12-2022

A NOVEL APPROACH TO ORBITAL DEBRIS MITIGATION

Timothy S. Turk
University of Tennessee, tturk@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Timothy S. Turk entitled "A NOVEL APPROACH TO ORBITAL DEBRIS MITIGATION." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Dr. James Simonton, Major Professor

We have read this dissertation and recommend its acceptance:

Dr. James Simonton, Dr. Andrew Yu, Dr. Janice Tolk, Dr. Mingzhou Jin, Dr. Trevor Moeller

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

A NOVEL APPROACH TO ORBITAL DEBRIS MITIGATION

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Timothy Scott Turk

December 2022

ACKNOWLEDGEMENTS

I thank my advisor Dr. James Simonton for his continuous support, patience, and mentoring throughout my journey to complete this research. I also thank my entire committee, Dr. Janice Tolk, Dr. Mingzhou Jin, Dr. Andrew Yu and Dr. Trevor Moeller, for their guidance, candid feedback, and mentorship while I worked on this paper over the past years. Additionally, I am grateful to Ms. Charlotte Henley for keeping me in line, making sure I met all deadlines and answering myriad questions and emails. I also thank my parents for setting the example and always supporting me in my quest to be a lifelong learner. I am grateful to God for blessing me with two children I am very proud to call my own. Last, and most importantly, I thank my wonderful wife for her unwavering support, love, and encouragement throughout this effort. I could not have finished this journey had she not been there all the way and for managing our household while I studied and worked on this paper.

ABSTRACT

Since mankind launched the first satellite into orbit in 1957, we have been inadvertently, yet deliberately, creating an environment in space that may ultimately lead to the end of our space exploration. Space debris, more specifically, orbital debris is a growing problem that must be dealt with sooner, rather than later. Several ideas have been developed to address the complex problem of orbital debris mitigation.

This research will investigate the possibility of removing orbital debris from the Low Earth Orbit (LEO) regime by using a metaheuristic algorithm to maximize collection of debris resulting from the February 2009 on-orbit collision of Iridium 33 and Cosmos 2251. This treatment will concentrate on the Iridium debris field for analysis. This research is necessary today, more than ever, as we embark on the launch of thousands of LEO spacecraft, which could result in the realization of the Kessler Syndrome, “The certain risk of failure on launch or during operations due to an on-orbit collision with debris” (Kessler & Cour-Palais, 1978).

TABLE OF CONTENTS

LIST OF TABLES

List of Equations

LIST OF FIGURES

INTRODUCTION AND GENERAL INFORMATION

Space is Vast

In fact, we do not know how incredibly vast our Universe actually is, simply because we cannot see beyond the distance it takes light to travel to us. Thus, this distance is calculated by using our estimation of the age of the Universe and the fact that light travels at approximately 186,000 miles per second, which results in an approximate visible Universe with a diameter estimated at 28 billion light years. However, in the vastness of our known Universe, we, the Earthbound human race, only occupy a very small sliver; indeed, only a small fraction of a light year.

Consider, for a moment, the vast majority of man-made spacecraft are orbiting 35,786km or less above the center of the Earth and it can quickly become evident that ***our*** space is not so vast. The actual volume of free space most of our satellite constellations orbit about the Earth is only:

$$V=\frac{4}{3}\pi r^3 \tag{1}$$

$$V=\frac{4}{3}\pi(35{,}786\text{km})^3-\frac{4}{3}\pi(6{,}371\text{km})^3 \quad (2)$$

$$V=1.92E+14\text{km}^3-1.08E+12\text{km}^3 \quad (3)$$

$$V=1.90E+14\text{km}^3 \quad (4)$$

Now, considering that since Sputnik was launched on April 10th, 1959 and, as of December 31, 2021, there have been 5,655 (Kyle, 2021) successful launches of man-made spacecraft into Earth orbit it should quickly become evident that, relatively speaking, the free space in Earth's orbit, out to the geosynchronous belt, has quickly become quite crowded.

Problem Definition

In fact, there have been concerns about man's ability to continue to exploit our near-Earth orbit since as early as 1978, when Kessler and Cour-Palais completed their ground-breaking work on man's ability to operate in space. In their work, Collision Frequency of Artificial Satellites: The Creation of a Debris Belt (Kessler & Cour-Palais, 1978), Kessler and Cour-Palais predicted there will become a point in time when space becomes too crowded for man to continue to operate freely, and without "certain risk of failure on launch or

during operations due to an on orbit collision." (Kessler & Cour-Palais, 1978) The prediction, now known as the Kessler Syndrome, is nearing reality.

The Chinese Direct Ascent – Anti Satellite (DA-ASAT) missile launch in 2007, a hypervelocity collision of Iridium 33 and Cosmos 2251 satellites over Siberia in 2009 and the 2021 Russian DA-ASAT missile launch contributed over 13,000 pieces of *trackable* space debris. Fortunately, according to Kelso (2022) as of March 23, 2022, only 5,181 pieces of *trackable* debris from these events remain on orbit. Trackable debris are objects with a RADAR cross section of at least 10 centimeters. Debris smaller than 10 centimeters cannot be tracked by the United States Space Force's Space Surveillance Network (SSN). Much of the remaining trackable debris from these three events have since decayed into Earth's atmosphere and are no longer a threat. However, these three events alone contributed more space debris to our near-Earth orbit than missions man has launched since we became a space faring race. Wall (2021) reported that the European Space Agency (ESA) estimated there are more than 36,500 pieces of debris that have a RADAR cross section of over ten

centimeters, millions with a RADAR cross section of one to ten centimeters and over 3,000,000 smaller than one centimeter.

The Russian DA-ASAT explosion on November 15, 2021 forced the International Space Station (ISS) astronauts to enter their emergency escape pods due to risk of collision with debris.

Wall (2021) also noted that since 1999, the ISS has had to make 29 maneuvers to move out of the way of oncoming, potentially damaging, orbital debris. Three such maneuvers occurred in 2021. It should be mentioned that these maneuvers were to avoid orbital debris with a RADAR cross section of ten centimeters or larger, not smaller objects, which are not trackable by the SSN.

The total number of launches, to include 504 failures (Kyle, 2021), since 1957 proves space activities in Earth orbit have become progressively more essential to humankind. Those 504 failures (Kyle 2021) likely contributed to the orbital debris population, too. Operational spacecraft orbiting in the skies above us perform crucial roles. These roles include communications links, position, navigation and timing (PNT) or Global Positioning System (GPS), navigation beacons, scientific investigation platforms, providers of remote

sensing data for weather, climate, land use, and national security purposes. The spacecraft performing these vital tasks are primarily concentrated in a few orbital regimes, including Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Sun-synchronous orbit, and Geosynchronous Earth Orbit (GEO). While the preceding examples of orbital regimes is not inclusive, it does represent the majority of objects on orbit in the skies above. These regimes are important as they are populated with myriad spacecraft performing important missions. These spacecraft must coexist with debris from past missions, spacecraft breakup fragments and natural space debris objects. The satellite operator SES (2017) reported they suffered the loss of their satellite, AMC-9, due to impact by an orbiting piece of debris. The origin of the debris is unknown as it was not tracked by the SSN due to its small RADAR cross section.

Efforts to clean up orbital debris include:

Astroscale Holdings, as noted by Pfeiffer (2021), is a startup company from the United Kingdom (UK) and is developing an orbital debris mitigation capability to help remove large debris objects such as defunct spacecraft and failed rocket bodies. Pfeiffer (2021)

reported that Astroscales concept of operations would be to grapple with such objects using a remote-controlled robotic arm. After attaching to the failed spacecraft, the adjoined system would push toward Earth, with the idea that they would both burn up upon reentry into Earth's atmosphere.

Another mission in development to mitigate the orbital debris problem is called Active Debris Removal by Astroscale-Japan (ADRAS-J). Astroscale (2022) advertised on their website that JAXA, the Japanese equivalent to NASA, chose them to create a mission to push orbital debris into Earth's atmosphere. The concept would use a magnetic probe to pull the two spacecraft toward each other. Once the two spacecraft are attached to one another, Astroscales spacecraft would push the failed satellite or rocket body down into Earth's atmosphere where they would both burn up upon reentry.

According to Astroscale (2021) the company launched an experimental mission to test their ability to use a "servicer" satellite to dock onto a "client" spacecraft. In this experimental mission the client spacecraft would emulate a dead satellite for testing purposes. This capability has already been proven by Northrop Grumman (2020). In

their mission, Northrop Grumman launched a Mission Extension Vehicle (MEV), which was fully fueled, to attach itself to a client satellite. One of these missions involved Intelsat's 10-02 spacecraft. The MEV uses its own thrusters to maintain the orbit of Intelsat's satellite, which extends the life of an otherwise operational satellite, but with little or no fuel onboard.

ESA (2014) publicly introduced the e.DeOrtbit mission, which would launch into a low Earth polar orbit. For this mission ESA considered a variety of apparatuses to capture the debris, including harpoons, tentacles, nets and robotic arms.

Another proposal by ESA (2019) is called ClearSpace-1 and is anticipated to rendezvous with a VEga Secondary Payload Adapter (VESPA) for collection. VESPA is located in the LEO regime and is an excellent target for ClearSpace-1 to attempt to grapple with as it is robust and about the same mass as a small satellite. If successful, ESA (2019) says it will then attempt to capture debris of larger mass and eventually attempt to conduct a multiple object collection.

ELSA-d is the only mission currently on orbit. The mission is testing the concept of capturing debris before it actually attempts to attach itself to a satellite.

These missions are funded by the United Kingdom (UK), ESA and JAXA. Future efforts could be funded jointly, which would be in the best interest of all nations since this is a global issue. The cost of these missions ranges from $100 million to $250 million depending on the duration, complexity of the mission and debris collection spacecraft involved.

ESA's Luisa Innocenti (2019) said, "Even if all space launches were halted tomorrow, projections show that the overall orbital debris population will continue to grow, as collisions between items generate fresh debris in a cascade effect. We need to develop technologies to avoid creating new debris and removing the debris already up there" (ESA, 2019).

What is Space Debris?

Space debris consists of natural space objects, such as asteroids and meteoroids, micrometeoroids, and non-functional man-

made objects. The Committee on Space Debris (1995) identified that debris is generally categorized into various classes; natural objects, solar system probes and objects orbiting the Earth. We are concerned with objects such as operational satellites and orbital debris, which could be dead satellites, rocket bodies or other mission related debris, debris from explosions and satellite collisions or intentionally blowing up dead spacecraft.

There are millions of pieces of space debris in orbit in our Solar system. However, there are myriad debris objects from our exploits into space orbiting the Earth; this is *orbital debris*. In our Solar system, the other, natural space debris orbit the Sun and is more generally referred to as *space debris*. Of the millions of pieces of debris in Earth's orbit, indeed only a fraction of those are tracked by the SSN.

Orbital decay is based on several factors to include altitude, atmospheric density, spacecraft or debris cross section, velocity, gravity and other perturbations. According to analysts, debris from the Iridium 33/Cosmos 2251 collision will remain in orbit for hundreds of years to come. Since decay happens so slowly and considering the

November 15, 2021, Russian DA-ASAT explosion contributed 1,213 pieces of new debris, one can surmise our orbital debris problem is growing, and at an alarming rate.

One might ask, “Why do we care so much about orbital debris?” Simply put, to continue our freedom to launch satellites, safely conduct manned space missions and deploy experimental missions, manned and unmanned, we must understand, monitor and responsibly maintain the space environment around the Earth, especially in LEO.

Although debris varies in size and mass, even the smallest pieces of debris can cause catastrophic damage or loss of life in manned space missions. Debris objects as small as one gram are capable of causing catastrophic damage to orbiting spacecraft. In fact, one of the space shuttle windows suffered a large divot by what was later determined to be a small chip of paint. Had that been a larger piece of debris, it could have shattered the window and killed everyone one board. The following equation for kinetic energy is used:

$$K=\frac{1}{2}mv^2 \quad (5)$$

Thus, a one-gram piece of space debris in LEO traveling at approximately 27,400 kilometers per hour (7.61 meters per second) results in:

$$K=\frac{1}{2}(.001\text{ kg})\left(7{,}600\frac{m}{s}\right)^2 \quad (6)$$

$$K=2.9\text{x}10^4\text{ joules} \quad (7)$$

Now, by definition, “one joule is equal to the energy used to accelerate a body with a mass of one kilogram using one newton of force over a distance of one meter” (Your Dictionary, 2015). Therefore, since one Joule is equivalent to 0.738 foot-pounds of force and $2.9\text{x}10^4$ Joules is equal to $2.1\text{x}10^4$ foot-pounds of force, even a one-gram piece of debris could cause significant damage. Further, if the impacted object is traveling in the same plane, but 180 degrees in the opposite direction, and same speed and mass as the debris, the result is:

$$K=\frac{1}{2}(.001\text{ kg})\left(15{,}200\frac{m}{s}\right)^2 \quad (8)$$

$$K=1.2\text{x}10^5\text{ joules} \quad (9)$$

Since a body of one gram with $1.2x10^5$ joules of kinetic energy is substantial, in fact an order of magnitude greater than equation 7, significant damage could be inflicted on orbiting spacecraft, the ISS or astronauts during extravehicular activity (EVA). The significance of on-orbit damage increases as the mass of the debris increases, and very quickly becomes catastrophic to operational manned and un-manned missions. To put the impact of a collision of this magnitude into context, let's consider a crash between a sports car travelling at 100 km/h and a brick wall.

Suppose the sports car weighs approximately 1450 kilograms (3200 lbs), we will use that mass in our equation:

$$K=\frac{1}{2}(1450\ \text{kg})\left(27.8\frac{\text{m}}{\text{s}}\right)^2 \tag{10}$$

$$K=5.6x10^5\ \text{joules} \tag{11}$$

Now, the sports car example clearly results in a higher kinetic energy, about 4.85 times higher, however, keep in mind the vehicle weighs 1,450,000 times as much as the playing card! This simple comparison, once again, reminds us that even a small chip of paint

travelling at hyper-velocity speeds can cause significant damage to on-orbit spacecraft.

The Committee on Space Debris (1995) reminded us how incredibly important space operations are to mankind. We rely on on-orbit spacecraft for myriad services including satellite communications, the Global Positioning System, scientific missions and national security to name a few. As an example, the United States' GPS, provides, in addition to the means to determine latitude, longitude and altitude, critical timing data to the banking industry, power grids, communications systems and many others, to ensure precise timing synchronization of disparate systems. The loss of GPS timing due to failure caused by orbital debris would be catastrophic to the global economy.

Deorbiting of LEO spacecraft is essential to maintaining a healthy LEO orbital regime. The United States (2020) has recently changed the requirement to deorbit LEO spacecraft from 15 years to five years. This may help to reduce the possibility of collision. With thousands of spacecraft, such as those in the Starlink constellation, being launched into LEO orbit it will quickly become overcrowded and

introduce an irreversible problem, such as the Kessler Syndrome. It is essential that spacecraft be deorbited before they run out of fuel to ensure a successful deorbiting process, else it may take many, possibly thousands, of years to naturally deorbit.

Responsible use of space is required by all nations. Deliberate destruction of on-orbit spacecraft by DA-ASAT's must come to an end if we are to ensure continued use of space in all regimes. Further, development of more exquisite systems to track smaller debris objects would help to protect on-orbit objects such as the ISS and other spacecraft in LEO orbit.

The Committee on the Peaceful Use of Outers Space (2022) noted the importance of mankind's use of near-Earth space, after the Russians used a DA-ASAT missile to explode one of their satellites, Cosmos 1408. That explosion alone contributed more than 1,500 new orbital debris objects into the near-Earth regime and those are only the debris objects that can be tracked by the SSN. It is unknown just how many smaller, untrackable debris objects were created as a result of this one event. Five other debris contributing events occurred in 2021 alone. The other debris producing events include an

unintentional explosion of both an active satellite and an out of service satellite, and destruction of an active satellite by a piece of debris which was large enough to be tracked by the SSN.

Mitigation

If we are to continue manned and unmanned operations in space, orbital debris must be contained, harvested or disposed of now. Several methods and approaches have been considered by space faring nations and organizations, yet only a few have been adopted or tested to date. The current development and launch of massive mega-constellations of satellites in the LEO regime will only further contribute to the realization of the Kessler Syndrome, if orbital debris is not addressed now. Space-X alone plans to launch 12,000 Starlink satellites into LEO. An approach to orbital debris mitigation must be identified, engineered and deployed to ensure freedom of operations in space for generations to come. For the purposes of this research, the Iridium 33 debris field will be analyzed for mitigation. The Iridium 33/COSMOS 2251 collision contributed thousands of pieces of new debris, which need to be removed to ensure safe

operations in space. While there are only 317 trackable Iridium 33 debris objects still on orbit from the collision, there are thousands smaller, untrackable, objects on orbit. Those 317 pieces are cataloged and tracked very closely by the SSN, every day. There is a specific number, the Satellite Catalog Number or SATCAT, assigned to each and every trackable object on orbit. If one were to collide with another object, the SSN data would be used by analysts to determine exactly which objects were involved. An untrackable five-centimeter object would cause significant, likely mission ending, damage to neighboring spacecraft, or even worse, death of an astronaut.

In the case of the Iridium 33/Cosmos 2251 collision, which occurred in 2010, 2,000 pieces of trackable space debris were generated. A hyper-velocity collision such as this would have also generated many thousands of additional pieces of untrackable orbital debris.

Iridium 33 was one of a constellation of 66 satellites. The constellation is in six orbital planes with 11 operational satellites in each plane. The satellites are crosslinked which provides a mesh network for users on the ground. Fortunately for Iridium, the company

also orbits back up satellites and was able to maneuver a replacement into place fairly quickly to maintain service to users on the ground.

According to Weeden (2010), when the collision occurred, the spacecraft were nearly at right angles to each other. The relative speed was estimated to be about 10 kilometers per second. Telescopic investigation after the collision revealed that only the upper portion of Iridium 33 was damaged, which is how the debris field was created. However, had this collision completely destroyed Iridium 33 many more pieces of debris would have been generated.

As of March 23, 2022, the U.S. SSN was tracking 317 pieces of debris from Iridium 33 and 1,025 pieces of debris from with Cosmos 2251, all larger than 10 cm in size.

The United States Strategic Commands (USSTRATCOM) Joint Space Operations Center (JSpOC) was tracking both satellites involved in the collision. The JSpOC reported the close approach of the two satellites to both Iridium and Russian officials. The Russians could do nothing about the possibility of a collision as Cosmos 2251 was an inoperable satellite and could not maneuver to prevent the

collision. Iridium monitored the situation and ultimately chose to not maneuver to avoid collision, which proved to be a costly mistake.

Research Question

What kind of metaheuristic algorithm can be used to develop a multiple factor equation and arrive at a response which optimizes capture of one centimeter or smaller debris objects from the Iridium 33/COSMOS 2251 collision while avoiding larger, mission ending debris objects? Orbital debris must be removed from orbit to prevent realization of the Kessler Syndrome. Although the probability of an on-orbit collision is low, the consequences would be high. Destruction of a $400 million satellite by a one-centimeter piece of debris is a high consequence of not addressing this problem. Worse yet would be the loss of life during a manned mission. As mentioned in the problem statement, there are a few experimental efforts currently underway to capture and remove large debris objects, such as rocket bodies or spacecraft. However, the problem needs a solution for capturing small debris objects that are not currently trackable. Capture of these small debris objects is currently the best method for removal.

General Hypothesis

A metaheuristic algorithm can be defined and used to develop a novel, optimized debris collection plan for effective collection of debris objects from the Iridium 33 debris field. The algorithm would use multiple objectives and constraints to optimize collection of debris objects in the Iridium 33 debris field. The algorithm should optimize the orbit of a collection spacecraft to maximize collection of debris during each pass through the debris field, in the vicinity of the South Pole convergence zone.

Assumptions

Since this treatment is a theoretical approach to orbital debris mitigation, several assumptions need to be mentioned.

a. We must assume some type of spacecraft capable of capturing the debris will be launched and on orbit in the same plane and altitude as the Iridium 33 debris field. This spacecraft could use swarming drones, which employ RADARs to track debris objects smaller than 10 centimeters and a modified Stuffed Whipple shield to capture debris.

b. Drones and the Whipple Shields exist today. Whipple shields are already used in space to protect the ISS from debris objects as large as one centimeter. Six to eight space rated drones tethered to a circular Whipple Shield could be employed to capture debris in the future.
c. Six to eight drones could work together harmoniously without causing damage each other. This was displayed during the opening ceremony at the Tokyo Olympics, where 1,824 drones successfully worked together without damage.
d. Six to eight drones positioned around the modified Whipple shield should be sufficient to conduct operations, identification and collection of debris objects.
e. Drone failure due to various factors could occur during a mission to identify and collect orbital debris.
f. Future work is required to determine the maximum number of drone failures permitted before the hypothetical mission is terminated and reentry is attempted.
g. The size of the shield could be 15 – 20 meters in diameter. McKie (2022) noted the recent launch of the James Webb

Space Telescope proves that large arrays can be successfully folded for launch and deployed on orbit.

h. The drones would be powered by a solar array and batteries protected by the Whipple Shield. A modified Whipple Shield is capable of stopping debris up to one centimeter in diameter. Therefore, the drone swarm would need to steer clear of debris objects larger than one centimeter.

i. Several satellite ground systems must be deployed or leased to communicate with the spacecraft. Although the spacecraft would need to be able to operate autonomously as it captures the debris, some communication would be necessary to conduct telemetry, tracking and commanding (TT&C) operations as the swarm passes over each in-range ground station.

j. The spacecraft would employ a thruster type capable of maneuvering to collect debris, avoid large, potentially damaging debris, and collision avoidance. We want to remove debris, not add to it. Debris larger than one centimeter, which are certainly damaging debris objects,

could be collected by missions mentioned in the problem statement.

k. The spacecraft drone “sensors” should be some type of RADAR that is capable of detecting and measuring the RADAR cross section of debris objects, so as to avoid large, damaging objects. Now, by using the pulse repetition rate (PRR) and frequency of each sensor in the notional swarm of drones, we can use the algorithm to determine the optimal position for maximum debris capture. Each sensor in the swarm will require a unique PRR and frequency, so it knows what return signal to expect when it encounters debris, and a signal is returned to the receiver. The drones will constantly share RADAR return information with the other members of the swarm in order to define an optimized orbit. The RADAR returns will be processed and shared via radio frequency links, perhaps similar to Bluetooth, between the individual drones. This is exploration and exploitation of the search space, as described by swarm-based metaheuristic algorithms.

l. The mission will be capable of collecting at least 4,800 pieces of debris no larger than 1 centimeter in 30 days using at least six drones. This is assuming each orbital period is approximately 90 minutes for 16 orbits per day and collecting 10 pieces of debris per orbit. 10 pieces per orbit was selected to help bound the problem and lend to more productive analysis of the algorithm. However, since debris too small to be detected by the RADARs on the drones may be incidentally collected by the Whipple shield, the actual number of debris collected could be far greater.
m. None of the drones will malfunction during the 30-day mission.
n. None of the drones will cause damage to the Whipple Shield.

Limitations

a. No such spacecraft exists today.
b. Drones are not space capable today.

c. Future work is required to determine the maximum number of drone failures permitted before the mission is terminated and reentry is attempted.
d. This work will not delve into the removal of large debris objects. Other work is already underway related to the removal of large rocket bodies and satellites. This research will mainly focus on the capture and removal of small (1 cm or smaller) debris objects.
e. Complex mission cost models will not be addressed in this work as there are too many variables to consider. Cost modeling may be addressed in future work or as a function of planning a real-world mission employing the concepts laid out in this research.
f. Specific launch vehicles and launch locations will not be addressed in this work. Nor will launch or on orbit failures.

LITERATURE REVIEW

Until recently, perhaps in the last ten to 15 years, there has not been a great deal of work devoted to the actual mitigation of orbital debris. Indeed, there has been a great deal of work developed and published related to the orbital debris environment. However, development of clear plans to mitigate, that is, capture or remove debris by some other means has a limited body of work devoted to it. Nevertheless, more recently, and certainly after the Fengyun DA-ASAT launch, Iridium 33/COSMOS 2251 hypervelocity collision and Cosmos 1408 DA-ASAT, interest in removal of space debris has grown. To that end, there are several more recent bodies of work published concerning the removal or mitigation of orbital debris.

Since John Holland (1975) introduced the Genetic Algorithm (GA) and the publication of Particle Swarm Optimization (PSO) by James Kennedy and Russell Eberhart (1995), a great deal of work has been committed to the study of PSO and GA, and many other heuristic and metaheuristic algorithms.

An extensive literary search has been completed and summarized in the following pages.

“Collision Frequency of Artificial Satellites; The Creation of a Debris Belt” (Kessler & Cour-Palais, 1978)**.**

In their 1978 work, Kessler and Cour-Palais identified the possibility of catastrophic, cascading on-orbit collisions between orbital debris and spacecraft, functional and non-functional, or two non-functional spacecraft which could result in enough debris to eliminate the possibility of using space for generations to come. Clearly, their ground-breaking work was published before deployment of GPS, which provides position and navigation services, but, more importantly, timing services to myriad users worldwide, to include banking institutions globally. Our reliance on GPS alone drives the need to solve this problem.

