	443.9	443.9	633.8	594.9	287.4	0	239	443.9
23	1.7	1.7	2.5	239	239	239	443.9	633.8	594.9	287.4	0	239
24	287.4	287.4	287.4	1.7	1.7	2.5	239	443.9	633.8	594.9	287.4	0

Appendix C. Copy of Model Code in Mosel

The following is the code for the model as written in Xpress-MP. The language closely follows mathematical equations. Lines that are in italics are comments only and are not considered part of the model by the computer. The character “|” is read as a condition that must be met.

```
model "On-Orbit Servicer"
uses "mmxprs","mmive"
options noimplicit
```

This section defines the parameters for the model, the number of clients and depot spacecraft, servicing vehicles, time periods, etc.

```
parameters
        DSNODE = 0
        DSPERIOD = 0
        NCLIENTS = 18
        NVISITS = 2
        NLOCATIONS = 0
        NSTYPES = 3
        NSERVICERS = 6
        NPERIODS = 13
        NDEPOTS = 6
        DENODE = NCLIENTS + NDEPOTS + 1
        DEPERIOD = NPERIODS + 1
        NNODES = NCLIENTS + NDEPOTS
end-parameters
```

This section defines the sets used in the model as either a range of numbers or a real number

```
declarations
        DELTAMAX: real
        ORUMAX: real
        nodes: range
        clients: range
        depots: range
        stypes: range
        periods: range
        servicers: range
        visits: range
end-declarations
```

The sets used are specifically defined in this section

```
stypes := 1..NSTYPES
nodes := DSNODE..DENODE
depots := NCLIENTS+1..NCLIENTS+NDEPOTS
clients := 1..NCLIENTS
periods := DSPERIOD..DEPERIOD
servicers := 1..NSERVICERS
visits := 1..NVISITS
```

These are the decision variables for the model. The w variable is binary while the flowDV and flowORU variables are continuous.

```
declarations
```

VARIABLES

```
        w: array(stypes,servicers,nodes,nodes,periods) of mpvar
        flowDV: array(stypes,servicers,nodes,nodes,periods) of mpvar
        flowORU: array(stypes,servicers,nodes,nodes,periods) of mpvar
```

PARAMETERS

The arc parameter determines which w decisions are allowed. When set to 0, the move between those two nodes is forbidden. This keeps spurious variables from being produced.

```
        arc: array(stypes,servicers,nodes,nodes,periods) of integer
```

This section tells the computer to look for the values of these parameters in arrays based on the criteria listed in parentheses.

```
        demDV: array(clients) of real
        demORU: array(clients) of real
        timeearly: array(visits,clients) of real
        timelate: array(visits,clients) of real
        Txdelta: array(nodes,nodes) of real
        Ttime: array(nodes,nodes) of real
        DeltaCap: array(stypes) of real
        ORUCap: array(stypes) of real
        costdry: array(stypes) of real
        costwet: array(stypes) of real

        RHSBAL: array(nodes,periods) of real
        DVBAL: array(nodes,periods) of real
        ORUBAL: array(nodes,periods) of real
        DVCAP: array(stypes,servicers,nodes,nodes,periods) of linctr
        ORUCAP: array(stypes,servicers,nodes,nodes,periods) of linctr
        NEWBALANCE: array(stypes,servicers,nodes,periods) of linctr
        DVBALANCE: array(stypes,servicers,nodes,periods) of linctr
        ORUBALANCE: array(stypes,servicers,nodes,periods) of linctr
        LEAVECLIENT: array(visits,clients) of linctr
        ENTERCLIENT: array(visits,clients) of linctr
        Cost: linctr
        FirstStage: array(servicers,nodes,nodes,periods) of linctr
end-declarations
```

These terms define the costs to launch the servicing vehicles wet or dry

```
costwet := [Cost in millions, figures available in Appendix D]
costdry := [Cost in millions, figures available in Appendix D]
```

This section sets the client demand values and time windows for each visit.

```
forall( c in clients | c > DSNODE and c <= NCLIENTS ) do
        demDV(c) := 10
        demORU(c) := 150
        timeearly(1,c) := 1
        timelate(1,c) := 4
        timeearly(2,c) := 6
        timelate(2,c) := NPERIODS
end-do
```

These terms define the capacities of each servicing vehicle type

```
DeltaCap := [ in m/s, figures available in Appendix D ]
ORUCap := [in Kg, figures available in Appendix D]
```

This section is used to allow the flow of commodities from the depot spacecraft or dummy start node to be set at no more than the capacity for the servicing vehicle type chosen.

```
forall( st in stypes ) do
        if DELTAMAX < DeltaCap(st)
        then DELTAMAX := DeltaCap(st)
        end-if
        if ORUMAX < ORUCap(st)
        then ORUMAX := ORUCap(st)
        end-if
end-do
```

These terms are the flow balance constraints for the dummy start and end nodes

```
RHSBAL(DSNODE,DSPERIOD) := -1
RHSBAL(DENODE,DEPERIOD-1) := 1
DVBAL(DSNODE,DSPERIOD) := -DELTAMAX
ORUBAL(DSNODE,DSPERIOD) := -ORUMAX
```

This term sets the balance constraints for the depot spacecraft and the clients. The flow balance for the depot spacecraft is set to be the capacity of the servicing vehicle chosen. It is a negative because it represents a supply node. The balance for the client satellites is set to be their demand for each commodity.

```
forall( n in nodes, t in periods | t > DSPERIOD and t < DEPERIOD ) do
        if n in depots
        then DVBAL(n,t) := -DELTAMAX; ORUBAL(n,t) := -ORUMAX
        elif n in clients
        then DVBAL(n,t) := demDV(n); ORUBAL(n,t) := demORU(n)
        end-if
end-do
```

This array defines the time it takes to make a move from one node to another. For this instance, every entry in the 26 x 26 matrix is equal to 1.

```
Ttime := [array of Ttime values here]
```

This array defines the delta-V in m/s required to make a move from one node to another.

```
forall (st in stypes)
Txdelta := [array of delta-V costs here, listed in Appendix B]
```

The constraints on the problem are listed here:

CONSTRAINTS

Arcs out of dummy start node go to all other nodes in time period 1

```
forall( st in stypes, s in servicers, nn in nodes ) arc(st,s,DSNODE,nn,DSPERIOD) := 1
```

Arcs into dummy end node go into end period

```
forall( st in stypes, s in servicers, n in nodes, t in periods |
        t+Ttime(n,DENODE) = DEPERIOD ) arc(st,s,n,DENODE,t) := 1
```

Arcs from nodes other than the dummy start and dummy end nodes

```
forall( st in stypes, s in servicers, n in nodes, nn in nodes, t in periods |
        t > 0 and t+Ttime(n,nn) < DEPERIOD and n <> DENODE ) arc(st,s,n,nn,t) := 1
```

Arc parameter equals 0 if delta-V required to make move exceeds servicing vehicle type capacity

```
forall( st in stypes, s in servicers, n in nodes, nn in nodes, t in periods ) do
        if Txdelta(n,nn) > DeltaCap(st)
        then arc(st,s,n,nn,t) := 0
        end-if
end-do
```

W variables are binary, Flow DV and ORU variables are continuous but must be <= servicing vehicle capacities

```
forall( st in stypes, s in servicers, n in nodes, nn in nodes, t in periods ) do
        w(st,s,n,nn,t) is_binary
        DVCAP(st,s,n,nn,t) := flowDV(st,s,n,nn,t) <=
                DeltaCap(st)*arc(st,s,n,nn,t)*w(st,s,n,nn,t)
        ORUCAP(st,s,n,nn,t) := flowORU(st,s,n,nn,t) <=
                ORUCap(st)*arc(st,s,n,nn,t)*w(st,s,n,nn,t)
end-do
```

Node balance constraints

```
forall( st in stypes, s in servicers, n in nodes, t in periods )
NEWBALANCE(st,s,n,t) := sum( nn in nodes, tp in periods | tp + Ttime(nn,n) = t )
        arc(st,s,nn,n,tp)*w(st,s,nn,n,tp) -sum( nn in nodes ) arc(st,s,n,nn,t)*w(st,s,n,nn,t) =
        RHSBAL(n,t)
```

Delta-V left at clients must meet client demands and delta-V required for next move

```
forall( st in stypes, s in servicers, c in clients, t in periods )
DVBALANCE(st,s,c,t) := sum( nn in nodes, tp in periods | tp + Ttime(nn,c) = t )
        arc(st,s,nn,c,tp)*flowDV(st,s,nn,c,tp) -sum( nn in nodes )
        arc(st,s,c,nn,t)*flowDV(st,s,c,nn,t) =      DVBAL(c,t)* sum( nn in nodes | c <> nn )
        arc(st,s,c,nn,t)*w(st,s,c,nn,t) +sum( nn in nodes )
        Txdelta(c,nn)*arc(st,s,c,nn,t)*w(st,s,c,nn,t)
```