“Orbital Debris: A Technical Assessment” (Committee on Space Debris, 1995)**.**

In the publication from the Committee on Space Debris (1995), the committee reaffirmed the need to mitigate the near-Earth debris environment and develop plans to mitigate or otherwise remove

orbital debris. The committee also identified the requirement for better modeling of debris fields and the need to improve the ability to track debris with a RADAR cross section of less than ten centimeters in both the LEO and GEO regimes. The committee acknowledged the need for tracking and identifying the source of debris smaller than ten centimeters, which will require significant investment in the SSN. They further identified that information gleaned from these efforts should be compiled for future modeling and reference by the scientific community.

Of particular note, and even more so since the Fengyun DA-ASAT launch, Iridium 33/COSMOS 2251 collision and Cosmos 1408 DA-ASAT, we need to better understand and mitigate the LEO environment. LEO is still the most polluted and populated orbital regime we currently rely upon for space operations, and we must launch through the LEO regime to access all other orbits. There will continue to be a need for widespread use of MEO, GEO and other regimes in the future. GEO communications satellites will continue to enjoy widespread use; O3b and OneWeb will soon launch several more spacecraft into MEO as part of their new constellations.

Enhanced Mitigation

In a publication by the Committee for the Assessment of NASA's Orbital Debris Programs (2011), the committee, which was led by Donald Kessler, PhD, published several findings. These findings include the need for improved methods to remove or mitigate orbital debris in order to minimize the generation of additional debris in an already dangerous environment. The committee also noted that although the need for removal of orbital debris has been identified by NASA and other organizations, such as JAXA and the UK Space Agency, there is much more work to do to address and examine many other considerations. Additional studies in need of examination include diplomatic involvement, monetary impact, conceptual investigation and the international legalities that further complicate an already complex situation.

There is no evidence the committee met more than once after the 2011 session. However, several meetings have occurred to discuss orbital debris mitigation. Reports from many of those meetings can be found on NASA's Orbital Debris Program Office's web page (NASA Orbital Debris Program Office, 2022). These

reports continue to identify the need to remove orbital debris and prevent more on-orbit collisions. They also helped in the decision to conduct this research and, hopefully, bring more attention to the problem of orbital debris and the need for mitigation.

The Satellite Fragmentation Problem

Anz-Meador, et al, (2018) noted that satellite fragmentation has been a serious issue almost since mankind launched the first satellite in 1957. The first time a satellite incident occurred and contributed to the orbital debris environment was in 1961. This one fragmentation event caused the orbital debris environment to grow by 400 percent. One must acknowledge, however, even though that is an enormous growth in debris population, there was little debris identified or cataloged at the time since tracking technology was in its infancy. That said, the event did bring the concern of orbital debris to the attention of satellite developers and scientists. Many more fragmentation events occurred in the 1970's and 1980's and those events served as a call for wider concern in the international space community. Additional unintentional and intentional fragmentation

events since the 1980's have exponentially increased the orbital debris environment. These events highlighted the need to further examine the problem and identify actual, realistic methods for the identification and removal of debris, especially debris too small to be tracked by the SSN.

Also noted by Anz-Meador, et al, (2018) the principal reasons for growth of debris in the near-Earth environment are global launch cadence and fragmentation events. They also reported that debris generated from satellite breakup caused by collisions, explosions and events by debris itself has eclipsed the cataloged population by over 50 percent. Further, about 75% of previously operational spacecraft are inoperable and present a real danger to operational spacecraft.

Near Space Environment

The United States Government Orbital Debris Mitigation Standard Practices (ODMSP) (2019) was commissioned in 2001 to study the concerning increase in near-Earth orbital debris. The ODMSP was charged with the goal of identifying methods to reduce additional contributions of space debris by artifacts of international

space operations. Such artifacts are the result of rocket body and battery explosions, DA-ASAT events and accidental satellite collisions similar to the Iridium 33/Cosmos 2251 on-orbit collision, to name a few. Also, an ODMSP (2019) report provides updates to the initial goals and new objectives for other types of operations in space. In addition to these new updates, the report encourages space faring nations, both in the United States and abroad, to endorse and adopt safe practices for both existing and new space endeavors.

Fundamental of Astrodynamics

Fundamentals of Astrodynamics (Bate, et al, 1971) is a popular handbook for individuals new to the study of space operations as well as seasoned practitioners. It is an excellent reference for both fundamental and more complex approaches to solving difficult space related engineering problems. The equations presented in the book also address such topics as co-orbital operations and approaches to collect or otherwise mitigate the orbital debris environment. It will also enable me to accurately design and predict safe rendezvous operations within the hypervelocity regimes, which are likely to result

in maximum engagement and collection or pulverization of debris encountered.

Moving Target Traveling Salesman Problem

According to Chouby (2013), The Traveling Salesman Problem (TSP) is thought to have first been mentioned in the literature by Hamilton and Kirkman in the 18th century. The TSP is based on the idea that given a number of delivery sites; cities, stores, etcetera, there is an optimal route through those locations which minimizes the cost of travelling to all of them only once and returning to the origin.

Since its first mention in the literature, the TSP has been studied by myriad mathematicians and it has been adopted to solve a wide variety of problems.

In their treatment, C.S. Helvig et al. (2003), proposed a TSP problem which can be employed to solve the traditional TSP, but with moving targets. Choubey (2013) elaborated on the work of Helvig with an application of their work based on the Genetic algorithm.

C.S. Helvig et al. (2003) proposed that a solution to the moving

target TSP could be developed. Their proposals are divided into three different approaches to solving the moving target TSP.

The first approach is limited to one dimension and the number of moving targets must be small, moving at a constant speed and in various directions. The second approach requires the salesman to return to the point of origin after each encounter with a delivery location. It also dictates that all the targets must be moving on a straight line to or from the origin. The third design calls for more than one pursuer and all pursuers must move at the same velocity. Although interesting, none of these methodologies are applicable to this work.

Metaheuristic Algorithms

Meta-heuristic algorithms are more sophisticated heuristic algorithms used to find solutions to myriad problems. They are categorized into four general groups; evolutionary algorithms, human-based algorithms, physics-based algorithms and, finally swarm-based algorithms. Although the genetic algorithm was briefly studied and is mentioned in this research, the remainder of the focus is on swarm-

based algorithms. Swarm-based metaheuristic algorithms are used to solve a wide variety of real-world problems. Some examples include beam welding, truss support design and robotic design.

In their work on PSO, Kennedy and Eberhart (1995) based their algorithm on the unique behavior of swarms of birds and fish.

Optimization

Optimization is so commonplace that people may not even realize they are doing it. From trying to get the best gas in mileage in their automobile (maximization) to buying groceries with the least amount of money (minimization), people optimize. Metaheuristic optimization is used in various applications in industry, computer analysis and mechanical engineering.

Swarm Intelligence

Optimization in the case of swarm intelligence for orbital debris collection should ensure the collection of the largest volume of debris. Swarm-based optimization appears to be the most likely solution to maximize collection of debris. To that end, the following swarm-based algorithms will be analyzed and objectively rated against each other.

This analysis and rating will be used to determine the algorithm with the likeliest chance for adaptation to our space debris collection maximization problem. The Genetic Algorithm was also reviewed for contrast purposes and due to its ties to PSO.

A review of several metaheuristic algorithms follows:

Genetic Algorithm (GA)

Genetic algorithms, first introduced by Holland (1975) are based on the notion of the survival of the fittest (Darwin 1869) in a given population of biological organisms. The basic concept is that in a population there are individuals who are more fit than others. Those individuals will be attracted to other fit individuals to mate. Through the process of fit individuals mating with one another, more fit individuals will be produced as a result of crossover, mutation and selection. Crossover is the process of mixing chromosomes to obtain new offspring. Mutation randomly changes genomes. In some cases, mutation results in increased fitness and in others reduced fitness is realized. Mutation results in the random increase in fitness of some of the population. Those individuals with lower fitness will eventually not

mate and eventually die off. This is the realized through the process of selection. In GA, mutation helps to prevent the algorithm from getting stuck in a local optimum, as the idea of the GA is to find the global optimum. Each generation continues on and on until the end condition is obtained, and the global optimum is attained.

Particle Swarm Optimization (PSO)

In PSO (Kennedy & Eberhart, 1995) the individuals (particles) perform a local search for an optimum solution. Individuals share their search information with other members of the swarm or school, so they can update their position toward a global optimum solution. Iterations of the search continue, and the individuals continue to share their positions until a global optimum solution is discovered.

PSO is an adaptable algorithm with good convergence speed, which reduces the processing and memory load on the computer. It has been found to be very efficient in solving a wide variety of optimizations problems.

Artificial Bee Colony (ABC) Algorithm

ABC is an optimization algorithm, developed by Karaboga, et al (2006), inspired by the searching behavior of bees as they search for food, or nectar. There are three categories of bees; employed bees, onlooker bees and scout bees. Scout bees search for nectar to bring to the colony to which they belong. Once they locate a food source, they become Employed bees. There is only one employed bee at each source of nectar. The location of the nectar is a possible (local) solution for the problem. As each iteration moves forward, some employed bees abandon their food source and become scout bees. Employed bees return to the colony and perform a waggle dance for the onlooker bees. The bee with the best waggle dance persuades the onlooker bees to follow them to the best food source. Meanwhile, scout bees search for other food sources with better nectar than the last food source. If the new location has worse or less nectar than the last location, they return to the last location. As they do, their search continues and they locate better and better food sources, which can be translated into the fitness of a problem. Employed bees and onlooker bees represent the number of solutions to the problem. As

the routine iterates, better and better solutions are developed until a global best solution is located. At this point the iterations cease and the bees remain at the location with the best fitness value.

Ant Colony Optimization (ACO)

ACO was originally introduced by Dorigo and Dicarlo (1992) and is based the pheromone trails left by ants to locate a solution to an optimization problem. During their search process for food, ants leave a pheromone trail on the ground. As the searches continue, ants in the population take differing trails to locate food for the colony. The trails may all lead to the same location for food, but some trails are longer than others. As a result, the ants that take the longer trails deposit fewer pheromones on the trails, as they make fewer trips to and from the food source. Because it takes the ants with the longer trails more time than the ant with the most direct route their pheromones evaporate, and the ant is attracted to other trails. This process iterates until the ants in the colony discover the optimal trail, and this results in a global best solution for the colony.

Imperialist Competitive Algorithm (ICA)

ICA is an evolutionary algorithm proposed by Atashpaz and Gargari (2007) and begins with all the countries of the world, both imperialists and colonies. This is the initial population of ICA. Initially, the colonies are allocated to the imperialists based on the initial strength or power of the imperialist country. The colonies begin to progress toward their respective imperialist. The collective power of the entire imperialist's empire is based on the power of the imperialist country and all of its colonies. As competition between the empires is initiated, stronger empires begin to dominate weaker empires. As this process continues, weaker empires eventually lose enough power to cause them to collapse. As a result of an empires collapse, the empire and its colonies are assimilated into other empires. This process continues until all the weak empires collapse and there is only one empire remaining, which maximizes its overall fitness value.

Firefly Algorithm (FA)

The FA was created by Xin-She Yang (2010). The basic concept of the FA is that fireflies are attracted to each other by the flashing attributes of the light in their tails. Brighter flashing fireflies

attract fireflies with dimmer light in their tails. The brighter the tail of one firefly the more attractive it is to other fireflies. As the range between fireflies grows the brightness of their tails decreases and, as a result, the farther away one firefly is from another firefly the less attracted they are to one another. Alternatively, as the range between two fireflies becomes closer the more attracted the firefly with the dimmer tail becomes to the other fireflies with brighter flashing tails. If the intensity of their tails is the same the fireflies will search for other fireflies and the process will repeat itself. Fireflies can also use their flashing tails as bait to attract their food or as a defensive tool to discourage predators from attacking them.

Monkey Algorithm (MA)

The MA was developed by Zhao and Tang (2008) and describes the behavior of monkeys, which are searching for the highest mountain top in their environment. There may be numerous mountains in their environment, so it may take many attempts to find the highest mountaintop in their search area. Monkeys climb mountains in a search for food. When they get to the top of a

mountain, they look around for other food sources on higher mountains, which is referred to as the watch and jump process. If they see a higher mountain, they somersault off the current mountain and proceed toward the higher mountain. Each mountain the monkey climbs results in a local maximum. The monkeys repeat this process until they do not see any mountain higher than the one they are on now, which results in a global maximum. At this point, the monkeys cease searching for higher mountains in their environment.

Social Spider Optimization (SSO)

The SSO algorithm was developed by Cuevas, et al (2013) as a means for solving complex optimization problems and to avoid convergence at a local solution. The algorithm mimics the behavior of male and female social spiders. There are two types of spiders, solitary spiders and social spiders and they interact quite differently with other spiders. Solitary spiders independently fabricate their own web and rarely interact with other spiders, with the exception being mating. On the other hand, social spiders live in a colony with other spiders and interact with one another. Within the colony there are

both male and female spiders. Female spiders represent approximately 70% of the population in the colony.

Male spiders represent fitness and the colony represents the search space. Male spiders can be dominate or non-dominate and the size typically represents their fitness. Dominate males are considered alpha males and are more likely to mate. Non-dominate males tend to stay in the middle of a group of alpha males and take advantage of their left over prey. Alpha male spiders are typically attracted to female spiders near them for mating purposes.

Grey Wolf Optimization (GWO)

The Grey Wolf Optimizer was developed by Mirjalili, et al (2013) and capitalizes on the behavior and hunting pattern of Grey Wolves. As apex predators, the Grey Wolf has very little competition in their natural environment. Grey Wolves are part of a very structured organization, which is typically 5 – 12 animals and they abide by a stringent social organization within the pack.

The pack has four elements, which includes an alpha male or female, a beta male or female, the deltas and, finally the omegas.

The alpha is the leader of the pack. As leader of the pack, the alpha controls the entire pack and their social behavior, such as hunting, roaming, sleeping, etcetera. The alpha is the only animal in the pack that is allowed to mate with the other wolves, this ensures only his or her dominate genes are propagated within the pack. The other wolves of the pack respect the alpha and, as a couple examples, demonstrate their respect by not attempting to mate with other members of the pack and lowering their tails while in a group with the alpha. Being the alpha does not necessarily mean it has to be the biggest or strongest member of the pack. Nor does it mean the alpha has to be a male; females can be alphas, too. More importantly is the requirement for all members of the pack to follow the lead of the alpha.

Next in line in the “chain of command” is the beta wolf. Similar to the alpha, the beta wolf may be a male or female. As the beta wolf, it is the likeliest to assume the position of the next alpha should the alpha die or become too weak to lead the pack. The beta wolf also provides counsel to the alpha when pack decisions for hunting, gathering and discipline need to be made. The beta wolf must play a

subordinate role to the alpha but enjoys the respect of all other members of the pack.

The next wolf in the pack is the delta wolf, which is subordinate to both the alpha and the beta but is more senior in the pack than the omegas. The deltas are comprised of five categories: hunters, sentinels, scouts, caretakers, and elders. As one might imagine, the hunters are part of the hunting group in the pack along with the alphas and betas. Sentinels are responsible for watching the area around the pack for threats to the pack. Sentinels alert the pack when there is danger and are the defensive group in the pack. Should the need to defend the pack arise, it is the sentinel's job to do so. Caretakers, as the name suggests, care for the older wolves in the pack as well as injured and sick or wounded members of the pack.

The omega is the most subordinate member of the pack and must defer to all other senior members of the pack. They must wait for all other members of the pack to eat before they are permitted to take advantage of the packs kill. Although the lowest ranking members of the pack, omegas play an important role. All senior members of the pack take out their frustrations on the omega.

Without the omega the other members of the pack would fight with each other, which could impact the social order of the pack. The omegas also care for the puppies of the pack.

Antlion Optimization (ALO)

The Antlion Optimizer, a work by Mirjalili, et al (2015), is based on the predation of antlions on insects and their preference is, as their name suggests, ants. Antlions spend most of their roughly three year lives as larvae. It is only in the last three weeks of their lives that they become full adults through the process known as metamorphosis. It is only during their years as larvae that they hunt their prey. Mating occurs in the adulthood phase of their lives. Antlions have a large protruding jaw that looks like pincers. They use this jaw to dig a hole to catch prey, and to capture and eat their prey.

Antlions have a unique hunting method. They live in sandy areas and dig circular, cone shaped holes in the sand, which are intended to trap their prey, typically ants. After they dig their holes, they burrow into the sand at the very bottom of the hole and wait for the unsuspecting ant to wander into it. Detecting the danger, they are

in, the ant attempts to climb out of the hole and escape. When this occurs, the antlion will toss sand at the ant with its jaws in the hopes of causing the ant to slide back down into the hole. If the antlion is successful in causing the ant to slide back into the bottom of the hole it will consume the ant and move its outer exoskeleton out of the hole. After removing the exoskeleton, the antlion repeats the process of grooming its hole in preparation for capturing more prey.

The size of the hole the antlion creates is dependent on two factors, first of which is the phase of the moon. Antlions dig wider holes during a full moon than any other phase of the moon's cycle. Another determining factor for the size of the hole the Antlion digs is driven by how hungry it is. The higher the hunger level is, the larger the hole it digs to capture prey.

Sine Cosine Algorithm (SCA)

The SCA was proposed by Mirjalili (2016) to solve different types of optimization problems. This population-based algorithm is capable of solving a variety of optimization problems. The SCA initializes with a variety of initial stochastic solutions and a

mathematical model of the Sine and Cosine functions is developed, which is intended to find the best outcome.

The exploration and exploitation phases of SCA are achieved by introducing a variety of adaptive and random variables. These steps help to provide for optimization of the search space. Population search and local search help the algorithm and drive the global exploration and local exploitation parts of the search.

Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm (WOA), created by Mirjalili and Lewis (2016), is based on the activity that humpback whales exhibit while hunting their prey. Whales are the largest mammalians on our planet. The blue whale can achieve a length of up to 80 feet long and weigh as much as 330,000 pounds. On the contrary, the humpback whale can be about 50 feet long and weigh as much as 66,000 pounds. Whales are very intelligent mammals and can exhibit a variety of behaviors similar to humans. These behaviors include communication, forming a familial organization (pod), emotion, learning and thinking are examples of a whale’s interesting behavior.

Humpback whales typically hunt in a pod, which increases their success rate in capturing their prey.

The humpback whale hunts by diving below a large group of Krill or other small fish. When the whale is at a certain depth, typically about 36 feet, it starts to swim in a circle and blow bubbles as it rises toward its prey. The intent is to confuse their prey and force them to swim into a ball, which the whale swims up to and captures in its massive mouth. The whale then closes its mouth and forces seawater out through its baleen and captures the krill in its mouth for consumption. Other whales in the pod join on the meal and the process is repeated. These hunting behaviors represent the exploration and exploitation phases of the algorithm, and this hunting method was the main reason the WOA was developed.

Crow Search Algorithm (CSA)

The CSA was crafted by Askarzadeh (2016). American Crows can be found in much of North America but are mostly found in the United States and the lower latitudes of Canada. The intelligence of crows cannot be underestimated. They are believed to be one of the

smartest birds on the planet. Crows have been subjected to tests to evaluate their intelligence. Crows performed very well on such tests, and it has been observed that crows can communicate with one another in very advanced ways. They can also recognize faces and make and use tools.

In the exploration phase of the algorithm a crow searches for food in the search space. Once a crow finds food it either eats it immediately or hides it for future consumption. Crows can be lazy, too. As an example, some crows follow other crows to covertly watch where their food is hidden. Once the crow that hid the food leaves the area, the lazy crow moves in to steal the food, which can be considered the exploitation phase. However, crows are wise and if they observe another crow watching them, they will try to trick the lazy crow and hide the food elsewhere.

As the algorithm is concerned, feasible solutions are locations in the crows local setting and fitness is related to the quality of the food the crow can find and consume or gather for later consumption.

Dragonfly Algorithm (DA)

Dragonflies start their lives out as nymphs and similar to the previously mentioned Antlion, dragonflies spend the majority of their lives in the nymph stage. Following metamorphosis, dragonflies become adults for the remainder of their short lives. Dragonflies are ferocious little insects and will search for and devour most other insects in their environment, both as nymphs in the water and adults buzzing around in their localized hunting area.

As adults, dragonflies fly together in swarms to capture prey and move to different locations as a group. While hunting in a static swarm, dragonflies capitalize on their group behavior to easily capture prey by buzzing around in a confined area. Dragonflies can cover extensive distances during their migratory swarm behavior, which is referred to as dynamic swarming. Static swarms can be considered as the exploitation phase of the DA, while dynamic swarms are considered the exploration phase of the DA.

Dragonflies' unique behavior can be categorized in three of their swarming categories; separation, alignment, and cohesion. During separation dragonflies closely observe other dragonflies in

their small groups to ensure they don't run into each other and prevent inadvertent injuries. While exhibiting the alignment behavior, they adjust their speed to ensure they remain in their small hunting or migrating swarms. During the cohesion phase, the dragonflies maneuver such that they gather in the swarm's main body.

For the sake of comparison, the DA emulates the PSO algorithm.

Grasshopper Optimization Algorithm (GOA)

The GAO, created by Saremi, et al (2017), describes the behavior of grasshoppers and how it relates to metaheuristic algorithms.

Grasshoppers are typically observed as loners but do gather in swarms, also. They are pests that can destroy acres of crops both as nymphs and as adults when they swarm into farm fields or large areas of natural vegetation, such as forests or grasslands. After hatching from an egg pod with up to as many as 300 other mates, grasshoppers begin their lives as nymphs. Nymphs are quite similar to adult grasshoppers, with the major difference being that nymphs

do not have wings. As the grasshopper molts, they become larger and larger, and begin to develop wings on the thorax. Adult grasshoppers can be migratory animals and can cover miles of terrain in large swarms all the while destroying acres of vegetation in their path.

Relevant to this work, the movement grasshoppers can be described as the exploration and exploitation phases of an algorithm. As nymphs the grasshopper moves slowly in a random walk sort of behavior, which can be considered the exploration phase of and algorithm. As adults they swarm into fields and other areas of natural vegetation and exploit the area around the swarm. In the algorithm, the local search continues as local best solutions are arrived at and continue until a best fitness and the global best solution is obtained.

Butterfly Optimization Algorithm (BOA)

BOA was introduced by Arora and Singh (2018). The BOA is based upon the feeding and mating behavior of the common butterfly to solve global optimization problems.

Interestingly, butterflies have the same five senses as humans; touch, taste, sight, smell and hearing. Butterflies use these senses to search for food and sexual partners. A butterfly's sense of smell is its most sensitive and important. Their sense of smell and other senses are used to search for flowers with the most intense fragrance. Fragrances can also emit from butterflies to attract or find a partner. Butterflies search for partners with strong fragrances because it helps them to continue to improve their fitness by maintaining and improving the genetics of their offspring. Fitness also varies as butterflies search for stronger fragrances which can lead a butterfly to sources of better nectar or stronger mating partners. Butterfly fragrances can travel across very long distances, which helps the butterflies to gather and form a strong, informed group of individuals.

As it pertains to this work, butterflies perform searches based on how strong or weak a fragrance is in their surroundings. Local searches are performed when a butterfly does not detect fragrances near them, and a global search is performed when a butterfly senses the strong fragrance of other butterflies. This process is iterated until the global best solution is found.

Harris' Hawks Optimization (HHO)

Heidari, et al (2019) introduced HHO, which is based on the social and hunting behaviors of Harris' Hawks. Harris' Hawks, like Crows, are very smart raptors and can be found in Southern Arizona. Unlike other raptors, Harris' Hawks live in a communal setting and hunt prey, usually rabbits, but occassionally, other small desert dwelling mammals, in a group.

Although they don't exhibit the same social structure as Grey Wolves, they do hunt their prey in a similar manner. Harris' Hawks perch themselves on tall trees, telephone poles or Saguaro cacti and use their powerful eyesight to search for prey. This is the exploration phase of the algorithm. Once prey is located they communicate with each other to share the location and cooperatively fly over the prey to confuse it. As the prey is circled, it runs to try and escape until it is exhausted and remains still. The energy level of the prey correlates its fitness value. It is at this point when one of the hawks dives down to capture the prey. This can be considered the exploitation phase of their hunt. The other members of the hunting party observe the attacker for success or failure. If the initial attacker fails, another hawk

will move in for the kill. Once a successful kill is made, all members of the family share the kill.

The behavior of the Harris' Hawk and its prey during the hunt is used to develop the HHO algorithm, which is pertinent to this work.

Multi-Objective Bat Algorithm (MOBA)

The Bat Algorithm (BA) was developed and published by Dr. Amir Hossein Gandomi and Xin-She Yang (2011). The MOBA was developed by Xin-She Yang (2012). The MOBA is based on the echolocation behavior of bats, which is quite similar to how a RADAR sensor might be employed on individual drones in space.

Bats use echolocation to navigate in their search area while hunting prey, navigating, and for locating and returning to their roost. Bat echolocation is the process of emitting short pules of sound which bounce off prey and other objects in their local search area. Echolocation can also be considered analogous to how a RADAR is used to locate and track objects such as ships or aircraft within its operating range. Of course, RADAR ranges vary from system to system similar to the echolocation behavior of bats. RADARs, among

other means, are used to track objects in orbit around the Earth, to include orbital debris.