Delta-V from depots must be at most servicing vehicle capacity plus delta-V required for next move

```
forall( st in stypes, s in servicers, d in depots, t in periods )
DVBALANCE(st,s,d,t) := sum( nn in nodes, tp in periods | tp + Ttime(nn,d) = t )
        arc(st,s,nn,d,tp)*flowDV(st,s,nn,d,tp) -sum( nn in nodes )
        arc(st,s,d,nn,t)*flowDV(st,s,d,nn,t) >= DVBAL(d,t)* sum( nn in nodes | d <> nn )
        arc(st,s,d,nn,t)*w(st,s,d,nn,t) +sum( nn in nodes )
        Txdelta(d,nn)*arc(st,s,d,nn,t)*w(st,s,d,nn,t)
```

Delta-V from dummy start node must be at most servicing vehicle capacity

```
forall( st in stypes, s in servicers, t in periods )
DVBALANCE(st,s,DSNODE,t) := sum( nn in nodes, tp in periods |
        tp + Ttime(nn,DSNODE) = t and DSNODE <>nn )
        arc(st,s,nn,DSNODE,tp)*flowDV(st,s,nn,DSNODE,tp) - sum( nn in nodes )
        arc(st,s,DSNODE,nn,t)*flowDV(st,s,DSNODE,nn,t) >=
        DVBAL(DSNODE,t)* sum( nn in nodes) arc(st,s,DSNODE,nn,t)*w(st,s,DSNODE,nn,t)
```

ORUs left at clients must meet client demands

```
forall( st in stypes, s in servicers, c in clients, t in periods )
ORUBALANCE(st,s,c,t) := sum( nn in nodes, tp in periods | tp + Ttime(nn,c) = t )
        arc(st,s,nn,c,tp)*flowORU(st,s,nn,c,tp) -sum( nn in nodes )
        arc(st,s,c,nn,t)*flowORU(st,s,c,nn,t) = ORUBAL(c,t)* sum( nn in nodes |
        c <> nn ) arc(st,s,c,nn,t)*w(st,s,c,nn,t)
```

ORUs from depots must be at most servicing vehicle capacity

```
forall( st in stypes, s in servicers, d in depots, t in periods )
ORUBALANCE(st,s,d,t) := sum( nn in nodes, tp in periods | tp + Ttime(nn,d) = t )
        arc(st,s,nn,d,tp)*flowORU(st,s,nn,d,tp) -sum( nn in nodes )
        arc(st,s,d,nn,t)*flowORU(st,s,d,nn,t) >= ORUBAL(d,t)* sum( nn in nodes |
        d <> nn ) arc(st,s,d,nn,t)*w(st,s,d,nn,t)
```

ORUs from dummy start node must be at most servicing vehicle capacity

```
forall( st in stypes, s in servicers, t in periods )
ORUBALANCE(st,s,DSNODE,t) := sum( nn in nodes, tp in periods |
        tp + Ttime(nn,DSNODE) = t and DSNODE <>nn )
        arc(st,s,nn,DSNODE,tp)*flowORU(st,s,nn,DSNODE,tp) -
        sum( nn in nodes ) arc(st,s,DSNODE,nn,t)*flowORU(st,s,DSNODE,nn,t) >=
        ORUBAL(DSNODE,t)* sum( nn in nodes )
        arc(st,s,DSNODE,nn,t)*w(st,s,DSNODE,nn,t)
```

Every client must have a servicing vehicle leave (service the client at exit) within time windows.

```
forall( v in visits, c in clients )
LEAVECLIENT(v,c) := sum( st in stypes, s in servicers, t in periods, n in nodes |
        t >= timeearly(v,c) and t <= timelate(v,c) and n > 0 and n<>c ) arc(st,s,c,n,t)*w(st,s,c,n,t)
        = 1
```

Every client must have a servicing vehicle arrive within time windows

```
forall( v in visits, c in clients )
ENTERCLIENT(v,c) := sum( st in stypes, s in servicers, t in periods, n in nodes |
        t+Ttime(n,c) >= timeearly(v,c) and t+Ttime(n,c) <= timelate(v,c) and n > 0 and n<>c )
        arc(st,s,n,c,t)*w(st,s,n,c,t) = 1
```

OBJECTIVE FUNCTION

```
Cost :=
sum( st in stypes, s in servicers, n in nodes, c in clients, t in periods | n = DSNODE )
        (costwet(st)+.01*s)*arc(st,s,n,c,t)*w(st,s,n,c,t) +

sum( st in stypes, s in servicers, n in nodes, d in depots, t in periods | n = DSNODE )
        (costdry(st)+.01*s)*arc(st,s,n,d,t)*w(st,s,n,d,t) +

sum( st in stypes, s in servicers, n in nodes, nn in nodes, t in periods | n > DSNODE )
        Txdelta(n,nn)*arc(st,s,n,nn,t)*w(st,s,n,nn,t) +

sum( st in stypes, s in servicers, n in nodes, nn in nodes, t in periods | n > DSNODE )
        .1*(NPERIODS-t)*arc(st,s,n,nn,t)*w(st,s,n,nn,t)
```