The sound emitted by bats varies from bat to bat. Some bats emit a very loud pulse, while other emit pulses with a lower intensity. Bats also decrease the volume they emit at as they get closer to their prey. Some bats emit a frequency modulated (FM) sound and other bats emit a fixed frequency sound pulse. The pulse width of a bats emitted sound is very short, on the order of 8 – 10 milliseconds. The frequency a bat emits also varies depending on the bat but is about 25 to 150 kilohertz (kHz), which correlates to the size of their prey. A bats PRR increases as they approach their prey, which is different than most RADARs. However, some RADARs can modify their PRR, which makes it more difficult to geolocate the origination of its transmitted signal. In the case of the notional RADARs mentioned in this work, their PRR would not necessarily need to be modified. That said, each drones PRR and or frequency would need to be different so it could identify its unique transmitted signal.

Bats echolocate their prey and roost location by calculating the timing difference of the returned signal between their ears. They have

a unique ability locate and track their prey, which is small insects, based on the timing and intensity of the returned pulses. As they detect and close in one their prey the PRR of the bat increases to roughly 200 pulses per second, while the sound level decreases. Bats transmit their pulses in the ultrasonic range, which is beyond the hearing range of humans, which is fortunate since they can emit pulses as strong as 110 decibels. Ultrasonic bursts of sound can be modeled using the following equation.

$$\lambda = v/f \tag{12}$$

Bats can analyze various components of the returned echo to develop a 3D picture of their environment. The variables used to create the 3D picture are time delay of the returned pulse, time difference of the returned pulse between the ears and volume of the returned signal. Bats use their incredible processing ability to process this information and determine the vector and magnitude of their prey's location, the type of insect or other prey and the velocity at which it is moving.

Algorithm Attributes

Desirable Attributes

Adaptable – Since our objective is to continuously map the debris field, we need an algorithm that can easily adapt to the highly dynamic environment of orbital debris in the Iridium 33 debris field.

Scalable – As the drone swarm orbits the Earth, either above or below the debris field, the density of debris will change. As a result, the selected algorithm must be capable of scaling up or down relative to the number of debris it must track and collect.

Robust – For this application, we need a strong algorithm which we are fairly confident will converge at a global optimum, while avoiding local stagnation.

Great balance between exploration and exploitation – In order to maintain a healthy balance between exploration and exploitation, the swarm of drones must efficiently search for new, unidentified debris while remembering the location and orbit of previously located debris for revisiting and collection. This balance will enable the swarm to effectively navigate through the field for maximized collection of debris.

Good convergence speed – In our treatment, we desire the best convergence speed possible. This because good convergence speed in our application should minimize taxing the processors in the swarm of drones. This should enable it to more effectively explore and exploit the debris field throughout its orbit.

Undesirable Attributes

Too easy to fall into local optima – As we explore the debris field during our orbits we do not want to fall into a local optima. Falling into a local optima can prevent us from continuing to explore to discover additional debris This mistake could prevent the drone swarm from locating the global optima, which is the maximum density of the debris field.

Premature convergence – To ensure our swarm of drones does not return a less than optimal result we must ensure we avoid the trap of premature convergence.

Unpredictable results – We need our algorithm to produce result that are predictable. That is, we want to ensure the algorithm properly tracks the debris field so as to efficiently and safely avoid

debris that could cause harm or mission ending damage to the swarm or Whipple shield.

Slow convergence – This mission is highly dynamic and cannot afford to be negatively impacted by slow convergence. The swarm must be able to track the debris and converge on the best global optima as quickly as possible.

Low accuracy – Accuracy must be a top priority of the swarm to ensure it does not inadvertently miss identifying larger, mission ending debris during its orbit and search through the debris field.

Tables 2-1 and 2-2 define the pros (advantages) and cons (disadvantages) related to the attributes selected for analysis. They also cite the sources that enabled the selection of each pro or con. The tables also capture the desirability or undesirability of each pro and con in the matrices.

The decision matrix in Table 2-3 was used to objectively rate each of the algorithms against each other based on the attributes previously defined. In the matrix the most desirable score ranges from five (best) to zero (worst). Alternatively, the least desirable score ranges from zero (least negative) to negative five (most negative).

Table 2-1 Algorithm Pros and Advantages

Pros/Advantages	Citation	Desirability
Adaptable	(Assiri, Hussien, & Amin, 2020) (Toklu & Bekdas, 2014) (Abualigah, Shehab, Alshinwan, Mirjalili, & Elaziz, 2021)	Desirable
Scalable	(Assiri, Hussien, & Amin, 2020) (Beheshti, Mariyam, & Shamsuddin, 2013) (Abualigah, Shehab, Alshinwan, Mirjalili, & Elaziz, 2021)	Desirable
Robust	(Li, Gong, & Gu, 2021) (Beheshti, Mariyam, & Shamsuddin 2013) (Abualigah, Shehab, Alshinwan, Mirjalili, & Elaziz, 2021)	Desirable
Great balance between exploration and exploitation	(Assiri, Hussien, & Amin, 2020) (Li, Gong, & Gu, 2021) (Yang, et al., 2020)	Very Desirable
Avoiding local optima	(Li, Gong, & Gu, 2021) (Beheshti & Shamsuddin, 2013) (Assiri, Hussien, & Amin, 2020) (Abualigah, Shehab, Alshinwan, Mirjalili, & Elaziz, 2021)	Very Desirable
Good convergence speed	(Yang, et al., 2020) (Beheshti, Mariyam, & Shamsuddin, 2013) (Abualigah, Shehab, Alshinwan, Mirjalili, & Elaziz, 2021)	Desirable

Table 2-2 Algorithm Cons and Disadvantages

Cons/Disadvantages	Citation	Desirability
Too easy to fall into a local optima	(Fernandez, 2017) (Li, Du, & Nian, 2014)(Karaboga & Basturk 2006) (Beheshti, Mariyam, & Shamsuddin, 2013) (Yang, et al., 2020) (Li, Gong, & Gu, 2021) (Abualigah, Shehab, Alshinwan, Mirjalili, & Elaziz, 2021)	Very Undesirable
Premature/Low convergence rate in the iterative process	(Li, Du, & Nian, 2014) (Beheshti, Mariyam, & Shamsuddin,2013) (Yang, et al., 2020) (Li, Gong, & Gu, 2021)	Undesirable
Unpredicted results	(Beheshti, Mariyam, & Shamsuddin, 2013)(Li, Gong, & Gu, 2021)	Very Undesirable
Slow convergence	(Li, Gong, & Gu, 2021) (Beheshti, Mariyam, & Shamsuddin, 2013) (Yang, et al., 2020)(Li, Gong, & Gu, 2021)	Undesirable
Low solution accuracy	(Yang, et al., 2020)	Very undesirable

Table 2-3 Algorithm Decision Matrix

	GA	PSO	ABC	ACO	ICA	FA	MA	SSO	GWO	ALO	SCA	WOA	CSA	DA	GOA	BOA	HHO	MOBA
Adaptable	3	2	3	3	2	4	4	4	3	3	3	1	4	3	2	3	4	4
Scalable	2	3	3	3	4	3	4	3	3	3	4	3	4	1	3	0	0	3
Robust	4	4	4	3	2	3	4	4	3	3	2	4	3	2	4	3	3	3
Great balance between exploration and exploitation	3	3	4	3	2	2	3	1	4	0	3	2	4	4	4	3	4	4
Good convergence speed	0	0	0	0	4	0	0	4	4	0	3	4	0	3	4	4	3	3
Too easy to fall into local optima	-4	-3	-3	-2	-1	0	-4	-5	-1	0	-4	0	-3	-2	-3	-1	-3	-2
Premature convergence	-3	-2	0	-3	-2	-4	-1	-1	-3	-5	-1	-1	-2	-2	-2	-1	-2	0
Unpredicted results	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slow convergence	-2	-3	-4	-3	0	-4	-3	0	0	-4	0	0	-3	0	0	2	-1	0
Low accuracy	0	0	-1	0	-2	0	-1	0	0	0	-4	0	-1	-1	0	0		-2
Total	3	4	6	4	9	4	6	10	13	0	6	13	6	8	12	13	8	13

Ultimately, the Multi-Objective Bat Algorithm was selected to work with in this treatment. In addition to having one of the highest scores on the algorithm decision matrix, it most closely matches the characteristics of a RADAR. That is, a bats behavior in locating its prey is quite similar to how a radar locates and determines the range and bearing to its target. The swarming behavior in the other algorithms that were studied do not resemble the transmission and reception characteristics of a radar.

The following detailed examination of Yang's (2012) MOBA provides an example of one metaheuristic algorithm, its rules and pseudocode. The actual computer code may be written in C++, MATLAB, Python or other similar computer languages. If this algorithm was to be used to optimize collection of debris, it could mimic the behavior of microbats' echolocation and use each sensor in the swarming drone to detect orbital debris. The sensors would need to share their "local best solution" with the other sensors to develop a "global best solution." Then, the swarm can maneuver to the optimal orbital position for maximum collection by the modified Stuffed

Whipple shield, while avoiding inadvertent impacts to the swarm's sensors and damaging debris.

Yang (2012) established a basic set of rules for the Bat Algorithm.

1. Bats can discern their environment using their unique echolocation ability. They use that ability to avoid obstacles, detect range, search for and capture their quarry and return to their roost.
2. While hunting their prey, trying to locate their roost or avoiding obstacles, bats fly in random searches until they home in on their target. While they are searching, they randomly adjust their speed at a given position. As they approach the target, they modulate the frequency and adjust the PRR of their emissions. This continues as they close in on and capture or avoid their intended objective.
3. The amplitude or loudness of the bat's emissions varies from loud to very low as they get closer to their prey, roost or obstacle.

From his paper, Yang (2011) provides the following definitions and formulas for the MOBA.

Rules for MOBA must be defined relative to their current position x_i and current velocity v_i in a d-dimensional environment. The purpose for defining these rules is to be able to update a new position and velocity from a current solution. After the current positions are updated the new positions and velocities at the next time step are calculated. The updated position x_i^t and velocities v_i^t at time step t are defined by the following:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (13)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*)f_i \quad (14)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (15)$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. In this example, the updated global best solution is x_*. This solution is arrived at by evaluating the best solution of all bats each time the process is repeated. The velocity is given as $\lambda_i f_i$, which represents wavelength and frequency,

respectively. At each timestep the velocity update can be computed by adjusting either the frequency or wavelength, while keeping the other component of the equation fixed. In Yang's (2012) example, f_{min} = 0 and f_{max} = $0(1)$ was used and is dependent on the size of the domain in the problem being analyzed. To initialize the problem, the bats are given a random frequency. These frequencies are attained in a uniform manner from $[f_{min}, f_{max}]$. When a solution is obtained during the local search step, the bats are assigned a new solution by means of a random walk as such:

$$x_{new} = x_{old} + \in A^t, \tag{16}$$

In this example $\in$ is a random number vector drawn from [-1, 1]. $A^t=<A_i^t>$ is drawn from averaging the amplitude of each bat during this step. The step change to the bats position and velocity resembles the PSO, because the initial frequency, f_i, drives the speed and distance of the moving particle swarm.

Likewise, the drone swarm has multiple elements to consider. The sensors will sense range (distance), the X, Y, and Z coordinates and velocity of each piece of debris in the field within an established area in view of the drone swarm; all of these factors will also help to establish our constraints. Further, in order to maximize harvest during each pass, the model must account for maximum density of debris as it calculates and arrives at a global best solution.

Each drone element will utilize its own organic sensor (RADAR transceiver), to transmit and receive its own pulse's, to resolve a local best solution. As local best solutions are identified by each of the drones in the swarm, they should communicate with the other drones to determine a global best solution. This solution will be used to maneuver the drone swarm to the optimal vector and harvest the maximum number of debris objects during each pass. The solutions must be arrived at by utilizing data collected at the anticipated collection point, a sphere if you will. In this case, mapping and harvest should occur near the previously defined Southern pole where the debris cloud converges, as this will allow for maximum engagement with the debris cloud. This concept also minimizes fuel

spent during harvest of debris as “in-plane” velocity changes are much less expensive, fuel-wise, than out of plane changes.

In contrast to the traditional MOBA, our sensors will be employed different from the way a bat uses its capabilities. Since the drones will be dumb as compared to a bat, pulse rates and loudness (amplitude) will remain fixed values for each drone This will assist each drone in deciphering its signal from adjacent drone transmissions. Each drone will receive echo’s or “returns” from debris in its field of regard, a sphere defined by the range, X, Y, Z coordinates, and derived map of the best local solution. This will be a slow, methodical process. In the case of the Iridium 33 the swarm will only pass the North or South convergence zone about every 45 minutes. This allows the swarm to constantly collect, calculate and measure for maximum harvesting at each pole, as desired. We have chosen to concentrate on the South pole convergence zone to afford the “system” to further refine a global best solution between each pass.

The orbit of each object would naturally change slightly over time due to atmospheric drag, gravity and other perturbating factors,

which may cause the debris to eventually, over many years, decay into Earth's atmosphere. These orbital changes will not occur at the same rate for each piece of debris, as they are different mass, shape and cross section. The algorithm should optimize the orbit of the swarming drone spacecraft to maximize collection of debris during each pass through the highest density part of the debris field. Since the debris field will change so slightly over time, the system will be able to map and save global best solutions throughout the harvesting process. This approach may ultimately be used to update previous global best solutions as current local best solutions, along with freshly attained local best solutions, and arrive at new global best solutions more efficiently.

Optimality fronts, Yang (2012), are found or estimated in multi-objective problems because they are complex, as compared to single objective problems. In this design space, range and volume of the sphere of interest could be constraints and contribute to the feasible region, which will ultimately help to arrive at an optimality front for the problem.

There will be a variety of modeling and simulation tools used to calculate and model the validity of the algorithm. These tools include:

1. Systems Tool Kit (STK) by Ansys Government Initiatives (AGI). STK is widely used, in addition to other domains, in the commercial and government space community to model and simulate various activities including collision avoidance, debris dispersion following collisions, satellite communications link budget calculations, etc. I plan to use STK to provide a visual model of the drone swarm interaction with the Iridium 33/COSMOS 2251 debris field and help determine the location of the highest density location of the debris field.
2. MATLAB – MATLAB is a powerful computational tool used to solve a wide variety of complex mathematical problems. It can be used to create Pareto optimal fronts, 2-D and 3-D graphs, surface plots and offers the ability to plot objects, such as debris, on a 3-D globe. These capabilities can be used to interpret results of calculations and were decision factors used to select the software to work with on this

problem set. MATLAB was chosen over other programming software, such as Python or C++ because of the capabilities listed above. I plan to use MATLAB as a tool to locate the highest concentration of the debris field and help develop a fitness model for the MOBA.

3. Lingo 17.0 – LINGO is a robust computational tool used to efficiently solve a wide variety of mathematical problems and is a perfect tool to use to solve optimization problems. I plan to use LINGO to develop an optimized function to maximize collection of orbital debris by the drone swarm and will use the results for comparison with the MOBA.
4. Qlik – Qlik is a predictive analysis software tool that enables the user to enter historical data for analysis to predict future requirements for such things as consumer demand and developing artificial intelligence and machine learning solutions. I will use Qlik in an attempt to determine if the Travelling Salesman Problem can be used to solve the problem of collecting debris in a varied and dynamic environment.

MATERIALS AND METHODS

This treatment will describe a *theoretical approach* to orbital debris mitigation, which employs commercially available technologies, albeit with modifications required for space flight operations and safety. It will be shown that orbital debris harvesting, and disposal, is not only possible, but the time has come for all space faring nations to seriously consider experimental harvesting or disposal operations. It will be shown there are technologies and materials available to safely conduct harvesting or disposal operations in the LEO regime, the orbital regime with the most pressing orbital debris environment. Specific research question supporting this effort is as follows:

An operational approach to maximize co-orbital collection efficiency must be addressed. That is, what is the optimal orbital plane and altitude to encounter the highest concentration of orbital debris from the Iridium 33 debris field? A metaheuristic algorithm will be used to ensure we maximize co-orbital encounters for harvesting and disposal of orbital debris in the Iridium 33 debris field.

Debris fields from on-orbit hypervelocity collisions disperse at extreme velocities. As a result, the debris field does not remain in a uniform orbit, rather, it disperses outward from the point of initial impact. Eventually, the debris cloud becomes widely dispersed in its orbit. However, the debris cloud converges again at the point of initial impact.

It is appropriate to provide a brief description of orbital mechanics, to describe what kind of velocities and maneuver impacts there will be relative to fuel costs. Figure 3-1 depicts the orbital elements of a satellite orbiting a central body. Of course, in our case the central body is the Earth. We will use the six classical orbital elements, which are described below.

These values can be found in the following two-line element (TLE) set published by Celestrak (celestrak.org, 2022) and looks like the following example, which are colored to match the corresponding parts in Figure 3-1.

IRIDIUM 33

1 24946U 97051C 20096.76365512 .00000084 00000-0 23079-4
2 24946 86.3925 218.2787 0009311 137.9193 222.2719 14.336791

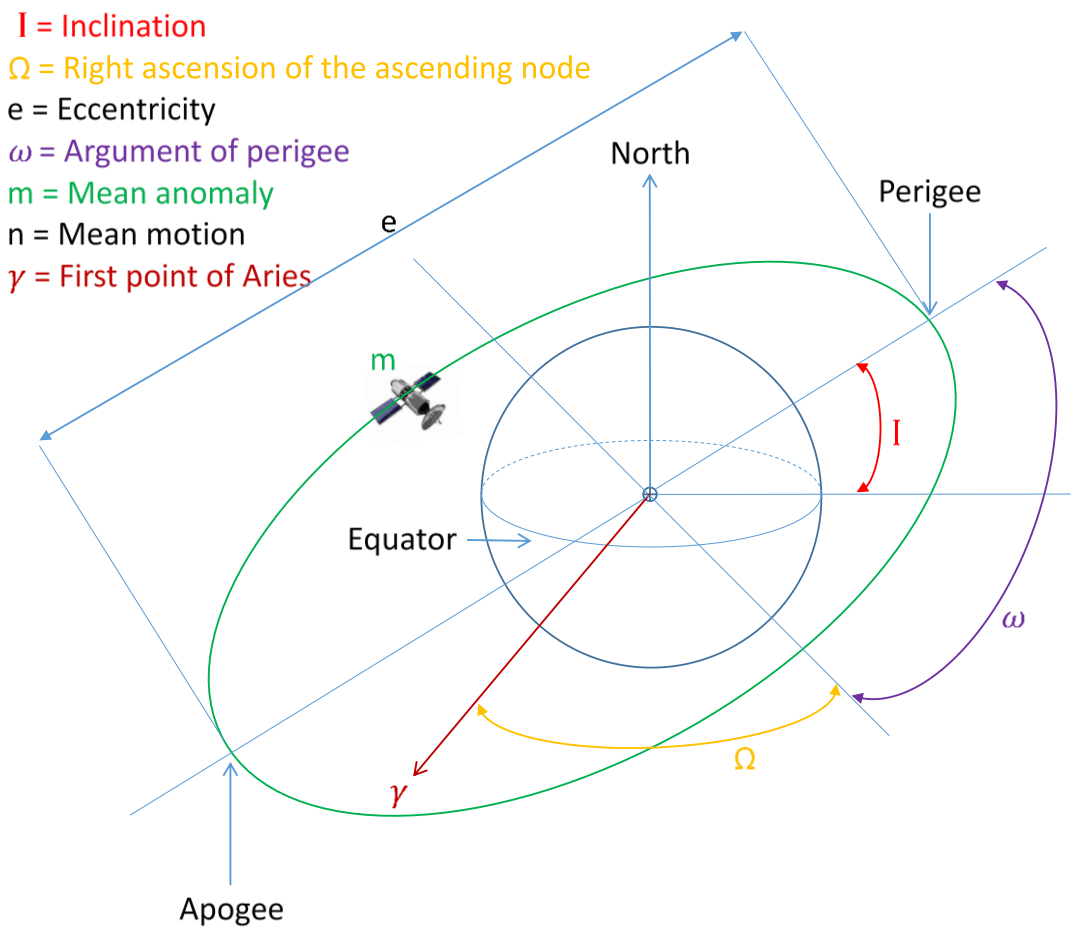

Figure 3-1 Orbital Elements

Apoapsis or Apogee - The point in a satellites orbit where it is farthest from the main body's center point; Orbital mechanics reduces everything to a point mass.

Periapsis or Perigee - The point in a satellites orbit where it is closest to the main body's center point.

Inclination – This is the angle between the main body's ecliptic plane (equator) and the satellites orbital plane. That is, the angle represented by the red arrow in figure 3-1, which is the angle between the green orbital plan and the equator.

Right Ascension of the Ascending Node (RAAN) - Defines the longitude in an Earth referenced position the satellite passes over the ecliptic through its orbit. RAAN is measured counterclockwise from the first point of Aries (vernal equinox) and calculating it will provide an absolute value of the satellite's longitude at any point in time.

Eccentricity - Describes the shape of the orbit. An orbit with zero eccentricity is a perfectly round orbit and as eccentricity increases, the orbit becomes more oblong. Further, the semi-major and semi-minor axes are equal in a circular orbit and diverge from one another as the orbit becomes more eccentric.

Argument of Perigee - The angle from RAAN to the semi-major axis and describes the rotation of the ellipse.

Mean Anomaly - The angle measured from perigee to the location of the satellite is in its orbit.

Mean Motion - The number of times a satellite orbits the Earth in one solar day, approximately 24 hours. Knowing these values, we can determine velocity and change in velocity or delta V (dV) required to make maneuvers.

The following equations will be used to determine velocity.

First, we must determine the average of the apogee, r_1, and perigee, r_2; a.

$$a=\left(\frac{r_1+r_2}{2}\right) \tag{13}$$

In a circular orbit r_1 and r_2 are equal and since we are using point mass, we must include the radius of the Earth, which is 6,371 km. The following calculation is for demonstration purposes only to describe the process for a non-circular orbit. Clearly, we could come to the same conclusion by adding the Earth's radius and height of Iridium's orbit to

arrive at the same conclusion. Since we are studying Iridium 33, we'll use its nominal operational altitude, which is 781 km.

Therefore, we have:

$$7{,}152 \text{ km} = \left(\frac{(6{,}371 \text{ km} + 781 \text{ km}) + (6{,}371 \text{ km} + 781 \text{ km})}{2} \right) \quad (14)$$

Now, to find the velocity of a satellite at any point in time, we will use the following formula:

$$V^2 = GM\left(\frac{2}{r} - \frac{1}{a}\right) \quad (15)$$

Where G is the universal constant of gravitation and M is the mass of the central object, in this case, Earth.

$$G = 6.674 \times 10^{-11} \quad (16)$$

$$M = 5.972 \times 10^{24} \quad (17)$$

$$GM = \mu = 6.674 \times 10^{-11} \times 5.972 \times 10^{24} = 3.986 \times 10^{14} \quad (18)$$

$$V = \sqrt{3.986 \times 10^{14} \left(\frac{2}{7{,}152 \text{ km}} - \frac{1}{7{,}152 \text{ km}} \right)} = 7{,}465 \text{ m/s} \quad (19)$$

Again, the previous calculation is for demonstration purposes only to describe the process for a non-circular orbit. Clearly, since r and a are equal, we can come to the same conclusion by reducing the equation as follows:

$$V=\sqrt{(\mu\left(\frac{1}{a}\right))} \tag{20}$$

$$V=\sqrt{3.986 \text{ x } 10^{14}\left(\frac{1}{7{,}152 \text{ km}}\right)}=7{,}465 \text{ m/s} \tag{21}$$

Now, a simple dV, with no inclination change, is simply raising or lowering the altitude of the spacecraft. The following formula enables us to calculate the dV required for an orbit raising maneuver from 7,152 km to 7,175 km a difference of 23 km.

Therefore, we have:

$$7{,}164 \text{ km}=\left(\frac{(7{,}175 \text{ km})+(7{,}152 \text{ km})}{2}\right) \tag{22}$$

$$V=\sqrt{3.986 \text{ x } 10^{14}\left(\frac{2}{7{,}152 \text{ km}}-\frac{1}{7{,}164 \text{ km}}\right)}=7{,}459 \text{ m/s} \tag{23}$$

So, to conduct an orbit raising maneuver of 23 km we must reduce our velocity 6 m/s, which requires very little energy.

There are currently 317 trackable pieces of debris from Iridium 33 on orbit and 1,025 pieces of debris from COSMOS 2251. Although the debris cloud disperses after initial impact Figure 3-2 demonstrates the debris cloud converges at the North and South poles, which would provide the optimal location to begin the search for maximized collection.

Specific goals and objectives are:

a. Identify, using a metaheuristic algorithm, based on a specific epoch, the most desirable orbit for maximized capture of orbital debris in the LEO regime.
b. Safely pass through the debris field while minimizing the possibility of impacting an object that could cause mission failure.
c. Identify a metaheuristic algorithm that can be utilized to optimize the orbit of the drone swarm. Optimization will be

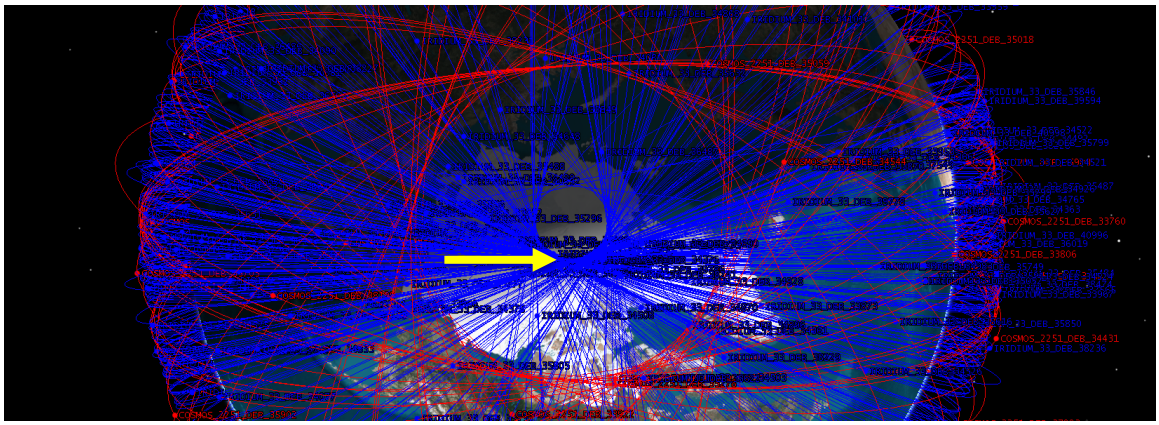

Figure 3-2 South Pole Debris Convergence Zone

based on inputs from the drone’s sensors, similar to the way the members of a swarm function in particle swarms.

d. Collect 4,800 pieces of debris during a 30-day mission. This is based on a 90-minute orbit from the South pole convergence zone back to the point of origin and the goal of collecting 10 pieces of debris during each orbit. The debris collected shall be smaller than one centimeter RADAR cross section. By limiting the goal of collection to 10 pieces per day, it will help to bound the problem and may help to build upon the MOBA for better analysis.

RESULTS AND DISCUSSION

Debris fields from on orbit hypervelocity collisions disperse at extreme velocities. As a result, the debris field does not remain in a uniform orbit, rather, it disperses outward from the point of initial impact. Eventually, the debris cloud becomes widely dispersed in its orbit. However, the debris cloud converges again at the point of initial impact.

As of March 23, 2022, there are 317 pieces of trackable debris from Iridium 33 on orbit and 1,141 pieces of debris from COSMOS 2251. Although the debris cloud disperses after initial impact Figure 4-1 demonstrates the debris cloud converges at the South pole, which would provide the optimal location to begin the search for maximized collection. As mentioned in the Materials and Methods chapter, section d of the specific goals and objectives, the problem has been bound to 10 pieces of debris per pass. By bounding the problem, we may be able to conduct better analysis. If we are successful in analyzing the problem and coming to a solution with MOBA, we can then scale the problem to a larger data set for

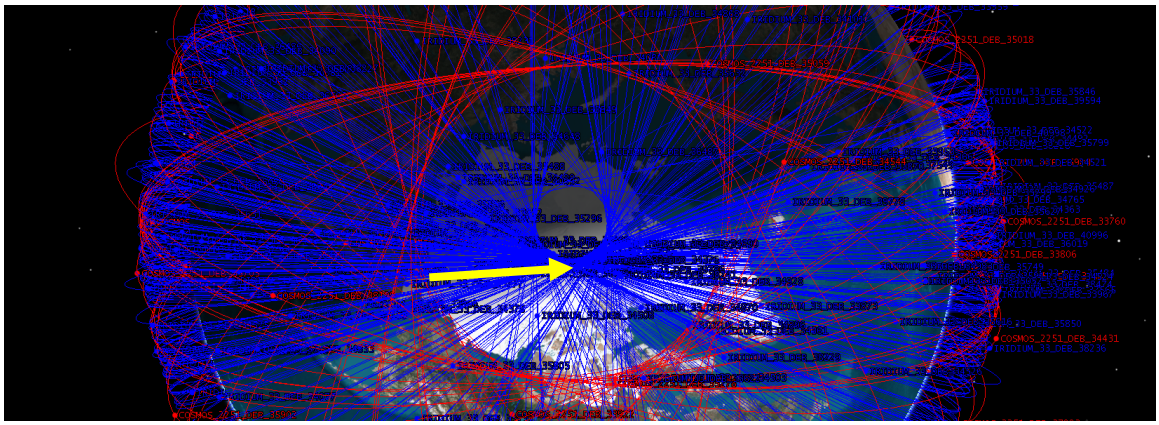

Figure 4-1 South Pole Debris Convergence Zone

analysis and possible collection. One must recall that there are millions of pieces of debris smaller than one centimeter. As such, even though the goal is to limit analysis and collection to 10 pieces per pass, incidental collection of smaller debris objects is likely to occur. In the event more than ten pieces are collected per pass, and provided no serious damage is done to the Whipple shield, the pass shall be considered successful. Our goal is to collect debris and help to create safe environment for future space operations.

Mr. Yang's MOBA paper (2012) provides the foundation to build upon for the modified algorithm. The following decision variables, constraints and equations define the modified MOBA to be used for the analysis of this problem and will ultimately result in a global optima for debris collection. The following decision variables and constraints must be met to maximize the collection of debris objects during our 30-day mission.

Decision variables include:

1. Collect at least 4,800 pieces of debris during the 30-day mission.

2. Objective of swarming drones is to collect at least 10 pieces of debris during each pass through the South pole convergence zone.

Constraints include:

1. RADAR cross section of debris collected must be smaller than one centimeter.
2. Drone swarm must maneuver to avoid debris larger than one centimeter to avoid damage to the Whipple shield.
3. In order to conserve fuel, the drone swarm may not perform out of plane maneuvers, unless it is required to do so to avoid debris larger than one centimeter.
4. Drone swarm must end mission when remaining fuel level equals that required to deorbit spacecraft, regardless of number of debris collected.
5. Swarming drones must be able to autonomously end mission if communication with ground station(s) is lost during two or more orbits in one day.
6. Each ground station must be able to conduct TT&C of the spacecraft when it is within direct line of sight.

For the drones in simulations, we have to define the rules for how their positions x_i and velocities v_i in a 3-dimensional search space are updated. The new solutions x_i^t and velocities v_i^t at time step t are given by

$$v_i^{t+1}=v_i^t+(x_i^t-x_*)f_i \quad (24)$$

$$x_i^{t+1}=x_i^t+v_i^t \quad (25)$$

where $f_i = 2.45\text{GHz} + f, \text{with } f \in [10,20]\text{MHz}$. 2.45GHz was selected as f_i because it is the frequency used for Bluetooth communications. 2.45GHz will be mixed with the randomly generated frequency between 10MHz and 20MHz. Here x_* is the current global best location (solution) which is located after comparing all the local best solutions among all the 6 drones at each iteration t. The global best solution is the location in the debris field within the range of the RADARs, which has the highest concentration of debris. As the product $\lambda_i f_i$ is the velocity increment, where λ is the wavelength and f is frequency. As such, we can only use λ_i to adjust the velocity change while the other factor f_i is not variable. This is because the frequency of each drone will be fixed. However, this is only applicable

in a static or initialization condition. One must consider the fact that the Doppler effect will cause the relative λ to decrease and the relative f to increase as the swarm approaches the debris objects. Since performing a dV of the spacecraft is relatively slow using Xenon Ion Propulsion System (XIPS) or similar drives, and expensive, the product $\lambda_i f_i$ should not vary significantly. For the local search part, once a solution is selected among the current best (local) solutions, a new solution for each drone is generated locally using random walk

$$x_{new} = x_{old} + \in A^t, \tag{26}$$

where $\in$ is a random number vector drawn from $[-1, 1]$, while $A^t = < A_i^t >$ is the average transmit power of all six drones at this time step. The update of the velocities and positions of the drones have some similarity to the procedure in the standard particle swarm optimization, as f_i essentially controls the pace and range of the movement of the swarming drones. The following modified algorithm pseudocode will provide the basis from which we will develop our algorithm.

Modified Bat Algorithm
Objective function f (x), $x = (x_1, ..., x_d)^t$

Initialize the drone population
Initialize the drone position and velocity xi and vi
Define pulse frequency f_i at x_i
Initialize pulse rates r_i and the transmit power A_i
while (*t < Max number of iterations*)
Generate new solutions by adjusting wavelength,
and updating velocities and locations/solutions
if (*rand <r_i*)
Select a solution among the best solutions
Generate a local solution around the selected best solution
end if
Generate a new solution by flying randomly
if *(rand < A_i & $f(x_i) < f(x^*)$)*
Accept the new solutions
Increase r_i and reduce A_i
end if
*Rank the drones and find the current best x**
end while
postprocess results and visualization

The drones must constantly communicate with other members of the swarm to ensure they are using independent frequencies and PRR's and prevent interference amongst the swarm. Additionally, each of the six to eight drones shall have the ability to determine the RADAR cross section of the debris objects. It will be of critical importance to avoid, at all costs, debris large enough to damage drones or cause catastrophic, mission ending impact to the modified Whipple Shield. Debris will be collected by the Whipple Shield as it

passes through the debris cloud by conducting dV's to raise and lower the orbit.

Using modeling of the debris fields with MATLAB, we can attempt to initiate our search in an area with the highest density of debris. This will help to determine the optimal launch window to insert the spacecraft into orbit, which will save precious spacecraft fuel. STK modeling enables us to see the debris and ground track for each piece of debris in a specific epoch. Each new epoch provides a new TLE from which the latitude, longitude and altitude is derived for each piece of debris in the debris field. Each new 90 minute epoch also provides a new ground track representation. Figure 4-1 depicts, graphically, the South pole debris convergence zone, where we should be able to locate the highest density of debris during a full orbit. We can tell, by observing Figure 4-1, the highest density of debris will occur at the debris convergence zone, but when will that actually occur? As the Iridium 33 debris orbits the Earth it converges at the poles and diverges as it departs the poles in its orbit. Further, since the debris is at varying altitudes the debris is moving at different velocities, therefore with each additional orbit the maximum

convergence changes at the poles. That is, rarely will the same two (or more) pieces of debris converge at the same time at the debris convergence zone. This is the result of multiple perturbations as the debris orbits the Earth and results in different orbits, specifically orbital velocities, which causes the density of the debris field to change over time. As a result, with each passing orbit it becomes obvious to conclude it is nearly impossible to determine the maximum density of debris due to its dynamic nature. Even if the debris were organized into spheres in an effort to categorize them, the number of objects in each sphere would change with each passing orbit as objects enter and leave the spheres. Organization of debris in such a manner simply creates a smaller number of debris objects as they are clumped together in the larger spheres, however, the problem of identifying the highest concentration of debris objects remains the same. MATLAB was used to try and determine the velocity of the objects in the debris field, trajectory of the debris and dispersion of the debris as it orbits the Earth. Although STK does model the debris field, MATLAB might have been better to manipulate the outcome of the analysis and enable the discovery of the best orbit to launch the

swarming drone into for maximum debris mapping. MATLAB was also used in an attempt to use the modified MOBA to obtain a Pareto optimal front and surface plot of the local and global best solutions. As it turned out, there was no way to use the Two-Line Element data in the MATLAB MOBA algorithm, as MOBA uses random numbers to obtain its solutions. STK TLE data was also culled down into latitude, longitude and altitude data for analysis in MATLAB. This data only represents a momentary snapshot in time and does not truly represent the dynamic nature of the debris field as it orbits the Earth. The latitude, longitude and altitude data obtained from STK was successfully used to create a 2-Dimensional and 3-Dimensional representation of the debris field in MATLAB. Figures 4-2 and 4-3 depict both a MATLAB 2-Dimensional and 3-Dimensional image of scattered nature of the debris objects in the field in a snapshot of time. These snapshots depict the position of each of the Iridium 33 debris objects imported from STK and plotted with MATLAB. One could repeatedly import the Iridium 33 debris objects into MATLAB

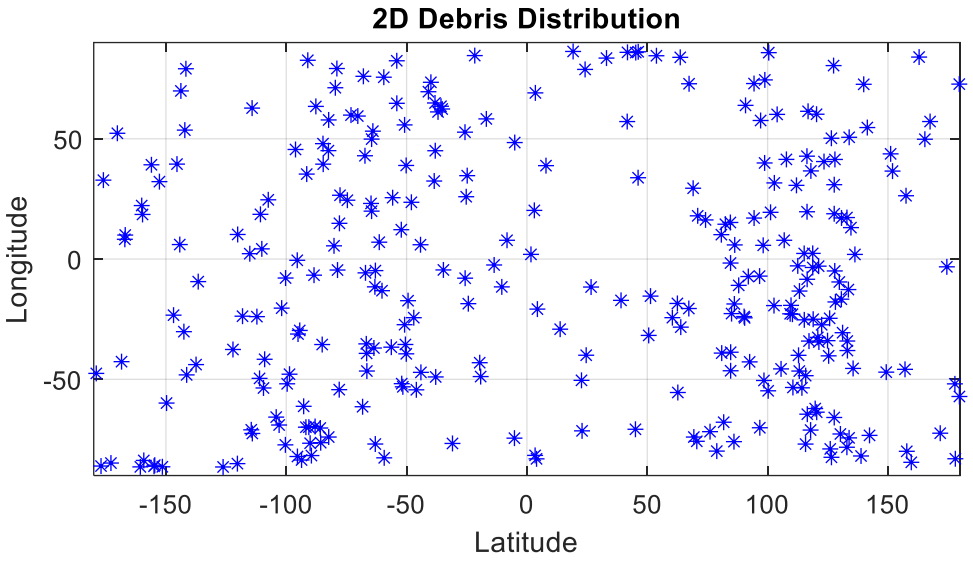

Figure 4-2 2-Dimensional Debris Field Snapshot

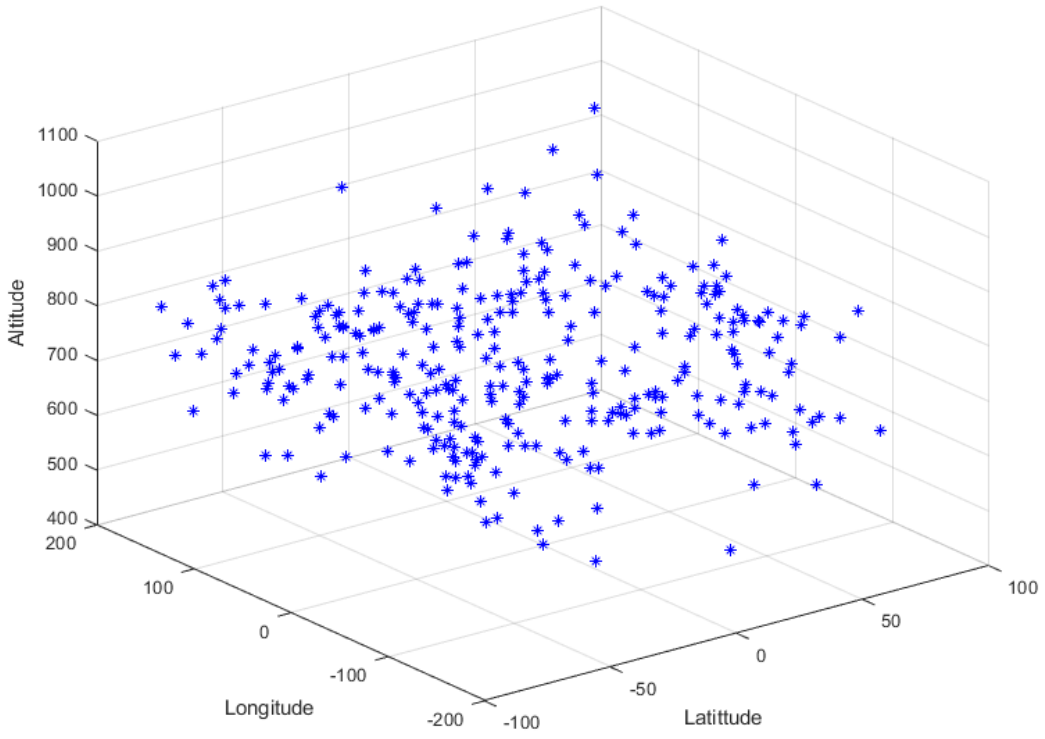

Figure 4-3 3-Dimensional Debris Field Snapshot

and replot, but the result would not assist in determining the location of the highest density of debris at any given epoch. It would simply replot the debris objects in a new epoch and a follow-on position in latitude, longitude, and altitude. MATLAB was also used in an attempt to develop a spherical graph of the debris cloud and a surface plot and with local and global optima but were unsuccessful due to the nature of the STK data that was used. This leaves us with what appears to be more of a TSP, which is complex in itself, let alone one with constantly moving stops.

The traditional TSP algorithm and the Moving Target TSP algorithm developed by Helvig, et al. (2003) were both considered to come to a solution to this problem. The moving target TSP algorithms cannot be applied to this problem, as the dynamics of the orbital debris movement do not fit the parameters suggested in any of the three versions of the moving target TSP. The traditional TSP algorithm requires the delivery locations to be fixed to come to a viable solution. To ensure the traditional TSP could not be applied to

solve this problem, a predictive algorithm was applied to determine the future location of the objects in the debris field.

The outcome of the predictive algorithm analysis was used to develop Figure 4-4, which used a small subset, only seven objects, of the data available from the Iridium 33 debris field. Qlik, an open-source predictive analysis tool, was used to develop the figure. The figure captures the latitude, longitude, and altitude of the randomly selected debris as they orbit the Earth, in a snapshot in time. The vertical and horizontal labels are longitude and latitude, respectively. The number above or below each blue bubble represents the altitude, in kilometers, of each debris object. As can be seen in the figure, the latitude, longitude, and altitude of the selected debris objects vary significantly. Notably, the altitude variance is more significant than the latitude and longitude. Figure 4-5 captures the entire Iridium 33 debris field and portrays the compounded complexity of this problem when the entire debris field is modeled. Observing the variances in both figures, one can come to the conclusion that the TSP is not applicable to this dataset, especially from an orbital mechanics perspective. First and foremost, as previously mentioned, the data

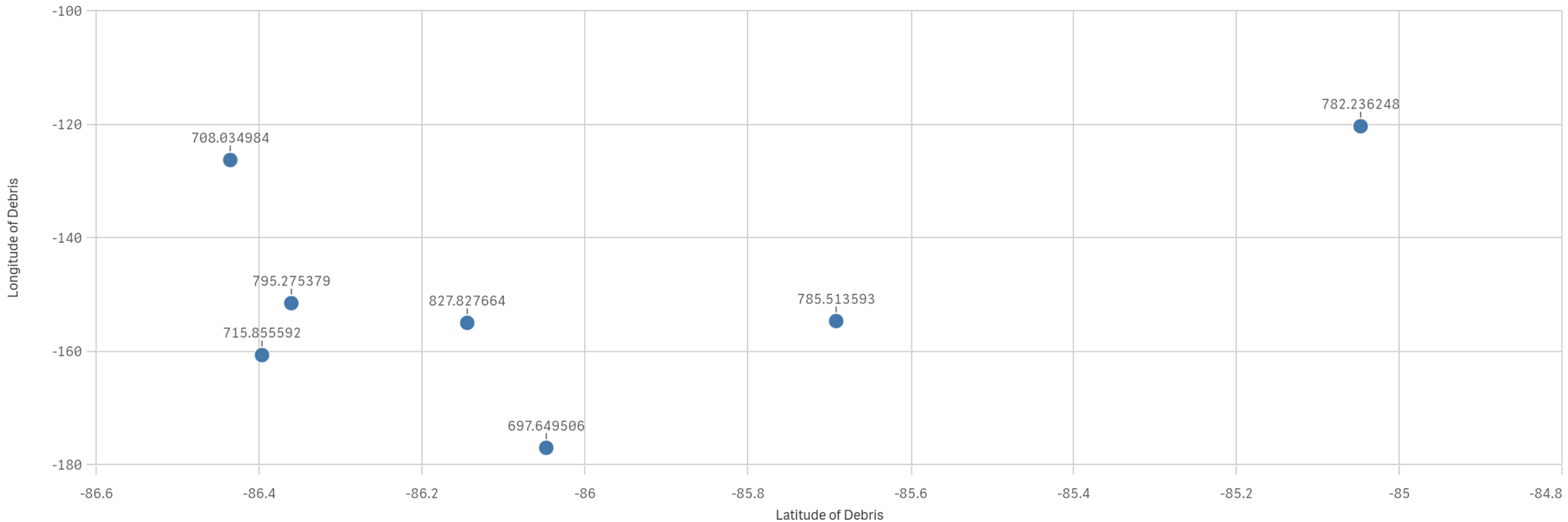

Figure 4-4 Selected Debris Latitude, Longitude and Altitude

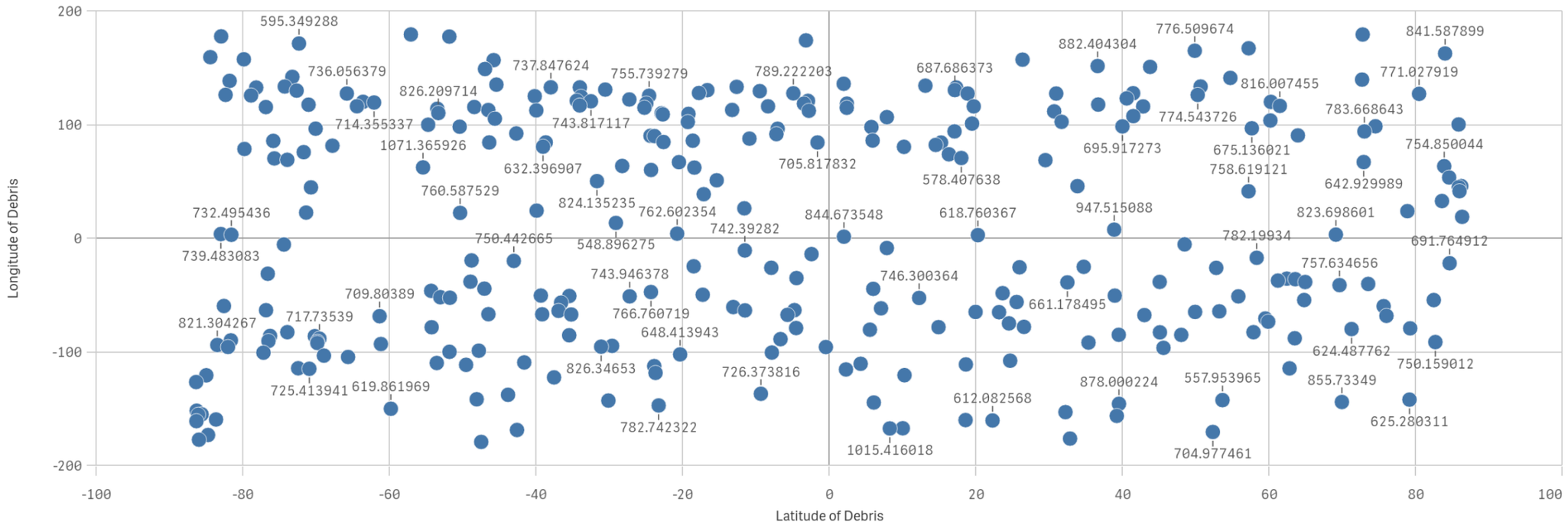

Figure 4-5 Entire Iridium 33 Debris Field with Latitude, Longitude and Altitude

used for these analyses is from one snapshot in time. Due to their differing altitudes, the debris is orbiting the Earth at varying velocities, which means rarely will the same two pieces be in the vicinity of each other as they orbit the Earth. Although predicting the orbit of the debris is possible, attempting to collect the debris in this dynamic environment is not realistically feasible. On-orbit out-of-plane changes would be required to move from one object to another. Orbital plane changes, as described in the orbital mechanics demonstration, are extremely costly from a fuel perspective and are typically avoided whenever possible. Additionally, dV maneuvers would be required to be conducted simultaneously with the out-of-plane maneuvers, which would add to the untenable cost of fuel. Performing the in-plane dV and out-of-plane maneuvers to try and capture constantly moving debris at various latitude, longitude and altitude would be like playing the children's game, whack-a-mole. As a result, predictive analysis, although a powerful tool in itself, does not result in an outcome that can be applied to solve the TSP in this case. Therefore, the traditional TSP is not applicable to this problem, either.

CONCLUSIONS AND RECOMMENDATIONS

During my research and attempt to use MATLAB with the modified MOBA and the traditional TSP and moving target TSP. A predictive algorithm was used to project where the debris field would be located in the future so, perhaps, the moving target TSP could be used to dictate the vector and velocity of the drone swarm for engagement with debris. I was not able to achieve useful results to solve this problem with any approach I attempted. Therefore, neither the MOBA nor either version of the TSP can be used to develop a solution to maneuver the drone swarm into the highest concentration of debris for collection. Perhaps an algorithm or software solution will be developed in the future that can be used to address the dire need to mitigate the ever-growing population of orbital debris.