This is part of the routing for the solution from the first stage run. These terms fix the routes for the first visit while allowing the servicing vehicle type to vary. Appendix E lists the full solution to the first and second stage runs

```
FirstStage(1,0,21,0) := sum(st in stypes) w(st,1,0,21,0)=1;
FirstStage(2,0,22,0) := sum(st in stypes) w(st,2,0,22,0)=1;
FirstStage(3,0,20,0) := sum(st in stypes) w(st,3,0,20,0)=1;
FirstStage(4,0,24,0) := sum(st in stypes) w(st,4,0,24,0)=1;
FirstStage(5,0,19,0) := sum(st in stypes) w(st,5,0,19,0)=1;
FirstStage(6,0,23,0) := sum(st in stypes) w(st,6,0,23,0)=1;
```

Solution set from First Stage run to determine the above constraint
(Complete list available in Appendix E) These are comments to help the modeler write the above constraints and do not affect the model directly.

```
w(1,1,0,21,0)=1;   flowDV(1,1,0,21,0)=0;   flowORU(1,1,0,21,0)=0;
w(1,2,0,22,0)=1;   flowDV(1,2,0,22,0)=0;   flowORU(1,2,0,22,0)=0;
w(1,3,0,20,0)=1;   flowDV(1,3,0,20,0)=0;   flowORU(1,3,0,20,0)=0;
w(1,4,0,24,0)=1;   flowDV(1,4,0,24,0)=0;   flowORU(1,4,0,24,0)=0;
w(1,5,0,19,0)=1;   flowDV(1,5,0,19,0)=0;   flowORU(1,5,0,19,0)=0;
w(1,6,0,23,0)=1;   flowDV(1,6,0,23,0)=0;   flowORU(1,6,0,23,0)=0;
```

This line tells the computer to solve the LP relaxation using the Newton-Barrier method.

```
minimize(XPRS_BAR,Cost)
```

These statements determine the output of the model when the solution is found.

```
forall( st in stypes, s in servicers, n in nodes, nn in nodes, t in periods | getsol(w(st,s,n,nn,t))>0 )
        writeln("w(",st,",",s,",",n,",",nn,",",t,")   flowDV=",getsol(flowDV(st,s,n,nn,t)), "
flowORU=",getsol(flowORU(st,s,n,nn,t)))

end-model
```

Appendix D. Data Used in Model Formulation

The data provided here are derived from results from The Boeing Company’s Orbital Express program *(Proprietary information pending release authorization as of 10 March 2005).*

Appendix E. Solution Outputs Generated

This section lists the output from the model for both the first and second stage runs. The w term is the decision to move servicing vehicle type *st*, number *s*, from node *j* to node *k* at the beginning of period *t*. For example, the term w(1,1,0,20,0) means that servicing vehicle type 1 number 1 will move from node 0 to node 20 at the beginning of period 0. FlowDV equals the meter-seconds worth of delta-V propellant carried along the chosen arc by the associated servicing vehicle, while flowORU is the mass of orbit-replaceable units (ORU) carried.

First stage run solution generated

```
w(1,1,0,20,0)   flowDV=0   flowORU=0
w(1,1,4,20,4)   flowDV=0   flowORU=0
w(1,1,5,6,2)   flowDV=102.5   flowORU=300
w(1,1,6,4,3)   flowDV=50.8   flowORU=150
w(1,1,20,5,1)   flowDV=154.2   flowORU=450
w(1,1,20,25,5)   flowDV=0   flowORU=0

w(1,2,0,23,0)   flowDV=0   flowORU=0
w(1,2,13,15,2)   flowDV=102.5   flowORU=848
w(1,2,14,23,4)   flowDV=0   flowORU=548
w(1,2,15,14,3)   flowDV=50.8   flowORU=698
w(1,2,23,13,1)   flowDV=154.2   flowORU=998
w(1,2,23,25,5)   flowDV=0   flowORU=0

w(1,3,0,24,0)   flowDV=0   flowORU=0
w(1,3,16,24,4)   flowDV=0   flowORU=0
w(1,3,17,18,2)   flowDV=102.5   flowORU=300
w(1,3,18,16,3)   flowDV=50.8   flowORU=150
w(1,3,24,17,1)   flowDV=154.2   flowORU=450
w(1,3,24,25,5)   flowDV=0   flowORU=0

w(1,4,0,22,0)   flowDV=0   flowORU=0
w(1,4,10,22,4)   flowDV=0   flowORU=0
w(1,4,11,12,2)   flowDV=102.5   flowORU=300
w(1,4,12,10,3)   flowDV=50.8   flowORU=150
w(1,4,22,11,1)   flowDV=154.2   flowORU=450
w(1,4,22,25,5)   flowDV=0   flowORU=0
```

w(1,5,0,21,0)   flowDV=0   flowORU=0
w(1,5,7,21,4)   flowDV=173.8   flowORU=0
w(1,5,8,9,2)   flowDV=276.3   flowORU=300
w(1,5,9,7,3)   flowDV=224.6   flowORU=150
w(1,5,21,8,1)   flowDV=328   flowORU=450
w(1,5,21,25,5)   flowDV=0   flowORU=0

w(1,6,0,19,0)   flowDV=0   flowORU=0
w(1,6,1,3,2)   flowDV=102.5   flowORU=300
w(1,6,2,19,4)   flowDV=0   flowORU=0
w(1,6,3,2,3)   flowDV=50.8   flowORU=150
w(1,6,19,1,1)   flowDV=154.2   flowORU=450
w(1,6,19,25,5)   flowDV=0   flowORU=0

The remaining servicing vehicles were not used in the optimal solution. W variables were generated because the decision was made for each of these vehicles to remain at the start node until the final time period, at which time they exited the network. This is equivalent to their not being used. The solution for servicing vehicle type 2 number 2 is listed as an example of the remaining w and flow variable outputs for the first stage solution.

w(2,1,0,0,0)   flowDV=0   flowORU=0
w(2,1,0,0,1)   flowDV=0   flowORU=0
w(2,1,0,0,2)   flowDV=0   flowORU=0
w(2,1,0,0,3)   flowDV=0   flowORU=0
w(2,1,0,0,4)   flowDV=0   flowORU=0
w(2,1,0,25,5)   flowDV=0   flowORU=0

As stated in the full text, the route from the first stage solution is fixed and used as a basis for the second stage solution. The notation is the same as in the first stage solution, though this solution is over 13 periods instead of 6.

Second stage run solution generated

```
w(1,1,0,20,0)   flowDV=0   flowORU=0
w(1,1,4,4,6)   flowDV=153.4   flowORU=450
w(1,1,4,5,7)   flowDV=101.7   flowORU=300
w(1,1,4,20,4)   flowDV=0   flowORU=0
w(1,1,5,6,2)   flowDV=102.5   flowORU=300
w(1,1,5,6,8)   flowDV=50   flowORU=150
w(1,1,6,4,3)   flowDV=50.8   flowORU=150
w(1,1,6,6,9)   flowDV=50   flowORU=150
w(1,1,6,6,10)   flowDV=50   flowORU=150
w(1,1,6,6,11)   flowDV=50   flowORU=150
w(1,1,6,25,12)   flowDV=0   flowORU=0
w(1,1,20,4,5)   flowDV=153.4   flowORU=450
w(1,1,20,5,1)   flowDV=154.2   flowORU=450

w(1,2,0,23,0)   flowDV=0   flowORU=0
w(1,2,13,15,2)   flowDV=102.5   flowORU=300
w(1,2,13,25,12)   flowDV=174.6   flowORU=0
w(1,2,14,15,10)   flowDV=276.3   flowORU=300
w(1,2,14,23,4)   flowDV=0   flowORU=0
w(1,2,15,13,11)   flowDV=224.6   flowORU=150
w(1,2,15,14,3)   flowDV=50.8   flowORU=150
w(1,2,23,13,1)   flowDV=154.2   flowORU=450
w(1,2,23,14,9)   flowDV=328   flowORU=450
w(1,2,23,23,5)   flowDV=0   flowORU=0
w(1,2,23,23,6)   flowDV=0   flowORU=0
w(1,2,23,23,7)   flowDV=0   flowORU=0
w(1,2,23,23,8)   flowDV=0   flowORU=0

w(1,3,0,24,0)   flowDV=0   flowORU=0
w(1,3,16,16,7)   flowDV=153.4   flowORU=998
w(1,3,16,18,8)   flowDV=101.7   flowORU=848
w(1,3,16,24,4)   flowDV=0   flowORU=0
w(1,3,17,17,11)   flowDV=50   flowORU=698
w(1,3,17,18,2)   flowDV=102.5   flowORU=300
w(1,3,17,25,12)   flowDV=0   flowORU=548
w(1,3,18,16,3)   flowDV=50.8   flowORU=150
w(1,3,18,17,10)   flowDV=50   flowORU=698
w(1,3,18,18,9)   flowDV=101.7   flowORU=848
w(1,3,24,16,6)   flowDV=153.4   flowORU=998
w(1,3,24,17,1)   flowDV=154.2   flowORU=450
w(1,3,24,24,5)   flowDV=0   flowORU=0
```

```
w(1,4,0,22,0)   flowDV=0   flowORU=0
w(1,4,10,10,9)   flowDV=101.7   flowORU=300
w(1,4,10,10,10)   flowDV=101.7   flowORU=300
w(1,4,10,12,11)   flowDV=50   flowORU=150
w(1,4,10,22,4)   flowDV=0   flowORU=0
w(1,4,11,10,8)   flowDV=101.7   flowORU=300
w(1,4,11,11,7)   flowDV=153.4   flowORU=450
w(1,4,11,12,2)   flowDV=102.5   flowORU=300
w(1,4,12,10,3)   flowDV=50.8   flowORU=150
w(1,4,12,25,12)   flowDV=0   flowORU=0
w(1,4,22,11,1)   flowDV=154.2   flowORU=450
w(1,4,22,11,6)   flowDV=153.4   flowORU=450
w(1,4,22,22,5)   flowDV=0   flowORU=0

w(1,5,0,21,0)   flowDV=0   flowORU=0
w(1,5,7,7,7)   flowDV=153.4   flowORU=998
w(1,5,7,7,8)   flowDV=153.4   flowORU=998
w(1,5,7,8,9)   flowDV=101.7   flowORU=848
w(1,5,7,21,4)   flowDV=0   flowORU=0
w(1,5,8,9,2)   flowDV=102.5   flowORU=300
w(1,5,8,9,10)   flowDV=50   flowORU=698
w(1,5,9,7,3)   flowDV=50.8   flowORU=150
w(1,5,9,9,11)   flowDV=50   flowORU=698
w(1,5,9,25,12)   flowDV=0   flowORU=548
w(1,5,21,7,6)   flowDV=153.4   flowORU=998
w(1,5,21,8,1)   flowDV=154.2   flowORU=450
w(1,5,21,21,5)   flowDV=0   flowORU=0

w(1,6,0,19,0)   flowDV=0   flowORU=0
w(1,6,1,1,9)   flowDV=276.3   flowORU=848
w(1,6,1,1,10)   flowDV=276.3   flowORU=848
w(1,6,1,3,2)   flowDV=102.5   flowORU=300
w(1,6,1,3,11)   flowDV=224.6   flowORU=698
w(1,6,2,1,8)   flowDV=276.3   flowORU=848
w(1,6,2,2,7)   flowDV=328   flowORU=998
w(1,6,2,19,4)   flowDV=0   flowORU=0
w(1,6,3,2,3)   flowDV=50.8   flowORU=150
w(1,6,3,25,12)   flowDV=174.6   flowORU=548
w(1,6,19,1,1)   flowDV=154.2   flowORU=450
w(1,6,19,2,6)   flowDV=328   flowORU=998
w(1,6,19,19,5)   flowDV=0   flowORU=0
```