One of the causes for this failure was due, in part, to the dynamic nature of the debris field and fact that the TLE data could not be ingested into MATLAB. MOBA uses a random number generator to obtain solutions in MATLAB and could not be adapted to use the TLE data. Additionally, because the data could not be ingested for

analysis, neither a surface plot nor a Pareto optimal solution could be produced.

After detailed analysis of the problem and unsuccessful results from MATLAB, it was discovered this problem is more akin to a TSP, but with moving stops. In TSP, a given number of cities with various distances between separate pairs of the cities is used to determine the shortest, most efficient and lowest cost path through all the cities. The traveling salesman can only pass through each city once and must return to the point of origin. With the orbital debris problem, the “cities” can be characterized as the debris objects, which are constantly moving at different velocities around the Earth in a three-dimensional space.

There have been proposals made to solve the moving city TSP. Helvig, et al. (2003) proposed solving the moving-target TSP. In their work, they developed three approaches to solving the moving-target TSP. The first approach is limited to one dimension and the number of moving targets must be small, moving at a constant speed and in various directions. The second approach requires the salesman to return to the point of origin after each encounter with a target. It also

dictates that all the targets must be moving on a straight line to or from the origin. The third design calls for more than one pursuer and all pursuers must move at the same velocity.

Since none of the proposed solutions developed by Helvig, et al. (2003) are applicable to this problem, it would be impossible to solve a TSP with constantly moving cities in a three-dimensional space. Therefore, it is not possible to solve this problem with moving debris. Further, it is not possible to determine where the highest density of space debris is in the debris field because of the dynamic nature of the debris field.

As a result, the hypothesis is unsupported and cannot currently be proven. Future work with the available data may prove otherwise.

There are benefits, and weaknesses, to the approach presented in this paper.

Benefits include:

1. With future advances in technology, a metaheuristic algorithm or specialized software may be developed to help solve the dilemma of space debris mitigation. This work could expose others to the idea of using a metaheuristic

algorithm to mitigate space debris. Metaheuristic algorithms are powerful tools capable of solving very complex problems. Future work could lead to a solution for identifying the highest concentration of debris so a mission could be safely launched into that area of a debris field.

2. The idea of using a swarm of drones to better map and track space debris smaller than 10 centimeters could use further analysis. Having knowledge of the entire space debris environment is important. Therefore, having an on-orbit capability to track and catalog space debris could be a huge advance in our ability to track small space debris objects and predict when and where close approaches may occur. Possessing that kind of a priori knowledge of the debris environment could help prevent collisions and avoid the Kessler syndrome, especially considering the launch of thousands of new spacecraft in the LEO regime.
3. Repurposing of a Whipple Shield may, in the future, be a viable capability for capturing space debris. We already know they have a proven record of preventing small debris

objects from damaging the International Space Station. As an alternative, a modified Whipple Shield could be able to capture and hold space debris smaller than 10 centimeters. Future technology may prove the development of nanomaterials are capable of reinforcing a Whipple Shield so one could capture larger space debris, too.

Weaknesses include:

1. A project similar to the one outlined in this paper would cost millions of dollars to execute. It remains to be seen who or what entity would be prepared to commit to such an investment. JAXA and the UK Space Agency, NASA equivalents in Japan and the UK, respectively, have funded the aforementioned efforts to remove large orbital debris objects; derelict spacecraft and rocket bodies, to be specific. Although those efforts are notable, they do little to address the problem of debris smaller than 10 centimeters. Perhaps peer pressure will implore other space faring nations to step up to help solve this problem. A collective effort will be

required to remove the thousands of debris objects on orbit. The best way to minimize the overwhelming cost of a space debris collection mission is to cost share between major space faring nations. Several countries contribute to the cost of the International Space Station and the same could be done to address the problem of space debris mitigation. An on-orbit collision with space debris could, in fact, result in the Kessler Syndrome and in that case all space faring nations will be involved.

2. The United Nations Office for Outer Space Affairs (UNOOSA) agreements 1971, stipulates the country launching a spacecraft is absolutely responsible for damage that may be caused by such spacecraft, whether in space, on the Earth or to aircraft in flight. The agreement goes on to outline procedures to settle demands for monetary settlements as a result of damages caused by spacecraft. No settlement was ever reached in the Iridium/Cosmos collision. Both countries disavowed any and all responsibility

for the collision and neither country has done anything to clean up the debris field.

3. There are a lot of assumptions in this paper. Those assumptions will need to be addressed in future work to make this a viable solution for space debris mitigation. Space qualified drones will need to be designed and built, a modified Whipple shield will need to be designed and built. Ground stations capable of TT&C of the space craft will also be required. Some of these risks and weaknesses could be resolved with technology that already exists and others will require future work to mitigate.

Future Work

Future work to solve the orbital debris problem is necessary. As of October 2022, more than 3,500 Starlink satellites have been launched. Space-X plans to launch at least 12,000 Starlink satellites to complete its constellation. This should be cause for concern for anyone who is already aware of the problem of orbital debris. Astronomers are already complaining about the Starlink constellation

causing irregularities in their telescopic imagery. These irregularities, they say, are impeding on their ability to identify, monitor and track asteroids and other objects in space. One would assume the same could be true for some sensors in the SSN. If we lose the ability to track debris items larger than 10 centimeters as a result of these mega constellations, it is only a matter of time before a large cascading debris event occurs.

Space-X has plenty of competition launching satellites into LEO. Several satellite manufacturers are rushing to get into the LEO market to deliver on-orbit connectivity, globally. The need for connectivity in remote locations is driving this market and it will only grow. These competing mega-constellations, if not properly governed, create significant cause for concern.

LEO will become more and more congested and the odds of a catastrophic, cascading collision, the Kessler Syndrome realized, could occur. Without the means of cleaning up after such a catastrophe our use of space will end.

Future work to address space debris could include a mission with a spacecraft outfitted with a LASER. The concept and

employment of LASER ablation has been discussed for some time. LASER ablation can gently nudge a piece a space debris into the atmosphere by illuminating the debris with a high-powered LASER for a short period of time. The LASER energy imparted upon the debris causes a tiny layer of the outer surface of the debris to ablate. The result of the ablation is out-gassing, which is caused by the molecules in the outer layer of the debris heating up and moving away from the object. This out-gassing produces thrust. The thrust, if imparted from the correct direction, can cause the debris to move toward the atmosphere and eventually burn up as it reenters the atmosphere. A satellite could be launched with the capability to perform such ablation techniques and clean-up existing debris fields. Such a mission could create the opportunity for a commercial mission in space to reduce or possibly even eliminate space debris and keep our skies open to space exploration. The mission would execute a detect, acquire, engage, assess kill chain, which is further addressed in the following.

Detection - The spacecraft could use its organic RADAR detection capability to detect and map the portion of debris field within

its field of view. That field of view would change as the spacecraft orbits the Earth.

Acquisition - The spacecraft will use its organic RADAR to acquire pieces of space debris to be targeted and formulate a firing solution to be passed to the LASER subsystem.

Engagement - The LASER will engage the space debris, firing at it for a prescribed period of time based on its RADAR cross section.

Assessment - Following engagement with the space debris by the LASER, the spacecraft will again map the field of view to ensure the piece of debris was actually ablated by the LASER and is moving away from the debris field and toward the atmosphere.

The mission will be deemed a success if it can locate, target, fire upon and de-orbit at least 85% of 50 pieces of debris. This will be determined by post-firing damage assessment conducted by the RADAR, onboard processors and ground operations personnel.

Last, at end of spacecraft life, mission controllers will command the vehicle to re-enter the atmosphere for destruction.

LIST OF REFERENCES

BIBLIOGRAPHY

18th Space Control Squadron. (2020, August). *Space Track.* Retrieved from Space Track: https://www.space-track.org/

Abualigah, L., & Diabat, A. (2021). Advances in Sine Cosine Algorithm: A comprehensive survey. *Artificial Intelligence Review*, 2567-2608.

Abualigah, L., Shehab, M., Alshinwan, M., Mirjalili, S., & Elaziz, M. (2021). Ant Lion Optimizer: A Comprehensive Survey of Its Variants and Applications. *Computational Methods in Engineering*, 1397-1416.

Ali, M. M. (2005). A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. *Journal of Global Optimization*, 635-372.

Ali, M., & Pant, M. (2011). Improving the Performance of Differential Evolution Algorithm using Cauchy Mutation. *Soft Computing 15*, 991-1007.

Anz-Meador, P. D., Opiela, J. N., & Shoots, D. (2018, July 4). *History of On-Orbit Satellite Fragmentations.* Retrieved from NASA: https://orbitaldebris.jsc.nasa.gov/library/20180008451.pdf

Arora, S., & Singh, S. (2018, March 08). *Butterfly optimization algorithm: a novel approach for global optimization.* Retrieved from Springer-Verlag GmbH Germany, part of Springer Nature 2018: https://fardapaper.ir/mohavaha/uploads/2018/07/Fardapaper-Butterfly-optimization-algorithm-a-novel-approach-for-global-optimization.pdf

Askarzadeh, A. (2016, March 3). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers and Structures*, pp. 1-12.

Assiri, A. S., Hussien, A. G., & Amin, M. (2020). Ant Lion Optimization: variants, hybrids,. *IEEE Access*, 2-21.

Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition. *IEEE Congress on Evolutionary Computation. 7*, 4661-4666.

Aziz, M. F. (2010). Quantum Particle Swarm Optimization for Elman Recurrent Network. *2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation* (pp. 133-137). Washington, D.C.: IEEE Computer Society.

Balev, S., Yanev, N., Freville, A., & Andonov, R. (2008). A dynamic programming based reduction procedure for the multidimensional 0–1 knapsack problem. *European Journal of Operational Research 186*, 63-76.

Beheshti, Z., & Shamsuddin, S. M. (2013). A Review of Population-based Meta-Heuristic Algorithm. *International Journal of Soft Computing and its Applications, Vol 5, No. 1*, 1-35.

Beheshti, Z., Mariyam, S., & Shamsuddin, H. (2013). A Review of Population-based Meta-Heuristic Algorithm. *International Journal of Advances in Soft Computing and its Applications*, 1-36.

Beheshti, Z., Shamsuddin, S. M., & Yuhaniz, S. S. (2013). Binary Accelerated Particle Swarm Algorithm (BAPSA) for discrete optimization problems. *Journal of Global Optimization 57*, 549-573.

Bengel, C. (2021, July 23). *Olympics 2021: Drone show highlights opening ceremonies at Tokyo Games*. Retrieved from CBS Sports: https://www.cbssports.com/olympics/news/olympics-2021-drone-show-highlights-opening-ceremonies-at-tokyo-games/

Betts, J. (2000). Very low-thrust trajectory optimization using a direct SQP method. *Journal of Computational and Applied Mathematics 120*, 27-40.

Birattari, M., Paquete, L., Stutzle, T., & Varrentrapp, K. (2001). *Classification of Metaheuristics and Design of Experiments for Analysis of Componenets.* Darmstadt: Darmstadt University of Technology.

Blanchard, B. S., & Fabrycky, W. J. (2011). *Systems Engineering and Analysis.* Upper Saddle River: Prentice Hall.

Brambilla, M., Ferrante, E., & Birattari, M. (2012). *Swarm robotics: A review from the swarm engineering perspective.* Bruxelles: IRIDIA - Technical Report Series.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self Organization in Biological Systems.* Princeton: Princeton University Press.

Chen, X., Tianfield, H., Mei, C., Du, W., & Liu, G. (2016). Biogeography-based learning particle swarm optimization. *Soft Computing*, 7519-7541.

Choubey, N. (2013). Moving Target Travelling Salesman Problem using Genetic Algorithm. *International Journal of Computer Applications*, 30-34.

Committee on Space Debris, N. R. (1995). *Orbital Debris: A Technical Assessment.* Washington, D.C.: National Academy Press.

Committee on the Peaceful Use of Outerspace (COPUOS). (2022, February 9). *2022 COPUOS STSC – U.S. on Space Debris*. Retrieved from U.S. Missions to International Organizations in Vienna: https://vienna.usmission.gov/2022-copuos-stsc-space-debris/

Cordon, O., Damas, S., & Santamaria, J. (2006). A fast and accurate approach for 3D image registration using the scatter search evolutionary algorithm. *Pattern Recognition Letters 27*, 1191-1200.

Cuevas, E., Cienfuegos, M., Zaldivar, D., & Perez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider . *Expert Systems with Applications*, pp. 6374-6384.

Dahiya, A., & Sangwan, S. (2018). Literature Review on Genetic Algorithm Volume 5, Issue 16. *International Journal of Research*, 1142-1146.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms.* Kanpur: Wiley.

Devi, R. V., Sathya, S. S., & Kumar, N. (2017). MONKEY ALGORITHM FOR ROBOT PATH PLANNING AND VEHICLE ROUTING PROBLEMS. *INTERNATIONAL CONFERENCE ON INFORMATION,COMMUNICATION & EMBEDDED SYSTEMS (ICICES 2017)* (pp. 1-6). Chennai: Institute of Electrical and Electronics Engineers (IEEE).

Dharanyadevi, P., Preethi, R., Suriyaakumar, G., & Venkatalakshmi, K. (2014). Proficient Path Optimization by Fusion of Intelligent Water Drop and Ford-Fulkerson's Algorithm in VANET Milieu. *International Journal of Scientific Research Engineering & Technology*, 858-865.

Directive, C. o. (2015). *Bulk Collection of Signals Intelligence: Technical Options.* Washington, D.C.: National Academies Press.

Dokeroglu, T., Kiziloz, H. E., & Deniz, A. (2022). A Comprehensive Survey on Recent Metaheuristics for Feature Selection. *Neurocomputing*, 1 - 67.

Dorigo, M., & Di Caro, G. (1992). *Ant Colony Optimization: A New Meta-Heuristic.* Bruxelles: Universite Libre de Bruxelles.

ESA. (2019, September 9). *ESA commissions world's first space debris removal*. Retrieved from The European Space Agency: https://www.esa.int/Safety_Security/Clean_Space/ESA_commissions_world_s_first_space_debris_removal#:~:text=ClearSpace%2D1%20will%20be%20the,as%20well%20as%20debris%20removal.

Fernandez, A. (2017). Understanding Genetic Algorithms. A use case in the organizational field. *Becoming Human: Artificial Intelligence*.

Floreano, D., & Mattiussi, C. (2000). *Bio-Inspired Artificial Intelligence.* Cambridge: MIT Press.

Gandomi, A. H., Alavi, A. H., & Yang, X. S. (2013). Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems. *Enigineering With Computers*.

Goebel, D. M., Polk, J. E., Sandler, I., Mikellides, I. G., Brophy, J., Tighe, W. G., & Chien, K.-R. (2009). *Evaluation of 25-cm XIPS Thruster Life for Deep Space.* Ann Arbor: Electric Rocket Propulsion Society.

Grasshopper optimization algorithm for multi-objective optimization problems. (2017, August 4). *Applied Intelligence*, pp. 1-17.

Hassanien, A. E., & Emary, E. (2015). *Swarm Intelligence.* Boca Raton: CRC Press.

Hazen, M. (2008). *Search Strategies for Global Optimization.* Seattle: University of Washington.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019, February 18). Harris hawks optimization: Algorithm and applications. *FutureGenerationComputerSystems*, pp. 1-36.

Helvig, C. S., Robins, G., & Zelikovsky, A. (2003). The moving-target traveling salesman problem. *Journal of Algorithms*, 153-174.

Howell, E. (2014, March 3). *Space Junk Clean Up: 7 Wild Ways to Destroy Orbital Debris*. Retrieved from Space: https://www.space.com/24895-space-junk-wild-clean-up-concepts.html

Jacobo, J. (2018, February 9). *Thousands of drones used for light show during Olympics openinng ceremony*. Retrieved from abcNEWS.

Jezowski, J., Bochenek, R., & Ziomek, G. (2005). Random search optimization approach for highly multi-modal nonlinear problems. *Advances in Engineering Software 36*, 504-517.

Kang, F., Li, J., & Ma, Z. (2011). Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences*, 3508-3531.

Karaboga, D. (2005). *AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION.* Kayseri: Erciyes University.

Karaboga, D., & Basturk, B. (2006). *Artificial Bee Colony (ABC) Optimiization Algorithm for Solving Constrained Optimization Problems.* Kayseri: Erciyes University.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mech 213*, 267-289.

Kelso, D. T. (2022, March 23). *NORAD General Perturbations (GP) Element Sets*. Retrieved from CelesTrak: https://www.celestrak.com/NORAD/elements/

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *IEEE*, 1942-1945.

Kessler, D. J., & Cour-Palais, B. G. (1978). Collision Frequency of Artificial Satellites; The Creation of a Debris Belt. *Journal of Geophysical Research*, 2637-2646.

Kyle, E. (2021, 12 31). *Space Launch Report - Orbital Launch Summary by Year*. Retrieved 2 17, 2015, from Space Launch Report: http://www.spacelaunchreport.com/logyear.html

Li, M., Du, W., & Nian, F. (2014). An Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network. *Mathematical Problems in Engineering*, 1-7.

Li, S., Gong, W., & Gu, Q. (2021). Acomprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models. *Renewable and Sustainable Energy Reviews*, 2-33.

Li, S., Wei, Y., Liu, X., Zhu, H., & Yu, Z. (2022). A New Fast Ant Colony Optimization Algorithm: The Saltatory Evolution Ant Colony Optimization Algorithm. *Mathematics*, 1-22.

Liang, J., & Suganthan, P. (2013). *Problem Definitions and evauation criteria for the CEC 2014 special session on single objective real-parameter numerical optimization.* Zhengzhou: School of Electrical Engineering.

Liang, J., Qin, A., Suganthan, P., & Baskar, S. (2006). Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 10*, 281-295.

Luque-Chang, A., Cuevas, E., Fausto, F., Zaldivar, D., & Perez, M. (2018, December 2). Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives. *Mathematical Problems in Engineering*, pp. 1-29.

Martí, R., Gallego, M., & Duarte, A. (2010). A branch and bound algorithm for the maximum diversity problem. *European Journal of Operational Research 200*, 36-44.

McKie, R. (2022, January 8). *NASA engineers complete the unfolding of the James Webb space telescope*. Retrieved from The Guardian: https://www.theguardian.com/science/2022/jan/08/nasa-engineers-complete-the-unfolding-of-the-james-webb-space-telescope

Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper Optimization Algorithm: Theory, Variants, and Applications. *Digital Object Identifier*, 1-24.

Mirjalili, S. (2015, May 29). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, pp. 1053-1073.

Mirjalili, S. (2015, January 13). The Ant Lion Optimizer. *Advances in Engineering Software*, pp. 80-98.

Mirjalili, S. (2016, January 6). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, pp. 120-133.

Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018, April 17). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, pp. 1-17.

Mirjalili, S., & Lewis, A. (2016, January 15). The Whale Optimization Algorithm. *Advances in Engineering Software*, pp. 51-67.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017, July 9). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, pp. 1-29.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2013, December 11). Grey Wolf Optimizer. *Advances in Engineering Software*, pp. 46-61.

NASA. (2021, November 15). *Space Science Data Coordinated Archive*. Retrieved from NASA: https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1982-092A

NASA Orbital Debris Program Office. (2022, April 22). *Reference Documents*. Retrieved from Astromaterials Research & Exploration Science NASA Orbital Debris Program Office: https://www.orbitaldebris.jsc.nasa.gov/reference-documents/

Nielsen, E. A. (2015, September). *Orbital Debris Impacts Photo Gallery*. Retrieved from NASA Orbital Debris Program Office: https://orbitaldebris.jsc.nasa.gov/photo-gallery.html

Obayashi, S., Deb, K. P., Hiroyasu, T., & Murata, T. (2007). Evolutionary Multi-Criterion Optimization. *Evolutionary Multi-Criterion Optimization* (pp. 1-954). Matsushima: Springer.

Okwu, M., & Tartibu, L. (2021). *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications.* Switzerland: Springer.

Osman, I. (1995). Routing Problems: A Bibliography. *Annals of Operations Research 61*, 227-262.

Programs, C. f., & Council, N. R. (2011). *Limiting Future Collision Risk to Spacecraft: An Assessment of NASA's Meteoroid and Orbital Debris Programs.* Washinton, D.C.: National Academies Press.

Qasema, S. N., & Shamsuddin, S. M. (2011). Memetic Elitist Pareto Differential Evolution algorithm based Radial Basis Function Networks for classification problems. *Applied Soft Computing 11*, 5565-5581.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences 179*, 2232–2248.

Rathod, S., Saha, A., & Sinha, K. (2020). Particle Swarm Optimization and its applications in agricultural research. *Food and Scientific Reports*, 37-41.

Ray, S. (2022, February 9). *At Least 40 Starlink Satellites Launched By SpaceX Last Week Have Been Destroyed By Geomagnetic Storm*. Retrieved from Forbes: https://www.forbes.com/sites/siladityaray/2022/02/09/at-least-40-starlink-satellites-launched-by-spacex-last-week-have-been-destroyed-by-geomagnetic-storm/?sh=26d5d0de7d9b

ReportLinker. (2021, August 26). *Low Earth Orbit (LEO) Satellites Global Market Report 2020-30: COVID-19 Growth And Change*. Retrieved from Yahoo: https://www.yahoo.com/now/low-earth-orbit-leo-satellites-141600373.html

Rezaiee-Pajand, M., Abad, J. M., Karimipour, A., & Rezaiee-Pajand, A. (2021). Propose new implement models to determine the compressive, tensile and flexural strengths of recycled coarse aggregate concrete via imperialist competetive algorithm. *Journal of Building Engineering*, 1-11.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software*, 30-47.

Shah-Hosseini, H. (2007). Problem solving by intelligent water drops. *IEEE Congress on Evolutionary Computation* (pp. 3226-3231). Singapore: IEEE.

Shi, Y., & Eberhart, R. (1998). *A Modified Particle Swarm Optimizer.* Indianapolis: Indiana University, Purdue University .

Smith, J. C. (1999). *The Use of Shock Physics to Predict the Mechanics of Hypervelocity Impact.* Houston: Rice University.

Tharakeshwar, T. K., Seetharamu, K. N., & Prasad, B. D. (2017). Multi-objective optimization using bat algorithm for shell and tube heat exchangers. *Applied Thermal Engineering*, 1029-1038.

The Ant System: Optimization by a colony of cooperating agents. (1996). *IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol 26, No. 1*, 1-13.

Thomas Pfeiffer, T. S. (2021, April 15). *The British Want to Clean Up Outer Space*. Retrieved from Bloomberg Business Week: https://www.bloomberg.com/news/articles/2021-04-16/u-k-space-industry-sees-opportunity-to-remove-debris-from-orbit

Toklu, Y. C., & Bekdas, G. (2014, March 24). METAHEURISTICS AND ENGINEERING. Istanbul, Turkey.

Toroslu, I. H. (2004). Dynamic programming for multiple query optimization problem. *Information Porcessing Letters 92*, 149-155.

Turk, T. S. (2009). *SPSM 5770 FINAL REPORT WITH ABSTRACT.* Colorado Springs, CO.

Tzanetos, A., Fister, J. I., & Dounias, G. (2020). A comprehensive database of Nature-Inspired Algorithms. *Elsevier*, 2352-3409.

U.S. STRATEGIC COMMAND. (2014, January 31). *U.S. STRATEGIC COMMAND*. Retrieved September 27, 2014, from USSTRATCOM Space Control and Space Surveillance: http://www.stratcom.mil/factsheets/11/Space_Control_and_Space_Surveillance/

United Nations Office for Outer Space Affairs. (2010). *Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space.* Vienna: United Nations.

United Nations Office for Outer Space Affairs. (n.d.). United Nations Treaties and Principles On Outer Space, related General Assembly resolutions and other Documents. *ST/SPACE/61/Rev.1*. United Nations.

United States Government Orbital Debris Mitigation Standard Practices. (2019, December 2019). *NASA ORBITAL DEBRIS PROGRAM OFFICE*. Retrieved from NASA: https://orbitaldebris.jsc.nasa.gov/mitigation/

UNOOSA. (1971, November 29). *Convention on International Liability for Damage Caused by Space Objects*. Retrieved from United Nations Office for Outer Space Affairs: https://www.unoosa.org/pdf/gares/ARES_26_2777E.pdf

UNOOSA. (2014, January 15). *Online Index of Objects Launched Into Outer Space*. Retrieved January 15, 2014, from United Nations Office for Outer Space Affairs: http://www.oosa.unvienna.org/oosa/search.do

Wall, M. (2021, November 15). *Kessler Syndrome and the space debris problem*. Retrieved from Space: https://www.space.com/kessler-syndrome-space-debris

Walter Buckley, D. S. (2016). An introduction to “Society as a complex adaptive system". *Emergence: Complexity and Organization*.

Weeden, B. (2010). *2009 Iridium-Cosmos Collision Fact Sheet.* Broomfield: Secure World Foundation.

Weiner, C. (2021, March 21). *New Effort To Clean Up Space Junk Reaches Orbit*. Retrieved from NPR: https://www.npr.org/2021/03/21/979815691/new-effort-to-clean-up-space-junk-prepares-to-launch

White, J. E., Bate, R. R., & Mueller, D. D. (1971). *Fundamentals of Astrodynamics.* Dover: Dover Publications, Incorporated.