As in the first stage solution, servicing vehicle types 2 and 3 were not used, and so solution outputs similar to those found in the first stage were generated for the remaining servicing vehicles.

Bibliography

AeroAstro. "Spacecraft Products." http://www.aeroastro.com/products.html. 10 August 2004.

Air University. "Air University Space Primer." http://space.au.af.mil/primer/. 15 December 2004.

Annaballi, RonJon. *A Multiple Ant Colony Metaheuristic for the Air Refueling Tanker Assignment Problem*. Master's thesis: Air Force Institute of Technology, 2002

Baker, Steven F., David P. Morton, Richard E. Rosenthal and Laura Melody Williams. "Optimizing Military Airlift," *Operations Research* 50(4): 582-602. (2002).

Ballou, Ronald H. "Heuristics: Rules of Thumb for Logistics Decision Making," *Journal of Business Logistics* 10(1): 122-132. (1989).

Boeing. *Orbital Express Servicing and Affordability Analysis* 2002.

Boeing. "Space Shuttle Facts and Figures." http://www.boeing.com/defense-space/space/hsfe_shuttle/facts.html. 11 August 2004.

Chiang, W. C. and R. Russell. *A reactive tabu search metaheuristic for the vehicle routing problem with time windows* 1995.

Columbia Accident Investigation Board (CAIB). *Columbia Accident Investigation Board Report, Volume 1*. Washington D.C.: Government Printing Office**:** 248 2003.

Crino, John R. *A group theoretic tabu search methodology for solving the theater distribution vehicle routing and scheduling problem*. Doctoral dissertation: Air Force Institute of Technology, 2002

Cullenbine, Christopher A. *A Tabu Search Approach to the Weapons Assignment Model*. Master's thesis: Air Force Institute of Technology, 2000

Defense Advanced Research Projects Agency (DARPA). "Orbital Express Space Operations Architecture." http://www.darpa.mil/tto/programs/oe.html. 14 August 2004.

Divinic, N., S. Chappie, A. Arkus and J. Greenberg. *Spacecraft Modular Architecture Design Study: Cost Benefit Analysis of On-Orbit Satellite Servicing*. IAF, 48th International Congress, Turin, Italy 1997.

Dorigo, Marco and Gianni DiCaro. "The Ant Colony Optimization Meta-Heuristic." in *New Ideas in Optimization*. David Cone, Marco Dorio and Fred Glover. London: McGraw-Hill. 1999.

Ebeling, Charles E. *An Introduction to Reliability and Maintainability Engineering*. New York: McGraw-Hill, 1997.

Gambardella, Luca Maria, Eric Taillard and Giovanni Agazzi. "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows." in *New Ideas in Optimization*. David Cone, Marco Dorio and Fred Glover. London: McGraw-Hill. 1999.

Glover, Fred and Manuel Laguna. *Tabu Search*. Boston: Kluwer Academic Press, 1997.

Harder, Robert W. *A Java universal vehicle router in support of routing unmanned aerial vehicles*. Master's thesis: Air Force Institute of Technology, 2000

Hardy, Dave, PhD. *AFSPC On-orbit servicing operational utility study* Air Force Space Command 2003.

Hirzinger, Gerd, Klaus Landzettel, Bernhard Brunner, Max Fischer, Carsten Preusche, Detlef Reintsema, Alin Albu-Schäffer, Günter Schreiber and Bernhard-Michael Steinmetz. "DLR's Robotics Technologies for On Orbit Servicing," *Advanced Robotics* (Special issue on "Service Robots in Space"). (2003).

Hoy, Kirk. Electronic mail to Michael McConnell. 3 September 2004

Isakowitz, Steven J., Joshua B. Hopkins and Joseph P. Hopkins, Jr. *International Reference Guide to Space Launch Systems*. Reston, VA: American Institute of Aeronautics and Astronautics, 2004.

Kobel, Chris. *Delta-V calculations for orbital transfers*. The Aerospace Corp: Microsoft EXCEL spreadsheet, 2004.

Lamassoure, Elisabeth Sylvie and Daniel E. Hastings. *A framework to account for flexibility in modeling the value of on-orbit servicing for space systems*. Master's thesis: Massachusetts Institute of Technology, 2001

Law, Averill M. and W. David Kelton. *Simulation Modeling and Analysis*. Boston: McGraw Hill, 2000.

Leisman, Gregg A. and Adam D. Wallen. *Design and Analysis of On-Orbit Servicing Architectures for the Global Positioning System Constellation*. Master's thesis: Air Force Institute of Technology, 1999

Mandl, Christoph. *Applied Network Optimization*. New York: Academic Press, 1979.

McCarthy, Mitchell J. *United States Marine Corps; (USMC) KC-130J Tanker Replacement Requirements and Cost/Benfit Analysis*. Master's thesis: Naval Postgraduate School, 1999

McCormick, Richard, Dave Hardy and Eric Sundberg. *Joint on-orbit servicing utility study* Defense Advanced Research Projects Agency, Air Force Space Command, National Reconnaissance Office 2003.

Michalewicz, Zbigniew. *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1996.

NASA. "The Hubble Project: Servicing missions." http://hubble.gsfc.nasa.gov/servicing-missions/. 5 September 2004.

NASA. "The ISS to date:." http://spaceflight.nasa.gov/station/isstodate.html. 5 September 2004.

NASA. "Space Shuttle facts and statistics - Extended duration missions." http://www-pao.ksc.nasa.gov/factoids/extnddur.htm. 18 October 2004.

Orbital Recovery Corporation. "About the Orbital Recovery Corporation." http://www.orbitalrecovery.com/about_us.html. 16 August 2004.

Osman, Ibrahim H. and Salhi Said. "Local search strategies for the vehicle fleet mix problem." in *Modern Heuristic Search Methods*. V. J. Rayward-Smith, I. H. Osman, C. R. Reeves and G. D. Smith. New York: John Wiley & Sons Ltd.**:** 131-153. 1996.

Papadimitriou, Christos and Kenneth Steiglitz. *Combinatorial Optimization: algorithms and complexity*. Mineola, NY: Dover Publications, Inc., 1998.

Perez, Ted, Daniel Pires and Gregory Singleton. *Methodology for the economic analysis of on-orbit servicing of satellites*. 2002 IEEE Systems and Information Design Symposium, University of Virginia 2002.

Potter, Seth D. *Orbital Express: leading the way to a new space architecture*. Seal Beach, CA: The Boeing Company 2003.

Rayward-Smith, V. J., I. H. Osman, C. R. Reeves and G. D. Smith, Eds. *Modern Heuristic Search Methods*. New York, John Wiley & Sons, 1996.