Xu, X.-L., & Xiong, Y.-Q. (2014). A method for calculating probability of collision between space objects. *Research in Astronomy and Astrophysics*, 601-609.

Yang, B., Wang, J., Zhang, X., Yu, T., Yao, W., Shu, H., . . . Sun, L. (2020). Comprehensive Overview of Meta-heuristic Algorithm Applications on PV Cell Parameter Identification. *Energy Conversion and Management*, 1-22.

Yang, X. S. (2010). Firefly Algorithm, Stochastic Test Functions and Design Optimisation. *International Journal of Bio-Inspired Computation Vol. 2, No. 2*, 78-84.

Yang, X. S. (2011). Bat Algorithm for Multi-objective Optimisation. *International Journal Bio-Inspired Computation, Vol. 3, No. 5,* , 267-274.

Yang, X.-S., & Gandomi, A. H. (2011). Bat Algorithm: A Novel Approach for Global Engineering Optimization. *Engineering Computations*, 464-483.

Your Dictionary. (2015, December 12). Retrieved from www.yourdictionary.com: https://www.yourdictionary.com/joule

Zhao, R., & Tang, W. (2008). Monkey Algorithm for Global Numerical Optimization. *Journal of Uncertain Systems Vol. 2, No. 3*, 165-176.

APPENDICES

Appendix A

This appendix contains Two Line Elements (TLE) for all trackable orbital debris resulting from the Iridium 33 and COSMOS 2251 spacecraft collision, which occurred on February 10th, 2009. There were clearly thousands of other pieces of debris as a result of the collision, however, current tracking capabilities only enable identification and tracking of which are greater than 10 centimeters in diameter. These additional pieces of debris, although not tracked, may be incidentally collected during harvesting or disposal operations. The smaller debris are more dangerous to space safety of flight, since we do not know where they are and cannot purposefully avoid them by maneuvering. These TLE will be used while considering the maximum efficiency for collection.

IRIDIUM 33
1 24946U 97051C 15335.60652574 .00000175 00000-0 56416-4 0 9998
2 24946 86.3876 162.6207 0005856 282.9461 146.7680 14.33439643953307
IRIDIUM 33 DEB
1 33772U 97051K 15335.64916171 .00013614 00000-0 24157-2 0 9992
2 33772 86.4142 172.4418 0031631 270.9185 130.6137 14.66820945356248

IRIDIUM 33 DEB
1 33773U 97051L 15335.83431062 .00000523 00000-0 16725-3 0 9993
2 33773 86.4026 161.9150 0010314 325.9737 34.0798 14.37737061356553
IRIDIUM 33 DEB
1 33775U 97051N 15335.80205039 .00000668 00000-0 23120-3 0 9992
2 33775 86.3683 158.1558 0012722 273.9727 146.4746 14.34276281355648
IRIDIUM 33 DEB
1 33776U 97051P 15335.72160817 .00000450 00000-0 15597-3 0 9996
2 33776 86.4064 169.5690 0012904 284.0258 138.6407 14.33425574355568
IRIDIUM 33 DEB
1 33777U 97051Q 15335.17582923 .00001123 00000-0 34784-3 0 9999
2 33777 86.3855 156.3864 0005980 114.6716 245.5106 14.40352442356629
IRIDIUM 33 DEB
1 33849U 97051S 15335.51711523 .00009013 00000-0 27858-2 0 9996
2 33849 86.1119 109.1251 0069889 115.8493 244.9934 14.39788055352291
IRIDIUM 33 DEB
1 33850U 97051T 15335.79133397 .00000753 00000-0 25742-3 0 9998
2 33850 86.3450 150.4110 0009926 284.6699 137.2623 14.35010534355440
IRIDIUM 33 DEB
1 33853U 97051W 15335.16620251 .00001485 00000-0 95783-3 0 9997
2 33853 86.0041 136.0291 0239778 184.5280 229.2507 13.89254887343797
IRIDIUM 33 DEB
1 33854U 97051X 15335.52196140 .00005577 00000-0 11402-2 0 9994
2 33854 86.2286 94.3015 0013627 252.3430 199.5325 14.60656001358769
IRIDIUM 33 DEB
1 33855U 97051Y 15334.12576410 .00008652 00000-0 24590-2 0 9991
2 33855 86.3873 208.2145 0104862 279.2193 79.7158 14.42084592348031
IRIDIUM 33 DEB
1 33858U 97051AB 15335.36089670 .00016723 00000-0 21996-2 0 9993

2 33858 86.1398 65.6392 0033226 168.1403 234.2623 14.79710453359239
IRIDIUM 33 DEB
1 33859U 97051AC 15335.03494376 .00001858 00000-0 60095-3 0 9998
2 33859 86.3452 156.6141 0043691 320.7433 136.5569 14.38057648354550
IRIDIUM 33 DEB
1 33860U 97051AD 15335.78422720 .00001694 00000-0 49870-3 0 9992
2 33860 86.3950 155.8282 0014001 309.2938 109.1850 14.43113326357067
IRIDIUM 33 DEB
1 33862U 97051AF 15335.53162436 .00001970 00000-0 55624-3 0 9993
2 33862 86.4167 160.6160 0054096 86.5622 297.6110 14.44379274357364
IRIDIUM 33 DEB
1 33864U 97051AH 15335.16851499 .00003504 00000-0 95623-3 0 9996
2 33864 86.2799 123.4002 0010412 150.1438 273.3605 14.47085345356897
IRIDIUM 33 DEB
1 33865U 97051AJ 15334.93783991 .00008042 00000-0 17282-2 0 9998
2 33865 86.3518 142.5916 0042062 258.2850 254.1024 14.57910137357146
IRIDIUM 33 DEB
1 33866U 97051AK 15335.10956189 .00001998 00000-0 99764-3 0 9999
2 33866 86.76 178.3689 0128119 230.1252 282.9025 14.12241353349654
IRIDIUM 33 DEB
1 33867U 97051AL 15334.93203312 .00002456 00000-0 58083-3 0 9992
2 33867 86.4185 151.6931 0028392 316.7995 43.0984 14.53526778358698
IRIDIUM 33 DEB
1 33868U 97051AM 15335.88678367 .00044688 00000-0 34322-2 0 9996
2 33868 86.3154 111.4641 0006294 321.2220 199.7240 15.01541964361107
IRIDIUM 33 DEB
1 33869U 97051AN 15335.50489753 .00006002 00000-0 20145-2 0 9991
2 33869 86.3737 179.5964 0085721 56.2865 304.6464 14.35024598352683
IRIDIUM 33 DEB

1 33870U 97051AP 15335.29702638 .00000941 00000-0 32876-3 0 9990
2 33870 86.3806 164.3093 0022283 311.1643 174.4033 14.34125013355195
IRIDIUM 33 DEB
1 33872U 97051AR 15335.16221216 .00037711 00000-0 33410-2 0 9995
2 33872 86.3145 109.8906 0008846 202.6827 279.7760 14.96044298360888
IRIDIUM 33 DEB
1 33873U 97051AS 15335.82180990 .00003306 00000-0 78945-3 0 9998
2 33873 86.3704 144.3794 0019350 290.7189 69.1943 14.53311768357871
IRIDIUM 33 DEB
1 33874U 97051AT 15335.49645466 .00002544 00000-0 96850-3 0 9994
2 33874 86.2912 159.3180 0085194 101.3300 343.7693 14.28563102352251
IRIDIUM 33 DEB
1 33875U 97051AU 15335.81671293 .00005664 00000-0 10374-2 0 9990
2 33875 86.4109 138.4934 0033806 209.9870 249.6803 14.65296487360488
IRIDIUM 33 DEB
1 33876U 97051AV 15335.43480403 .00014571 00000-0 24322-2 0 9995
2 33876 86.2002 94.4363 0024058 219.5986 182.4253 14.69657646358145
IRIDIUM 33 DEB
1 33878U 97051AX 15335.34979124 .00004259 00000-0 14116-2 0 9992
2 33878 86.3251 170.8730 0105886 101.9767 313.0767 14.34504001351975
IRIDIUM 33 DEB
1 33879U 97051AY 15335.76720832 .00011905 00000-0 22536-2 0 9990
2 33879 86.3646 136.3187 0014421 248.8433 111.1238 14.64189933358616
IRIDIUM 33 DEB
1 33881U 97051BA 15335.38748993 .00003467 00000-0 14894-2 0 9991
2 33881 86.3850 191.9845 0089233 148.0601 333.2399 14.22338341351368
IRIDIUM 33 DEB
1 33882U 97051BB 15335.58055819 .00075996 00000-0 40163-2 0 9996
2 33882 86.2948 92.7922 0019563 168.9691 319.7787 15.14996949362345

IRIDIUM 33 DEB
1 33884U 97051BD 15334.96736744 .00001226 00000-0 34763-3 0 9997
2 33884 86.3830 152.0514 0011105 81.8976 70.0067 14.44710204356945
IRIDIUM 33 DEB
1 33886U 97051BF 15335.22173321 .00000275 00000-0 93388-4 0 9997
2 33886 86.3844 163.1475 0013049 317.2005 161.9860 14.33032949355445
IRIDIUM 33 DEB
1 33887U 97051BG 15334.92312738 .00000235 00000-0 76168-4 0 9994
2 33887 86.3043 136.9778 0005765 342.3446 171.4278 14.34753523355937
IRIDIUM 33 DEB
1 33888U 97051BH 15334.91796269 .00003759 00000-0 12380-2 0 9997
2 33888 86.3261 156.4725 0057299 358.8398   1.2657 14.37026400353913
IRIDIUM 33 DEB
1 33950U 97051BK 15335.73897087 .00005986 00000-0 13955-2 0 9991
2 33950 86.3867 154.0318 0017747 199.1614 288.9507 14.54571434357001
IRIDIUM 33 DEB
1 33951U 97051BL 15335.16717189 .00037962 00000-0 38930-2 0 9992
2 33951 86.4312 164.0954 0003775 107.5220 345.8949 14.90261458358092
IRIDIUM 33 DEB
1 33952U 97051BM 15333.42953885 .00003927 00000-0 96691-3 0 9998
2 33952 86.3553 143.8069 0014088 275.2425 84.7173 14.51974798357081
IRIDIUM 33 DEB
1 33953U 97051BN 15335.30066238 .00001229 00000-0 40531-3 0 9990
2 33953 86.3944 165.2760 0022067 311.1342 171.6506 14.37240483355654
IRIDIUM 33 DEB
1 33954U 97051BP 15334.90539845 .00005413 00000-0 13478-2 0 9993
2 33954 86.3875 161.2904 0015325 225.7916 244.7916 14.51496459355929
IRIDIUM 33 DEB
1 33955U 97051BQ 15334.82783122 .00013098 00000-0 29466-2 0 9996

2 33955 86.3679 163.3763 0041115 288.0504 71.6226 14.55842137355236
IRIDIUM 33 DEB
1 33956U 97051BR 15335.85122895 .00006559 00000-0 13066-2 0 9998
2 33956 86.4294 169.8803 0006609 92.1430 331.0968 14.61904974356533
IRIDIUM 33 DEB
1 33959U 97051BU 15335.37937166 .00004373 00000-0 16288-2 0 9994
2 33959 86.3620 177.8525 0084617 72.4174 343.9126 14.29843181352191
IRIDIUM 33 DEB
1 33960U 97051BV 15334.92026144 .00001550 00000-0 44020-3 0 9991
2 33960 86.2759 121.4625 0004141 106.2632 352.1515 14.44842770356761
IRIDIUM 33 DEB
1 33961U 97051BW 15335.17399333 .00008431 00000-0 18366-2 0 9991
2 33961 86.2734 121.8367 0030819 260.7933 152.4396 14.57550185357098
IRIDIUM 33 DEB
1 33962U 97051BX 15335.37021494 .00004091 00000-0 92415-3 0 9990
2 33962 86.3555 143.4300 0008010 243.3730 192.0243 14.56055792357294
IRIDIUM 33 DEB
1 33964U 97051BZ 15335.11059876 .00028487 00000-0 30236-2 0 9996
2 33964 86.3314 121.4112 0017478 301.4619 112.0663 14.88809264359786
IRIDIUM 33 DEB
1 33965U 97051CA 15335.76641531 .00003559 00000-0 95890-3 0 9994
2 33965 86.3728 150.5873 0008663 226.8876 260.8732 14.47711591356915
IRIDIUM 33 DEB
1 33966U 97051CB 15335.31085682 .00000812 00000-0 25236-3 0 9991
2 33966 86.3957 158.0933 0010106 80.3138   4.1937 14.39817480356527
IRIDIUM 33 DEB
1 33967U 97051CC 15335.10507530 .00009374 00000-0 21194-2 0 9990
2 33967 86.3899 152.6809 0032132 226.9251 273.9174 14.55811834357392
IRIDIUM 33 DEB

1 34071U 97051CE 15335.21026650 .00000890 00000-0 28487-3 0 9995
2 34071 86.3490 147.5407 0006757 319.8632 40.2066 14.38512383356125
IRIDIUM 33 DEB
1 34075U 97051CJ 15335.53289100 .00038378 00000-0 35506-2 0 9996
2 34075 86.4462 170.4978 0018193 173.4493 216.1774 14.94242058357743
IRIDIUM 33 DEB
1 34076U 97051CK 15335.88542669 .00007961 00000-0 11780-2 0 9999
2 34076 86.3350 115.1232 0050151 193.7877 321.1889 14.74195931359931
IRIDIUM 33 DEB
1 34077U 97051CL 15335.50400046 .00001848 00000-0 60840-3 0 9991
2 34077 86.3973 167.2163 0018521 269.6248 174.1021 14.37629733354721
IRIDIUM 33 DEB
1 34079U 97051CN 15334.97313977 .00002188 00000-0 60101-3 0 9997
2 34079 86.3914 153.1140 0011242 53.1285 44.6141 14.46603530357417
IRIDIUM 33 DEB
1 34081U 97051CQ 15335.10178520 .00022869 00000-0 25694-2 0 9994
2 34081 86.3047 108.5978 0039092 260.1849 153.8858 14.86066070360619
IRIDIUM 33 DEB
1 34082U 97051CR 15335.62867889 .00005581 00000-0 16258-2 0 9993
2 34082 86.3737 203.5721 0154640 244.8698 143.0187 14.37281092350036
IRIDIUM 33 DEB
1 34086U 97051CV 15334.79534058 .00003163 00000-0 10042-2 0 9992
2 34086 86.3904 169.8212 0028984 302.1775 185.7558 14.39494549354753
IRIDIUM 33 DEB
1 34088U 97051CX 15335.19100509 .00002150 00000-0 64187-3 0 9994
2 34088 86.4035 163.3218 0008686 278.0323 173.1852 14.42593897356247
IRIDIUM 33 DEB
1 34090U 97051CZ 15335.24563383 .00011758 00000-0 17323-2 0 9999
2 34090 86.4187 144.1108 0018485 273.0188 86.8915 14.75165346360037

IRIDIUM 33 DEB
1 34091U 97051DA 15335.37066155 .00002944 00000-0 11445-2 0 9995
2 34091 86.3728 179.6100 0077245 85.1739 331.1314 14.27908808352479
IRIDIUM 33 DEB
1 34093U 97051DC 15335.08952542 .00017742 00000-0 20176-2 0 9992
2 34093 86.2971 97.8987 0023444 137.7245 282.3893 14.85945734361520
IRIDIUM 33 DEB
1 34095U 97051DE 15334.70240924 .00009431 00000-0 24167-2 0 9996
2 34095 86.3866 164.2251 0053829 290.0742 69.4673 14.49379001355020
IRIDIUM 33 DEB
1 34097U 97051DG 15335.68405991 .00001598 00000-0 73056-3 0 9991
2 34097 86.2043 143.2985 0103239 175.9155 307.3647 14.18113508351035
IRIDIUM 33 DEB
1 34098U 97051DH 15335.19998598 .00009398 00000-0 16321-2 0 9995
2 34098 86.1371 69.3224 0023311 218.4040 182.0210 14.67914659358908
IRIDIUM 33 DEB
1 34099U 97051DJ 15335.39357810 .00006186 00000-0 14729-2 0 9996
2 34099 86.3700 150.8675 0014400 200.6815 234.7827 14.53625408356740
IRIDIUM 33 DEB
1 34101U 97051DL 15335.77274957 .00005091 00000-0 18270-2 0 9991
2 34101 86.3807 198.4010 0119222 191.5561 168.2893 14.29826096350429
IRIDIUM 33 DEB
1 34102U 97051DM 15334.86292009 .00003550 00000-0 15124-2 0 9992
2 34102 86.3800 205.1216 0120405 237.1622 216.2529 14.21095369349237
IRIDIUM 33 DEB
1 34103U 97051DN 15335.52389969 .00005740 00000-0 14794-2 0 9994
2 34103 86.3500 149.1261 0018653 146.5670 297.7787 14.49832431356225
IRIDIUM 33 DEB
1 34104U 97051DP 15335.24818513 .00006048 00000-0 11779-2 0 9998

2 34104 86.3468 130.6620 0012096 347.3185 67.0617 14.62884422358583
IRIDIUM 33 DEB
1 34105U 97051DQ 15334.91680498 .00017599 00000-0 26250-2 0 9994
2 34105 86.3582 139.3717 0013252 41.2378 318.9839 14.74685224357869
IRIDIUM 33 DEB
1 34106U 97051DR 15335.18836365 .00003043 00000-0 80664-3 0 9990
2 34106 86.3649 144.6480 0018573 57.9641 349.8764 14.48358463357243
IRIDIUM 33 DEB
1 34107U 97051DS 15335.83545293 .00001504 00000-0 38049-3 0 9990
2 34107 86.3897 150.4246 0019125 72.4617 354.3177 14.50228306357359
IRIDIUM 33 DEB
1 34143U 97051DW 15334.91316088 .00006330 00000-0 14527-2 0 9992
2 34143 86.3952 156.0959 0011358 213.3852 246.9761 14.55366824356246
IRIDIUM 33 DEB
1 34145U 97051DY 15335.63098856 .00001242 00000-0 48957-3 0 9996
2 34145 86.2186 131.2724 0068348 76.5030 284.3765 14.27069677352532
IRIDIUM 33 DEB
1 34146U 97051DZ 15335.50627689 .00001111 00000-0 38175-3 0 9991
2 34146 86.2896 137.8060 0038118 345.8965 101.8441 14.34834721354474
IRIDIUM 33 DEB
1 34147U 97051EA 15335.29744078 .00012736 00000-0 25241-2 0 9994
2 34147 86.4502 190.5729 0066945 335.2622 24.5382 14.60742325354990
IRIDIUM 33 DEB
1 34148U 97051EB 15335.26857886 .00019134 00000-0 26616-2 0 9990
2 34148 86.3972 151.1544 0011507 311.7804 48.2432 14.77684962358414
IRIDIUM 33 DEB
1 34150U 97051ED 15334.85609588 .00010674 00000-0 27887-2 0 9992
2 34150 86.4903 202.9093 0073579   9.9231 112.3496 14.47709133353843
IRIDIUM 33 DEB

1 34155U 97051EJ 15335.59270772 .00008854 00000-0 21063-2 0 9995
2 34155 86.4769 187.9263 0044373 266.9878 120.7579 14.53152581355642
IRIDIUM 33 DEB
1 34156U 97051EK 15335.16839269 .00058927 00000-0 45423-2 0 9993
2 34156 86.2957 123.4646 0010639 310.8094 174.7042 15.01305653357917
IRIDIUM 33 DEB
1 34157U 97051EL 15334.99768259 .00193946 00000-0 86266-2 0 9995
2 34157 86.3688 133.5396 0005872 40.9813 106.9602 15.20620803359433
IRIDIUM 33 DEB
1 34159U 97051EN 15335.00311765 .00000541 00000-0 19273-3 0 9999
2 34159 86.3535 157.4812 0031019 352.3865 161.5148 14.32173358354463
IRIDIUM 33 DEB
1 34160U 97051EP 15335.06713440 .00002529 00000-0 63499-3 0 9990
2 34160 86.4146 172.8295 0028005 289.2065 168.9209 14.50735128354852
IRIDIUM 33 DEB
1 34350U 97051ES 15335.89679997 .00000587 00000-0 18368-3 0 9997
2 34350 86.3293 138.7925 0021553 329.2499 197.5537 14.38980822355462
IRIDIUM 33 DEB
1 34351U 97051ET 15335.78072640 .00006065 00000-0 13372-2 0 9994
2 34351 86.4264 159.3929 0040139 236.3019 123.4360 14.56756075357221
IRIDIUM 33 DEB
1 34354U 97051EW 15335.81210403 .00014768 00000-0 26033-2 0 9999
2 34354 86.2161 104.6420 0035590 129.5672 230.8695 14.67016664356688
IRIDIUM 33 DEB
1 34358U 97051FA 15335.21224801 .00003118 00000-0 84825-3 0 9995
2 34358 86.3320 141.3197 0009706 170.1477 316.7582 14.47200882355856
IRIDIUM 33 DEB
1 34359U 97051FB 15335.65429258 .00008121 00000-0 15190-2 0 9994
2 34359 86.3616 139.0007 0039609 40.1431 89.0542 14.64257485357386

IRIDIUM 33 DEB
1 34360U 97051FC 15313.96451148 .26142098 -15243-5 15847-1 0 9998
2 34360 86.2998 127.1460 0010548 328.4367 159.3918 16.18189066356457
IRIDIUM 33 DEB
1 34361U 97051FD 15335.26340284 .00017686 00000-0 29343-2 0 9996
2 34361 86.3632 152.7042 0026148 203.2188 156.7848 14.69882884355553
IRIDIUM 33 DEB
1 34363U 97051FF 15335.02884551 .00004681 00000-0 13069-2 0 9991
2 34363 86.3592 161.9405 0046121 344.5650 113.6672 14.45454879353537
IRIDIUM 33 DEB
1 34366U 97051FJ 15334.95630535 .00001211 00000-0 41394-3 0 9995
2 34366 86.3776 163.7269 0027210 334.2203 185.4528 14.35372468354530
IRIDIUM 33 DEB
1 34367U 97051FK 15335.51879361 .00006163 00000-0 15191-2 0 9997
2 34367 86.4214 171.9366 0031296 271.0180 110.0356 14.51764070355333
IRIDIUM 33 DEB
1 34368U 97051FL 15335.04818199 .00006760 00000-0 19351-2 0 9993
2 34368 86.3927 182.5955 0071086 40.0777 56.7316 14.43432827352290
IRIDIUM 33 DEB
1 34374U 97051FS 15335.32212013 .00071735 00000-0 47778-2 0 9992
2 34374 86.4547 172.4190 0008806 64.4957 59.4246 15.06754542358193
IRIDIUM 33 DEB
1 34375U 97051FT 15334.65819722 .00000530 00000-0 26004-3 0 9993
2 34375 86.4348 211.3391 0108150 184.5959 204.2699 14.12878890350078
IRIDIUM 33 DEB
1 34376U 97051FU 15335.22342137 .00001917 00000-0 67551-3 0 9991
2 34376 86.3131 150.5442 0040747 354.8710   5.2062 14.33888229354283
IRIDIUM 33 DEB
1 34378U 97051FW 15335.04176941 .00006165 00000-0 17923-2 0 9998

2 34378 86.3537 172.6780 0068935 32.4260 120.3317 14.42767320352528
IRIDIUM 33 DEB
1 34486U 97051GB 15335.78467772 .00002201 00000-0 57495-3 0 9990
2 34486 86.4133 155.4393 0017987 287.1647 131.5854 14.48953110357030
IRIDIUM 33 DEB
1 34487U 97051GC 15334.92031127 .00000344 00000-0 10359-3 0 9991
2 34487 86.3457 141.7484 0031194 321.6022 192.9035 14.39480395354922
IRIDIUM 33 DEB
1 34488U 97051GD 15335.19406514 .00002714 00000-0 66361-3 0 9995
2 34488 86.3478 132.4484 0024734 272.1431 214.3506 14.52061902357906
IRIDIUM 33 DEB
1 34489U 97051GE 15335.36667378 .00004854 00000-0 17249-2 0 9999
2 34489 86.4198 197.2870 0079616 87.1576 273.8727 14.32439327350869
IRIDIUM 33 DEB
1 34490U 97051GF 15335.15689501 .00019792 00000-0 36491-2 0 9997
2 34490 86.3292 139.7459 0035373 208.3784 204.4931 14.65014599355845
IRIDIUM 33 DEB
1 34492U 97051GH 15335.87393772 .00001668 00000-0 10528-2 0 9996
2 34492 86.5075 289.6790 0301865 279.3952 105.1906 13.83726854340619
IRIDIUM 33 DEB
1 34493U 97051GJ 15335.14188459 .00009848 00000-0 21801-2 0 9995
2 34493 86.2708 124.2833 0058632 160.7495 253.0017 14.56028792354784
IRIDIUM 33 DEB
1 34496U 97051GM 15335.88361582 .00005168 00000-0 18124-2 0 9994
2 34496 86.3967 202.0504 0103100 183.3504 176.7013 14.31937894349403
IRIDIUM 33 DEB
1 34497U 97051GN 15334.99157432 .00000761 00000-0 34055-3 0 9999
2 34497 86.2815 160.3896 0091706 149.5686   2.5851 14.19045815349649
IRIDIUM 33 DEB

1 34503U 97051GU 15335.87941571 .00004467 00000-0 10320-2 0 9995
2 34503 86.3163 119.2662 0007945 300.0320 213.8310 14.55028328357656
IRIDIUM 33 DEB
1 34508U 97051GZ 15335.54139847 .00001421 00000-0 60389-3 0 9997
2 34508 86.3806 183.9455 0081958 96.7737 264.2787 14.22779131350649
IRIDIUM 33 DEB
1 34511U 97051HC 15334.79943262 .00003621 00000-0 10560-2 0 9990
2 34511 86.3915 165.4524 0024268 271.0849 216.9766 14.43752409354118
IRIDIUM 33 DEB
1 34515U 97051HG 15335.74062726 .00354947 00000-0 79143-2 0 9999
2 34515 86.2621 93.9438 0013521 263.1739 96.7978 15.42151793360366
IRIDIUM 33 DEB
1 34517U 97051HJ 15335.25282674 .00009620 00000-0 17551-2 0 9993
2 34517 86.3704 144.5072 0005930 67.4787 53.7753 14.65896236356889
IRIDIUM 33 DEB
1 34518U 97051HK 15334.30166540 .00009846 00000-0 18926-2 0 9999
2 34518 86.4017 157.1905 0004915 202.7608 157.3387 14.63558933355555
IRIDIUM 33 DEB
1 34520U 97051HM 15334.48316725 .00016223 00000-0 29843-2 0 9990
2 34520 86.3728 152.7822 0019758 216.8074 143.1783 14.65410191355096
IRIDIUM 33 DEB
1 34521U 97051HN 15334.95013469 .00001438 00000-0 43998-3 0 9992
2 34521 86.3871 156.9892 0006342 207.3405 314.4140 14.41175819355041
IRIDIUM 33 DEB
1 34522U 97051HP 15335.03479282 .00006001 00000-0 15651-2 0 9997
2 34522 86.3716 172.1188 0058729 26.1532 70.7713 14.48335423352683
IRIDIUM 33 DEB
1 34524U 97051HR 15335.19270267 .00004224 00000-0 11807-2 0 9992
2 34524 86.3485 151.6006 0034718 307.8923 51.9140 14.45648479354541

IRIDIUM 33 DEB
1 34525U 97051HS 15335.26444535 .00000578 00000-0 19090-3 0 9993
2 34525 86.3760 156.9801 0003499 210.2439 275.2068 14.36350291354799
IRIDIUM 33 DEB
1 34526U 97051HT 15334.96572856 .00011097 00000-0 23019-2 0 9991
2 34526 86.3493 148.1452 0021277 232.4373 287.5731 14.59962163354981
IRIDIUM 33 DEB
1 34529U 97051HW 15333.98816791 .00000377 00000-0 11959-3 0 9994
2 34529 86.3654 150.9830 0008878   8.3646 351.7698 14.37467760354502
IRIDIUM 33 DEB
1 34532U 97051HZ 15334.98884119 .00009198 00000-0 22533-2 0 9997
2 34532 86.3538 164.0148 0057971 333.8317 25.9961 14.51352576353016
IRIDIUM 33 DEB
1 34535U 97051JC 15335.86076991 .00009637 00000-0 16849-2 0 9990
2 34535 86.3355 127.9647 0013620 218.9055 305.2892 14.67740437357692
IRIDIUM 33 DEB
1 34538U 97051JF 15334.27241130 .00001091 00000-0 41343-3 0 9990
2 34538 86.2812 143.9125 0042533   0.4700 55.4074 14.29790833353490
IRIDIUM 33 DEB
1 34540U 97051JH 15334.91817317 .00002194 00000-0 56104-3 0 9996
2 34540 86.3231 125.5664 0026817 293.6669 221.2393 14.49835568357162
IRIDIUM 33 DEB
1 34593U 97051JK 15335.71848324 .00026461 00000-0 35701-2 0 9997
2 34593 86.3818 154.9909 0016147 254.2714 228.6139 14.78907129356206
IRIDIUM 33 DEB
1 34643U 97051JU 15335.84317168 .00004477 00000-0 12049-2 0 9997
2 34643 86.3880 160.3239 0032303 290.7692 69.0051 14.47531520352255
IRIDIUM 33 DEB
1 34648U 97051JZ 15335.75292559 .00003800 00000-0 27128-2 0 9991

2 34648 86.4114 262.1091 0250649 270.1301 114.6296 13.83209938341195
IRIDIUM 33 DEB
1 34651U 97051KC 15335.62384551 .00001918 00000-0 11193-2 0 9998
2 34651 86.2952 205.1968 0198694 76.1177 286.2014 13.98554186344486
IRIDIUM 33 DEB
1 34652U 97051KD 15334.90166374 .00001471 00000-0 41295-3 0 9991
2 34652 86.3368 134.9948 0036705 287.5894 225.1272 14.44987016354904
IRIDIUM 33 DEB
1 34657U 97051KJ 15334.89615134 .00045372 00000-0 48250-2 0 9992
2 34657 86.3113 118.9430 0019107 326.5092 188.8064 14.88627812357052
IRIDIUM 33 DEB
1 34690U 97051KM 15335.79201537 .00001576 00000-0 48842-3 0 9993
2 34690 86.3873 159.4384 0009506 296.2875 121.8046 14.40576991352190
IRIDIUM 33 DEB
1 34693U 97051KQ 15334.89146297 .00003171 00000-0 79079-3 0 9996
2 34693 86.3402 131.9410 0025675 269.5918 243.6143 14.51172992353829
IRIDIUM 33 DEB
1 34696U 97051KT 15334.98130634 .00002601 00000-0 13154-2 0 9995
2 34696 86.2082 166.1912 0152583 320.8944 193.6317 14.10009371346870
IRIDIUM 33 DEB
1 34698U 97051KV 15335.22465604 .00054153 00000-0 42742-2 0 9997
2 34698 86.3395 132.2734 0012188 329.1191 85.6484 15.00411673354537
IRIDIUM 33 DEB
1 34702U 97051KZ 15335.13599600 .00003543 00000-0 77206-3 0 9998
2 34702 86.3097 114.1535 0043033 213.2878 146.5623 14.57110040357842
IRIDIUM 33 DEB
1 34705U 97051LC 15335.34745985 .00010292 00000-0 23508-2 0 9997
2 34705 86.4801 193.5818 0057253 286.3088 73.1827 14.54640414353620
IRIDIUM 33 DEB

1 34706U 97051LD 15334.85794297 .00013917 00000-0 19905-2 0 9996
2 34706 86.3164 114.9303 0028603 262.4544 259.1142 14.76267995357855
IRIDIUM 33 DEB
1 34709U 97051LG 15335.60507629 .00030996 00000-0 33154-2 0 9992
2 34709 86.3184 124.1642 0012496 65.0590 295.1933 14.88540508357305
IRIDIUM 33 DEB
1 34764U 97051LH 15335.19489035 .00012211 00000-0 20773-2 0 9990
2 34764 86.3680 143.7666 0006076 129.9617 230.2132 14.69007959355055
IRIDIUM 33 DEB
1 34765U 97051LJ 15334.95312858 .00000614 00000-0 21378-3 0 9999
2 34765 86.3409 151.4497 0016539 316.0425 196.7562 14.33780360350383
IRIDIUM 33 DEB
1 34773U 97051LS 15334.50764916 .00004524 00000-0 12176-2 0 9994
2 34773 86.4156 166.1259 0015273 196.1187 193.3491 14.47770575356424
IRIDIUM 33 DEB
1 34774U 97051LT 15335.53552105 .00004266 00000-0 10240-2 0 9997
2 34774 86.4327 165.1819 0013127 119.4211 259.6098 14.53189977357043
IRIDIUM 33 DEB
1 34775U 97051LU 15335.01704286 .00002977 00000-0 93888-3 0 9997
2 34775 86.3824 164.8543 0029113 292.6272 221.7574 14.39810603352997
IRIDIUM 33 DEB
1 34825U 97051LZ 15335.26612476 .00004345 00000-0 11518-2 0 9999
2 34825 86.3754 154.8612 0012854 219.5705 265.7613 14.48512510352082
IRIDIUM 33 DEB
1 34827U 97051MB 15335.26598471 .00006451 00000-0 11860-2 0 9999
2 34827 86.3889 150.1555 0015238 301.5535 182.9892 14.65456866353327
IRIDIUM 33 DEB
1 34833U 97051MH 15335.27043164 .00000972 00000-0 30961-3 0 9997
2 34833 86.3976 161.8860 0022368 321.5680 90.3775 14.38732381349151

IRIDIUM 33 DEB
1 34868U 97051MN 15335.44003875 .00006977 00000-0 17811-2 0 9998
2 34868 86.5425 218.6731 0137286 34.5173 326.4848 14.44843238348583
IRIDIUM 33 DEB
1 34869U 97051MP 15334.60587814 .00009405 00000-0 32321-2 0 9998
2 34869 86.4554 205.9148 0087700 74.1493 315.0296 14.33800770345093
IRIDIUM 33 DEB
1 34870U 97051MQ 15333.42973186 .00001942 00000-0 51770-3 0 9998
2 34870 86.3213 129.1955 0030187 286.5311 103.2682 14.47757038357669
IRIDIUM 33 DEB
1 34889U 97051MR 15333.44295959 .00006784 00000-0 15348-2 0 9994
2 34889 86.3751 150.9580 0009227 190.1221 169.9803 14.56052580350615
IRIDIUM 33 DEB
1 34890U 97051MS 15334.81871679 .00006764 00000-0 21131-2 0 9992
2 34890 86.3099 161.6976 0081084 39.7330 21.4231 14.38769841348798
IRIDIUM 33 DEB
1 34893U 97051MV 15335.08570561 .00033057 00000-0 47321-2 0 9995
2 34893 86.1338 95.2307 0046697 268.4863 91.1006 14.75687826355311
IRIDIUM 33 DEB
1 34895U 97051MX 15335.23600418 .00040316 00000-0 36232-2 0 9994
2 34895 86.4081 146.9902 0037858 132.2309 228.2144 14.94946033354446
IRIDIUM 33 DEB
1 34896U 97051MY 15335.24197029 .00001489 00000-0 39405-3 0 9999
2 34896 86.3355 130.0328 0035830 271.9520 206.3726 14.47827990351527
IRIDIUM 33 DEB
1 34897U 97051MZ 15333.95421988 .00022900 00000-0 22442-2 0 9993
2 34897 86.4250 153.9021 0017927 306.8676 53.0907 14.92029663352883
IRIDIUM 33 DEB
1 34898U 97051NA 15335.86270602 .00005490 00000-0 10801-2 0 9992

2 34898 86.3347 119.6339 0028530 210.9677 148.9854 14.62190964352929
IRIDIUM 33 DEB
1 34899U 97051NB 15334.98835593 .00013572 00000-0 27635-2 0 9994
2 34899 86.4008 165.3290 0035958 280.2934 233.4853 14.60541126347681
IRIDIUM 33 DEB
1 34926U 97051NF 15335.00783310 .00003618 00000-0 18744-2 0 9999
2 34926 86.1611 158.7436 0174296   8.7066 145.4886 14.07133278346150
IRIDIUM 33 DEB
1 34928U 97051NH 15333.78052350 .00002807 00000-0 63942-3 0 9997
2 34928 86.4040 144.9097 0032861 277.2758 144.4892 14.55235171352198
IRIDIUM 33 DEB
1 34931U 97051NL 15334.63797529 .00081619 00000-0 71850-2 0 9994
2 34931 86.3321 134.7062 0004804 158.3434 332.4831 14.96086996348852
IRIDIUM 33 DEB
1 34983U 97051NP 15334.84063260 .00012082 00000-0 27704-2 0 9999
2 34983 86.3671 161.6600 0041864 277.7538 81.8917 14.54937522346944
IRIDIUM 33 DEB
1 34985U 97051NR 15334.31018669 .00000866 00000-0 30707-3 0 9991
2 34985 86.3739 162.7035 0023259 329.3847 151.1190 14.33287177345876
IRIDIUM 33 DEB
1 34986U 97051NS 15335.27756495 .00004383 00000-0 12201-2 0 9990
2 34986 86.3886 165.1543 0031068 276.2580 83.5084 14.45931443346306
IRIDIUM 33 DEB
1 34987U 97051NT 15334.90460449 .00014612 00000-0 23815-2 0 9990
2 34987 86.3409 126.8650 0008759 271.4062 241.8314 14.70877415357666
IRIDIUM 33 DEB
1 35049U 97051NV 15334.96854932 .00005034 00000-0 11780-2 0 9996
2 35049 86.3764 144.9546 0011705 72.6193 83.3988 14.54421881181188
IRIDIUM 33 DEB

1 35050U 97051NW 15335.90378492 .00014080 00000-0 24911-2 0 9996
2 35050 86.3161 127.1027 0020359 219.5898 296.1853 14.67141861348324
IRIDIUM 33 DEB
1 35051U 97051NX 15335.25641384 .00000958 00000-0 31752-3 0 9999
2 35051 86.3270 143.7876 0007452 250.8593 229.4426 14.36892001343573
IRIDIUM 33 DEB
1 35052U 97051NY 15334.94392095 .00000337 00000-0 11088-3 0 9998
2 35052 86.3311 144.5019 0011998 173.6011 338.7721 14.35197492346795
IRIDIUM 33 DEB
1 35054U 97051PA 15334.75622942 .00008389 00000-0 15816-2 0 9993
2 35054 86.3532 144.1628 0007176 189.1949 170.9133 14.64413360348904
IRIDIUM 33 DEB
1 35077U 97051PC 15312.94427154 .00022987 00000-0 36196-2 0 9997
2 35077 86.2875 130.8040 0031799 266.0139 93.7442 14.72036066344388
IRIDIUM 33 DEB
1 35078U 97051PD 15334.26120576 .00032948 00000-0 42041-2 0 9999
2 35078 86.3911 153.1397 0009635 126.5331   0.5161 14.81282845345778
IRIDIUM 33 DEB
1 35079U 97051PE 15334.94876625 .00007109 00000-0 33779-2 0 9997
2 35079 86.4284 223.0200 0151031 318.8542 40.1331 14.13537762341086
IRIDIUM 33 DEB
1 35080U 97051PF 15335.36526800 .00003231 00000-0 15989-2 0 9998
2 35080 86.2485 182.2081 0186357   6.9737 115.6207 14.08442581340348
IRIDIUM 33 DEB
1 35293U 97051PH 15334.64299690 .00024707 00000-0 27081-2 0 9995
2 35293 86.3167 104.1205 0026390 98.5545 261.8672 14.87372643349939
IRIDIUM 33 DEB
1 35294U 97051PJ 15334.44669755 .00004880 00000-0 12119-2 0 9993
2 35294 86.3294 138.1946 0008739 257.2136 132.3741 14.51653862345578

IRIDIUM 33 DEB
1 35296U 97051PL 15334.22974435 .00003969 00000-0 10816-2 0 9994
2 35296 86.3118 134.8372 0016734 261.2490 98.6818 14.47129797178543
IRIDIUM 33 DEB
1 35297U 97051PM 15334.83818380 .00000913 00000-0 31458-3 0 9990
2 35297 86.3910 165.5322 0015624 279.7875 142.5744 14.34874100177770
IRIDIUM 33 DEB
1 35299U 97051PP 15334.96517537 .00001677 00000-0 68596-3 0 9997
2 35299 86.2971 156.3843 0065003 69.7913 83.4074 14.25526174339785
IRIDIUM 33 DEB
1 35477U 97051PX 15334.88302241 .00005929 00000-0 21254-2 0 9993
2 35477 86.3790 199.8277 0105644 191.0760 168.8109 14.30725606169898
IRIDIUM 33 DEB
1 35479U 97051PZ 15334.38645929 .00006594 00000-0 27557-2 0 9994
2 35479 86.3885 199.8278 0099373 206.8312 152.7720 14.23375659335128
IRIDIUM 33 DEB
1 35480U 97051QA 15334.82106634 .00000545 00000-0 17094-3 0 9996
2 35480 86.3617 150.6238 0015417 320.2147 104.5932 14.38770973345358
IRIDIUM 33 DEB
1 35483U 97051QD 15334.57107340 .00008822 00000-0 27086-2 0 9993
2 35483 86.4324 192.5748 0066320 28.1114 332.3639 14.40257514346074
IRIDIUM 33 DEB
1 35484U 97051QE 15334.74368268 .00002017 00000-0 53434-3 0 9998
2 35484 86.3798 147.8777 0055003 268.8929 90.5972 14.47432875346364
IRIDIUM 33 DEB
1 35487U 97051QH 15335.84724689 .00007140 00000-0 19424-2 0 9998
2 35487 86.3412 166.8339 0058808   8.5107 54.6696 14.46346921343375
IRIDIUM 33 DEB
1 35488U 97051QJ 15335.85364125 .00002286 00000-0 53535-3 0 9993

2 35488 86.3384 123.0918 0041288 240.2856 119.4238 14.53675767348314
IRIDIUM 33 DEB
1 35615U 97051QM 15334.90023916 .00025149 00000-0 27168-2 0 9995
2 35615 86.3658 130.3901 0010514 250.8192 109.1896 14.88181368349225
IRIDIUM 33 DEB
1 35616U 97051QN 15334.74965412 .00002338 00000-0 62605-3 0 9995
2 35616 86.3669 142.7526 0018899 300.5978 59.3361 14.47790708345214
IRIDIUM 33 DEB
1 35617U 97051QP 15334.93446554 .00016717 00000-0 30236-2 0 9998
2 35617 86.5021 201.4688 0051134 303.0322 56.5984 14.65433402131714
IRIDIUM 33 DEB
1 35618U 97051QQ 15333.75764696 .00000703 00000-0 20238-3 0 9998
2 35618 86.3419 137.1963 0031470 307.1376 52.6951 14.43163150342597
IRIDIUM 33 DEB
1 35620U 97051QS 15334.17112028 .00003697 00000-0 21008-2 0 9994
2 35620 86.4007 240.5375 0242922 156.4806 204.7706 13.95890350 84742
IRIDIUM 33 DEB
1 35622U 97051QU 15334.85008704 .00004081 00000-0 12548-2 0 9993
2 35622 86.2248 114.6513 0023147 315.5814 198.3349 14.41220788343627
IRIDIUM 33 DEB
1 35623U 97051QV 15306.49099139 .00061120 00000-0 71692-2 0 9997
2 35623 86.4538 212.2537 0050065 86.4598 274.2350 14.83761274335022
IRIDIUM 33 DEB
1 35624U 97051QW 15334.97287557 .00010251 00000-0 24096-2 0 9990
2 35624 86.3642 167.0841 0056126   4.1438 356.0221 14.53348640346389
IRIDIUM 33 DEB
1 35625U 97051QX 15334.27777987 .00001635 00000-0 81544-3 0 9993
2 35625 86.2001 154.2330 0131354 276.5003 208.6893 14.12007375338761
IRIDIUM 33 DEB

1 35627U 97051QZ 15335.83833735 .00005063 00000-0 14207-2 0 9992
2 35627 86.3610 161.7766 0035692 320.3674 39.4920 14.45480816343089
IRIDIUM 33 DEB
1 35628U 97051RA 15334.24208964 .00001830 00000-0 54503-3 0 9999
2 35628 86.3657 150.0067 0018180 309.8333 50.1267 14.42554676334035
IRIDIUM 33 DEB
1 35629U 97051RB 15335.57445561 .00005580 00000-0 14666-2 0 9994
2 35629 86.3865 183.7932 0089473 65.1674 295.8801 14.46592241337022
IRIDIUM 33 DEB
1 35631U 97051RD 15333.45128185 .00000697 00000-0 20859-3 0 9998
2 35631 86.3683 146.9757 0020031 10.6540 17.3510 14.41383771346138
IRIDIUM 33 DEB
1 35632U 97051RE 15335.36779448 .00001156 00000-0 48385-3 0 9990
2 35632 86.4308 211.2743 0123559 204.6150 154.9112 14.21242814334735
IRIDIUM 33 DEB
1 35678U 97051RG 15334.72899111 .00015043 00000-0 26432-2 0 9997
2 35678 86.3616 140.0302 0005132 200.6453 159.4554 14.67576134341254
IRIDIUM 33 DEB
1 35679U 97051RH 15335.18465064 .00020642 00000-0 33643-2 0 9998
2 35679 86.3274 144.9344 0031274 264.7732 94.9914 14.70554221342050
IRIDIUM 33 DEB
1 35680U 97051RJ 15317.54311640 .00001818 00000-0 10718-2 0 9992
2 35680 86.1150 162.9371 0203940 151.1344 240.3222 13.97512429328462
IRIDIUM 33 DEB
1 35732U 97051RK 15334.20773801 .00005568 00000-0 16497-2 0 9999
2 35732 86.2417 132.5826 0047786 353.5574   6.5006 14.42563902339874
IRIDIUM 33 DEB
1 35735U 97051RN 15334.16766938 .00014604 00000-0 19398-2 0 9994
2 35735 86.3836 131.9792 0015568 179.7225 180.4008 14.79611244344189

IRIDIUM 33 DEB
1 35737U 97051RQ 15335.41478105 .00006682 00000-0 18740-2 0 9998
2 35737 86.4335 192.5952 0055284 38.4244 79.5437 14.45004298346665
IRIDIUM 33 DEB
1 35739U 97051RS 15334.85245335 .00009563 00000-0 23959-2 0 9993
2 35739 86.4062 183.5383 0063941 34.9444 325.5931 14.50075956346628
IRIDIUM 33 DEB
1 35742U 97051RV 15335.89099094 .00006042 00000-0 15629-2 0 9995
2 35742 86.4373 183.9721 0057737 280.6739 78.7967 14.48764045343208
IRIDIUM 33 DEB
1 35744U 97051RX 15334.14678034 .00001575 00000-0 41537-3 0 9995
2 35744 86.3370 131.5925 0032203 279.5346 80.2219 14.48108717342185
IRIDIUM 33 DEB
1 35745U 97051RY 15335.89159253 .00007379 00000-0 22634-2 0 9990
2 35745 86.3688 189.8035 0094808 120.0149 241.0498 14.38968845336764
IRIDIUM 33 DEB
1 35747U 97051SA 15308.88300098 .00007266 00000-0 37596-2 0 9994
2 35747 86.1491 165.0037 0183791 112.1387 249.9414 14.06488637322716
IRIDIUM 33 DEB
1 35748U 97051SB 15334.19833133 .00028718 00000-0 30922-2 0 9990
2 35748 86.3804 144.0321 0002258 282.7072 77.3901 14.88337819338056
IRIDIUM 33 DEB
1 35749U 97051SC 15334.94116382 .00003721 00000-0 95921-3 0 9992
2 35749 86.3793 152.9573 0018781 154.9626 205.2494 14.49735575325462
IRIDIUM 33 DEB
1 35750U 97051SD 15334.94630323 .00122002 00000-0 54021-2 0 9999
2 35750 86.0837 47.0555 0003134 210.3521 149.7541 15.21116892338387
IRIDIUM 33 DEB
1 35797U 97051SF 15334.16028909 .00001962 00000-0 51877-3 0 9993

2 35797 86.3483 136.6999 0023791 290.2800 69.5846 14.48248633340380
IRIDIUM 33 DEB
1 35799U 97051SH 15334.89990779 .00007082 00000-0 18817-2 0 9993
2 35799 85.9395 37.9731 0083109 271.6884 87.4800 14.46450822214753
IRIDIUM 33 DEB
1 35800U 97051SJ 15335.12970088 .00016364 00000-0 20660-2 0 9996
2 35800 86.2690 95.1726 0033810 125.5784 359.6925 14.81410599348697
IRIDIUM 33 DEB
1 35802U 97051SL 15335.43258310 .00003430 00000-0 11498-2 0 9998
2 35802 86.4653 207.0337 0076176 91.0659 33.3921 14.35411834347257
IRIDIUM 33 DEB
1 35805U 97051SP 15335.87593707 .00001240 00000-0 49919-3 0 9998
2 35805 86.3301 182.5256 0130498 185.1548 174.8308 14.22795123330452
IRIDIUM 33 DEB
1 35806U 97051SQ 15334.22709006 .00001607 00000-0 53016-3 0 9998
2 35806 86.3503 152.8694 0016257 291.3904 68.5559 14.37466364334982
IRIDIUM 33 DEB
1 35809U 97051ST 15335.20186491 .00001950 00000-0 54634-3 0 9995
2 35809 86.3383 137.3742 0022891 296.2166 187.4116 14.45499237337086
IRIDIUM 33 DEB
1 35844U 97051SU 15334.68634720 .00025338 00000-0 34411-2 0 9994
2 35844 86.3154 118.7204 0018265 291.0130 68.9136 14.78607811336394
IRIDIUM 33 DEB
1 35846U 97051SW 15334.80380912 .00000735 00000-0 25589-3 0 9992
2 35846 86.3806 163.3976 0017794 289.0785 70.8483 14.34026724331315
IRIDIUM 33 DEB
1 35848U 97051SY 15334.77773663 .00002211 00000-0 60681-3 0 9992
2 35848 86.4012 154.4471 0015042 308.6582 51.3274 14.46626046330514
IRIDIUM 33 DEB