Schiffman, David L., Lieutenant, U.S. Navy. *An Underway Replenishment Module for the Battle Group Logistics Coordinator Support System*. Masters Thesis: Naval Postgraduate School, 1993

Smith, David K. *Network Optimisation Practice: a computational guide*. New York: John Wiley & Sons, 1982.

Tamashiro, Hiroshi, Morikazu Nakamura, Shiro Tamaki and Kenji Onaga. *Neighborhood Reduction in Local Search based on Geospatial Relation for Multi Depot Vehicle Routing Problems*. International Technical Conference On Circuits/Systems, Computers and Communications, Phuket, Thailand http://www.kmutt.ac.th/itc2002/CD/pdf/17_07_45/WP2_OB/6.pdf. 2002.

Tirpak, John A. *The Space Based Radar Plan. Air Force Magazine***:** 63-66 2002.

United States General Accounting Office (GAO). Senate Committee on Commerce Science and Transportation and House of Representatives Committee on Science. 31 August 2001

Waltz, Donald M. *On-orbit servicing of space systems*. Malabar, FL: Krieger Pub. Co., 1993.

Wertz, James R., John T. Collins, Simon Dawson, Hans J. Koenigsmann and Curtis W. Potterveld. *Autonomous Constellation Maintenance*. IAF Workshop on Satellite Constellations, Toulouse, France, Microcosm, Inc., 1997.

Wiley, Victor D. *The Aerial Fleet Refueling Problem*. Doctoral dissertation: The University of Texas at Austin, 2001

Vita

Captain Michael L. McConnell earned his Bachelor of Science degree in Environmental Science from North Carolina State University in 2000. After graduating with honors, he was commissioned as a Second Lieutenant through the ROTC program, and entered the Munitions and Missile Maintenance Officer career field. His first assignment was to Francis E. Warren Air Force Base in Cheyenne, Wyoming with the 90th Logistics Group. He attended the Aerospace Basic Course at Maxwell Air Force Base, Alabama in October of 2000, earned the Distinguished Graduate award from the Missile Maintenance Officer Course at Vandenberg Air Force Base, California in February 2001, and completed the Nuclear Munitions Officer Course at Sheppard Air Force Base, Texas in October 2002.

While at F. E. Warren, Lieutenant McConnell served as the Officer-in- Charge of the Team Training and the Minuteman Missile Maintenance Team Sections, gaining experience in both Minuteman III and Peacekeeper ICBM maintenance. Following his promotion to First Lieutenant, he was assigned as Resources Flight Commander.

In 2004 then Lieutenant McConnell attended the Air Force Institute of Technology to pursue a Master's of Logistics Management. While there, he was promoted to Captain. His research efforts concentrated on developing a flexible model to find the optimal on-orbit servicing architecture for the Space-Based Radar system under development by Air Force Space Command and the Defense Advanced Research Projects Agency. Captain McConnell's next assignment is with the 576th Flight Test Squadron at Vandenberg Air Force Base following his March 2005 graduation from AFIT.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 21-03-2005	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* August 2003 – March 2005

4. TITLE AND SUBTITLE AN APPROACH FOR OPTIMIZING THE ON-ORBIT SERVICING ARCHITECTURE FOR A GIVEN CLIENT SATELLITE CONSTELLATION	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) McConnell, Michael L., Captain, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GLM/ENS/05-17
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency Tactical Technology Office 3701 Fairfax Drive Arlington VA 22203-1714	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Satellite systems, once operational, are essentially a consumable item with no capacity to maintain, repair, or upgrade them while on-orbit. In order to avoid having to replace costly space assets, the Defense Advanced Research Projects Agency (DARPA) and Air Force Space Command (AFSPC) are looking to developing programs to provide an on-orbit servicing capability for future satellite systems under development, such as the Space-Based Radar (SBR) system. DARPA and AFSPC are studying on-orbit servicing using the Orbital Express platform as part of an Analysis of Alternatives for the SBR program. Like their satellite clients, on-orbit servicing assets are expected to be resource intensive, and so proper management of these space logistics assets is essential.

This research provides a flexible planning tool to determine the optimal on-orbit servicing architecture for a given client satellite constellation and applies it to the proposed SBR constellation. The model uses a generalized network structure with side constraints to efficiently solve this large combinatorial optimization problem. The optimal number and type of servicing vehicles to use is found, along with the associated most efficient routing to meet client satellite demand for two commodities within multiple time windows.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** STANLEY E. GRIFFIS, Lt Col , USAF VICTOR D. WILEY, Maj, USAF
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	91	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4708 (Stan.Griffis@afit.edu) (937) 255-6565, ext 4284 (Victor.Wiley@afit.edu)