1 35850U 97051TA 15334.75182308 .00002257 00000-0 61101-3 0 9991
2 35850 86.3603 141.6967 0018668 300.9755 58.9613 14.47252579181144
IRIDIUM 33 DEB
1 35851U 97051TB 15335.55211280 .00008420 00000-0 21636-2 0 9999
2 35851 86.4188 184.6714 0063060 353.7125   6.3280 14.48914835347180
IRIDIUM 33 DEB
1 35853U 97051TD 15334.78647305 .00009462 00000-0 26778-2 0 9995
2 35853 86.3072 157.3859 0076602   4.9588 355.2353 14.43768173327872
IRIDIUM 33 DEB
1 35856U 97051TG 15335.77062648 .00013644 00000-0 28149-2 0 9990
2 35856 86.3791 149.6445 0036999 199.2464 217.5998 14.59915925333085
IRIDIUM 33 DEB
1 35857U 97051TH 15334.48487866 .00007559 00000-0 15998-2 0 9997
2 35857 86.4337 156.3466 0035840 271.9222 87.7883 14.58762255173048
IRIDIUM 33 DEB
1 35858U 97051TJ 15334.96516275 .00010695 00000-0 28101-2 0 9993
2 35858 86.3643 165.4004 0045251 308.1841 51.5290 14.48449431339271
IRIDIUM 33 DEB
1 35862U 97051TN 15334.47309037 .00013604 00000-0 24628-2 0 9993
2 35862 86.3724 149.1871 0018739 176.1656 213.5749 14.66142495336989
IRIDIUM 33 DEB
1 35863U 97051TP 15334.85569740 .00007206 00000-0 22894-2 0 9999
2 35863 86.3505 170.6596 0061421 56.9687 359.4998 14.38751579333447
IRIDIUM 33 DEB
1 35910U 97051TT 15335.19259522 .00127748 00000-0 58296-2 0 9993
2 35910 86.3750 138.0139 0015177 313.1623 172.2561 15.19962863336493
IRIDIUM 33 DEB
1 35911U 97051TU 15334.46297297 .00003771 00000-0 15325-2 0 9999
2 35911 86.2372 150.7613 0097249 149.3195 211.3733 14.24802707343897

IRIDIUM 33 DEB
1 35915U 97051TY 15334.75840032 .00002363 00000-0 64806-3 0 9991
2 35915 86.3566 141.6384 0022997 292.0603 67.8157 14.46599973332242
IRIDIUM 33 DEB
1 35917U 97051UA 15333.77830786 .00004321 00000-0 95034-3 0 9992
2 35917 86.3861 149.7710 0017136 226.2439 133.7352 14.57222398331536
IRIDIUM 33 DEB
1 35918U 97051UB 15334.33824168 .00001343 00000-0 46844-3 0 9999
2 35918 86.4097 174.0259 0031149 334.3194 150.3874 14.34396035325157
IRIDIUM 33 DEB
1 35921U 97051UE 15334.42475959 .00004319 00000-0 18837-2 0 9999
2 35921 86.4218 220.8297 0156462 287.7800 195.6651 14.17428965129201
IRIDIUM 33 DEB
1 35922U 97051UF 15334.79512634 .00004270 00000-0 10867-2 0 9993
2 35922 86.3930 153.8644 0030748 268.7204 91.0478 14.50204735331360
IRIDIUM 33 DEB
1 35925U 97051UJ 15334.88382125 .00011974 00000-0 22067-2 0 9990
2 35925 86.3234 134.4888 0014853 132.0843 228.1635 14.65388969344490
IRIDIUM 33 DEB
1 35926U 97051UK 15335.19789600 .00004922 00000-0 11269-2 0 9992
2 35926 86.3492 136.8578 0009805 300.1236 59.8999 14.55455605337387
IRIDIUM 33 DEB
1 35929U 97051UN 15334.19118249 .00002377 00000-0 57756-3 0 9993
2 35929 86.3521 132.7460 0030470 262.1071 97.6675 14.52210878343368
IRIDIUM 33 DEB
1 36011U 97051US 15334.75655184 .00000743 00000-0 26089-3 0 9992
2 36011 86.3442 153.2382 0019464 321.5948 99.4078 14.33609526328672
IRIDIUM 33 DEB
1 36012U 97051UT 15335.78839720 .00000954 00000-0 62365-3 0 9998

2 36012 86.4600 275.0709 0289082 263.6872 93.1302 13.83050734326790
IRIDIUM 33 DEB
1 36017U 97051UY 15335.24427406 .00010470 00000-0 17824-2 0 9991
2 36017 86.3706 143.0939 0019811 144.5874 337.4594 14.68852501330776
IRIDIUM 33 DEB
1 36019U 97051VA 15334.95143745 .00003256 00000-0 81081-3 0 9992
2 36019 86.3962 152.1963 0008695   3.8564 148.6474 14.51440237330142
IRIDIUM 33 DEB
1 36021U 97051VC 15334.62644112 .00004600 00000-0 17833-2 0 9990
2 36021 86.4243 209.5813 0122421 190.8287 169.0267 14.25770059 58606
IRIDIUM 33 DEB
1 36023U 97051VE 15334.93300424 .00005102 00000-0 13684-2 0 9999
2 36023 86.3457 150.4661 0026319 273.0034 86.8157 14.47820037140424
IRIDIUM 33 DEB
1 36025U 97051VG 15334.47509108 .00008987 00000-0 24080-2 0 9996
2 36025 86.5644 232.0905 0080975 29.1273 331.4410 14.46164776330508
IRIDIUM 33 DEB
1 36026U 97051VH 15312.24523479 .05713706 00000-0 21771-1 0 9995
2 36026 86.3748 137.5676 0009867 331.7903 29.4083 15.83165965337554
IRIDIUM 33 DEB
1 36028U 97051VK 15334.19325336 .00002316 00000-0 57002-3 0 9998
2 36028 86.3384 129.2740 0033523 258.6311 101.1127 14.51535544336620
IRIDIUM 33 DEB
1 36081U 97051VN 15334.20780107 .00004783 00000-0 15507-2 0 9993
2 36081 86.2402 137.9512 0067386 54.6727 306.0745 14.37484450 94054
IRIDIUM 33 DEB
1 36083U 97051VQ 15334.23140001 .00000771 00000-0 26657-3 0 9993
2 36083 86.3467 153.5061 0029432 345.9809 14.0566 14.34294732329370
IRIDIUM 33 DEB

1 36390U 97051VU 15334.12074794 .00001816 00000-0 47333-3 0 9992
2 36390 86.3039 122.2132 0044590 258.9866 100.6321 14.48461495350772
IRIDIUM 33 DEB
1 36483U 97051VZ 15335.80338980 .00000738 00000-0 24702-3 0 9998
2 36483 86.3839 160.4418 0007967 280.8757 79.1543 14.36105553332104
IRIDIUM 33 DEB
1 36485U 97051WB 15334.12446044 .00001777 00000-0 71395-3 0 9999
2 36485 86.4442 219.6761 0136370 233.0774 125.7861 14.22748559311714
IRIDIUM 33 DEB
1 36486U 97051WC 15335.16436209 .00005900 00000-0 13024-2 0 9993
2 36486 86.3576 138.4553 0014478 244.3870 115.5843 14.57135626351203
IRIDIUM 33 DEB
1 36487U 97051WD 15334.39504421 .00004208 00000-0 21843-2 0 9998
2 36487 86.3850 222.5612 0182081 30.9026 330.2717 14.06403550311792
IRIDIUM 33 DEB
1 36488U 97051WE 15334.29284610 .00029110 00000-0 38511-2 0 9992
2 36488 86.4413 167.9468 0017432 131.8284 228.4429 14.79707688335449
IRIDIUM 33 DEB
1 36489U 97051WF 15317.88676037 .00015712 00000-0 26611-2 0 9998
2 36489 86.4286 169.6444 0011573 250.5353 154.2727 14.69165684342940
IRIDIUM 33 DEB
1 36490U 97051WG 15334.77168304 .00002078 00000-0 60238-3 0 9995
2 36490 86.3787 152.2525 0003139 67.4068 292.7465 14.44013824333568
IRIDIUM 33 DEB
1 36491U 97051WH 15335.21004242 .00052750 00000-0 59549-2 0 9990
2 36491 86.3152 145.9994 0038810 197.2881 162.7025 14.85738751336416
IRIDIUM 33 DEB
1 36492U 97051WJ 15334.97861756 .00004765 00000-0 11853-2 0 9993
2 36492 86.4289 167.3358 0014594 157.0270 203.1591 14.51528714341839

IRIDIUM 33 DEB
1 36493U 97051WK 15334.26006175 .00001741 00000-0 52460-3 0 9993
2 36493 86.3656 150.4249 0015380 310.9897 48.9972 14.41987009340986
IRIDIUM 33 DEB
1 36495U 97051WM 15335.76213371 .00016753 00000-0 26945-2 0 9996
2 36495 86.4050 161.2402 0015886 190.9989 169.0882 14.71391684315749
IRIDIUM 33 DEB
1 36497U 97051WP 15335.14077726 .00001567 00000-0 53222-3 0 9998
2 36497 86.1600 102.9059 0073320 289.6663 191.1773 14.34589630333699
IRIDIUM 33 DEB
1 36563U 97051WS 15326.82146184 .00014927 00000-0 29255-2 0 9994
2 36563 86.3720 149.6407 0020785 304.0164 55.9030 14.62548911330359
IRIDIUM 33 DEB
1 36566U 97051WV 15334.48242470 .00007800 00000-0 38065-2 0 9994
2 36566 86.4304 237.1689 0185107 34.6497 326.6581 14.09399040326050
IRIDIUM 33 DEB
1 36568U 97051WX 15333.83357029 .00045983 00000-0 41826-2 0 9994
2 36568 86.2951 106.0869 0020215 16.8421 343.3475 14.94843725329711
IRIDIUM 33 DEB
1 36642U 97051XC 15334.78860166 .00010535 00000-0 23551-2 0 9993
2 36642 86.3760 153.7529 0023390 167.2009 192.9798 14.56489839332849
IRIDIUM 33 DEB
1 37136U 97051XG 15334.19973156 .00000869 00000-0 39024-3 0 9993
2 37136 86.1753 130.4637 0087289 150.5746 210.0398 14.19272151337401
IRIDIUM 33 DEB
1 37548U 97051XH 15334.77001810 .00001371 00000-0 48592-3 0 9992
2 37548 86.3590 161.1384 0030206 336.2985 23.6817 14.33611989128701
IRIDIUM 33 DEB
1 37549U 97051XJ 15335.29967565 .00007099 00000-0 18893-2 0 9992

2 37549 86.3902 173.9377 0048438 311.9544 47.7536 14.47727974331820
IRIDIUM 33 DEB
1 37550U 97051XK 15334.27390792 .00001796 00000-0 53528-3 0 9996
2 37550 86.3862 155.8203 0003694 101.1111 259.0505 14.42602234332896
IRIDIUM 33 DEB
1 37554U 97051XP 15334.19743010 .00033970 00000-0 23578-2 0 9996
2 37554 86.3744 134.6430 0029862 241.5349 118.2877 15.05057928189008
IRIDIUM 33 DEB
1 37555U 97051XQ 15334.31531724 .00003610 00000-0 92640-3 0 9993
2 37555 86.4685 192.2697 0042409 326.4515 33.4004 14.49515520140303
IRIDIUM 33 DEB
1 37557U 97051XS 15334.38980560 .00004307 00000-0 17869-2 0 9991
2 37557 86.3855 209.1860 0126271 274.0732 84.6033 14.22115014326022
IRIDIUM 33 DEB
1 37558U 97051XT 15335.82503383 .00004184 00000-0 11677-2 0 9990
2 37558 86.4050 171.8908 0089595   4.8981 355.3074 14.43704982302478
IRIDIUM 33 DEB
1 37560U 97051XV 15334.40540659 .00005938 00000-0 21120-2 0 9995
2 37560 86.3937 199.1250 0103977 169.6584 315.8139 14.31209096304093
IRIDIUM 33 DEB
1 37562U 97051XX 15334.25951480 .00001501 00000-0 49293-3 0 9992
2 37562 86.3825 161.9070 0030512 331.4469 151.8170 14.37452636335096
IRIDIUM 33 DEB
1 37564U 97051XZ 15333.73669130 .00052975 00000-0 40571-2 0 9994
2 37564 86.3678 136.6743 0008601 350.3252 68.3762 15.01597339322361
IRIDIUM 33 DEB
1 37565U 97051YA 15334.20854432 .00000832 00000-0 25156-3 0 9997
2 37565 86.3529 144.4469 0014087 343.7710 139.5701 14.41179584129505
IRIDIUM 33 DEB

1 37566U 97051YB 15335.54746557 .00012040 00000-0 17758-2 0 9996
2 37566 86.4505 178.3004 0029230 308.7183 51.1423 14.74939784309066
IRIDIUM 33 DEB
1 38016U 97051YC 15334.30618348 .00016284 00000-0 32595-2 0 9996
2 38016 86.4055 174.6654 0040378 291.1351 193.6386 14.61203860100744
IRIDIUM 33 DEB
1 38017U 97051YD 15335.65216854 .00005981 00000-0 33978-2 0 9990
2 38017 86.3524 220.8709 0201624 85.2169 277.1998 14.00036824253554
IRIDIUM 33 DEB
1 38019U 97051YF 15334.38550612 .00007819 00000-0 24692-2 0 9999
2 38019 86.4022 197.1077 0092487 133.6713 227.2199 14.37691219299972
IRIDIUM 33 DEB
1 38020U 97051YG 15334.94818868 .00005611 00000-0 26388-2 0 9995
2 38020 86.3171 205.2556 0193643 37.3302 324.1155 14.10475226320299
IRIDIUM 33 DEB
1 38022U 97051YJ 15335.23413773 .00001158 00000-0 46190-3 0 9992
2 38022 86.2797 148.7318 0058866 33.5309 326.9586 14.26788085346230
IRIDIUM 33 DEB
1 38023U 97051YK 15334.39935651 .00007005 00000-0 24614-2 0 9996
2 38023 86.4220 213.0038 0107632 207.8278 151.7135 14.31617389315847
IRIDIUM 33 DEB
1 38024U 97051YL 15335.44951647 .00003715 00000-0 14760-2 0 9992
2 38024 86.4822 226.7566 0119834 221.6687 137.5329 14.24668792325101
IRIDIUM 33 DEB
1 38025U 97051YM 15335.08764071 .00007791 00000-0 35635-2 0 9998
2 38025 86.4945 248.4375 0186745   7.8654 50.1364 14.12546800339612
IRIDIUM 33 DEB
1 38028U 97051YQ 15330.80800329 .00000709 00000-0 22133-3 0 9995
2 38028 86.3275 140.3347 0028756 342.0605 17.9576 14.39249345126585

IRIDIUM 33 DEB
1 38030U 97051YS 15334.90329610 .00008712 00000-0 24244-2 0 9998
2 38030 86.3753 189.5603 0088481 101.6108 259.5041 14.43985581251671
IRIDIUM 33 DEB
1 38033U 97051YV 15334.08993665 .00012578 00000-0 18114-2 0 9991
2 38033 86.2809 102.5400 0026134 173.8458 186.3088 14.76020311710878
IRIDIUM 33 DEB
1 38034U 97051YW 15335.16173904 .00047900 00000-0 37544-2 0 9993
2 38034 86.3395 117.9632 0012196 219.9146 264.4487 15.00708761320942
IRIDIUM 33 DEB
1 38224U 97051YY 15328.90288870 .00004940 00000-0 17902-2 0 9990
2 38224 86.3909 203.3095 0098936 210.4721 149.0702 14.30542125317449
IRIDIUM 33 DEB
1 38225U 97051YZ 15334.96331227 .00001971 00000-0 10979-2 0 9997
2 38225 86.3621 225.9956 0196122 103.7675 316.8294 14.01213027106423
IRIDIUM 33 DEB
1 38226U 97051ZA 15333.50201508 .00008782 00000-0 23131-2 0 9996
2 38226 86.3834 159.6501 0007098 292.1555 67.8895 14.48930740331948
IRIDIUM 33 DEB
1 38227U 97051ZB 15333.72831353 .00003163 00000-0 83894-3 0 9999
2 38227 86.3147 132.3232 0007727 291.3259 68.7119 14.48435807329968
IRIDIUM 33 DEB
1 38228U 97051ZC 15334.85477006 .00000400 00000-0 16604-3 0 9992
2 38228 86.2968 156.6669 0068308 54.9127 11.3598 14.22621076326308
IRIDIUM 33 DEB
1 38229U 97051ZD 15331.54156091 .00002622 00000-0 16583-2 0 9997
2 38229 86.3822 245.5328 0229472 191.8473 167.7249 13.91613218303314
IRIDIUM 33 DEB
1 38231U 97051ZF 15334.19501450 .00013829 00000-0 26848-2 0 9993

2 38231 86.2490 130.1131 0049503 294.0306 190.6537 14.62324308332623
IRIDIUM 33 DEB
1 38232U 97051ZG 15325.98576742 .00007655 00000-0 25536-2 0 9990
2 38232 86.4600 220.4893 0098173 175.4175 184.7944 14.34710819334588
IRIDIUM 33 DEB
1 38233U 97051ZH 15334.33548876 .00004888 00000-0 23382-2 0 9999
2 38233 86.2267 178.0757 0169802   2.2807 119.9634 14.11675090135637
IRIDIUM 33 DEB
1 38234U 97051ZJ 15334.34211004 .00010404 00000-0 24988-2 0 9995
2 38234 86.4593 188.9810 0059285 277.0678 82.3788 14.52237471313647
IRIDIUM 33 DEB
1 38235U 97051ZK 15335.26449345 .00007767 00000-0 15397-2 0 9991
2 38235 86.3957 157.4005 0009235 259.4305 100.5866 14.62129400717066
IRIDIUM 33 DEB
1 38236U 97051ZL 15332.26684734 .00002587 00000-0 92485-3 0 9991
2 38236 86.2543 144.6672 0077252 63.9557 296.9565 14.32082973256655
IRIDIUM 33 DEB
1 38237U 97051ZM 15334.95388852 .00012418 00000-0 22651-2 0 9993
2 38237 86.3848 153.5811 0014093 192.1347 319.6767 14.65856745 94957
IRIDIUM 33 DEB
1 38241U 97051ZR 15334.19758821 .00001135 00000-0 34615-3 0 9999
2 38241 86.3266 138.7343 0003430   3.9460 356.1766 14.41156146309483
IRIDIUM 33 DEB
1 38243U 97051ZT 15327.26982230 .00008775 00000-0 21411-2 0 9990
2 38243 86.3817 156.2648 0035884 272.4408 87.2690 14.52174914267131
IRIDIUM 33 DEB
1 38244U 97051ZU 15335.82794166 .00010802 00000-0 16683-2 0 9990
2 38244 86.4390 160.5555 0010509 11.6931 348.4529 14.73206441309140
IRIDIUM 33 DEB

1 38468U 97051ZV 15316.94599319 .00003586 00000-0 13505-2 0 9994
2 38468 86.3606 188.8846 0092988 187.5107 172.4701 14.28848250290047
IRIDIUM 33 DEB
1 38469U 97051ZW 15334.88750063 .00004619 00000-0 21673-2 0 9996
2 38469 86.2820 194.0283 0193743 327.9890 30.9639 14.10576489 89440
IRIDIUM 33 DEB
1 38470U 97051ZX 15325.05779203 .00004806 00000-0 17452-2 0 9992
2 38470 86.4956 229.7060 0108517 179.6742 245.4319 14.29884292343407
IRIDIUM 33 DEB
1 38471U 97051ZY 15306.82844634 .00015412 00000-0 24560-2 0 9991
2 38471 86.2956 134.1384 0023058 305.2153 62.3627 14.71703926 90669
IRIDIUM 33 DEB
1 38472U 97051ZZ 15334.28253853 .00042864 00000-0 47920-2 0 9996
2 38472 86.4090 156.7990 0012032 157.8891 328.2773 14.86686981309615
IRIDIUM 33 DEB
1 38474U 97051AAB 15334.72027894 .00001006 00000-0 31312-3 0 9993
2 38474 86.3460 144.4566 0029059 320.3267 39.5805 14.39834710335521
IRIDIUM 33 DEB
1 38477U 97051AAE 15335.73068508 .00002579 00000-0 25477-2 0 9995
2 38477 86.4232 335.7475 0483472 91.9666 273.6896 13.36252285275108
IRIDIUM 33 DEB
1 39777U 97051AAJ 15333.71667504 .00002242 00000-0 93171-3 0 9990
2 39777 85.9835 76.5946 0097397 140.4549 220.3800 14.23476200 77673
IRIDIUM 33 DEB
1 39778U 97051AAK 15334.28064762 .00001396 00000-0 45290-3 0 9999
2 39778 86.3846 160.4411 0008631 241.0519 118.9814 14.38267377 78700
IRIDIUM 33 DEB
1 39779U 97051AAL 15335.25298904 .00002590 00000-0 92830-3 0 9994
2 39779 86.2954 154.9158 0073095 35.4545 87.8643 14.32133012183383

IRIDIUM 33 DEB
1 39781U 97051AAN 15333.11808139 .00002395 00000-0 15035-2 0 9991
2 39781 86.3782 258.7803 0299471 308.9270 111.8559 13.84403145211623
IRIDIUM 33 DEB
1 39782U 97051AAP 15335.27611260 .00006366 00000-0 12632-2 0 9990
2 39782 86.4179 155.6961 0014615 259.6327 226.4897 14.62024623203991
IRIDIUM 33 DEB
1 39783U 97051AAQ 15334.68452637 .00003324 00000-0 11473-2 0 9999
2 39783 86.1727 116.8869 0072857 72.5383 344.4257 14.34088680 78320
IRIDIUM 33 DEB
1 39785U 97051AAS 15332.83837162 .00024670 00000-0 38158-2 0 9992
2 39785 86.4017 165.7330 0033404 230.5758 129.9576 14.72766906201740
IRIDIUM 33 DEB
1 39786U 97051AAT 15334.77556862 .00009474 00000-0 20095-2 0 9999
2 39786 86.3758 149.0996 0008817 163.0642 257.2726 14.59059193183432
IRIDIUM 33 DEB
1 39787U 97051AAU 15335.14649686 .00004882 00000-0 13480-2 0 9997
2 39787 86.2104 111.0536 0042410 329.2389 155.0391 14.46082284200196
IRIDIUM 33 DEB
1 39788U 97051AAV 15315.48974015 .00036102 00000-0 84703-2 0 9996
2 39788 86.3323 221.3374 0121906 74.1532 289.0677 14.49819299252931
IRIDIUM 33 DEB
1 39790U 97051AAX 15335.59163972 .00007741 00000-0 18502-2 0 9996
2 39790 86.1843 94.1084 0043542 289.3625 70.2876 14.52943832350727
IRIDIUM 33 DEB
1 39792U 97051AAZ 15335.01539859 .00004247 00000-0 24318-2 0 9996
2 39792 86.4333 244.0569 0188937 82.3064 279.9522 14.00688278 90333
IRIDIUM 33 DEB
1 39793U 97051ABA 15334.90144763 .00032466 00000-0 38085-2 0 9990

2 39793 86.3696 138.2826 0003444 160.7726 199.3628 14.84810212 80462
IRIDIUM 33 DEB
1 39796U 97051ABD 15335.24998509 .00004271 00000-0 12134-2 0 9996
2 39796 86.3803 159.7577 0074057 345.5856 14.1960 14.43604591520975
IRIDIUM 33 DEB
1 40992U 97051ABE 15334.19733192 .00005507 00000-0 12909-2 0 9997
2 40992 86.3059 130.1326 0024887 269.4331 215.5776 14.54206408199107
IRIDIUM 33 DEB
1 40993U 97051ABF 15335.27008367 .00007554 00000-0 17088-2 0 9994
2 40993 86.3813 155.9486 0018973 221.3951 262.3293 14.55984925177133
IRIDIUM 33 DEB
1 40994U 97051ABG 15334.19764339 .00005988 00000-0 12028-2 0 9991
2 40994 86.3949 146.5405 0008543 277.5393 82.4848 14.61506668 61989
IRIDIUM 33 DEB
1 40995U 97051ABH 15330.82385565 .00002960 00000-0 88375-3 0 9994
2 40995 86.3926 168.8045 0041502 342.0328 73.4982 14.42229270177325
IRIDIUM 33 DEB
1 40996U 97051ABJ 15334.96125703 .00002509 00000-0 71893-3 0 9996
2 40996 86.3757 152.7383 0012392 136.4926 17.2175 14.44613362 92573
IRIDIUM 33 DEB
1 40997U 97051ABK 15335.23180157 .00017230 00000-0 26805-2 0 9991
2 40997 86.3666 145.5364 0012247 127.8990 232.3337 14.72874609230394
IRIDIUM 33 DEB
1 40998U 97051ABL 15334.81558135 .00000814 00000-0 27103-3 0 9993
2 40998 86.3865 161.0713 0007060 228.8888 188.3740 14.36438304544194

VITA

Timothy S. Turk was born in Bethesda, MD to Cecil “Wayne” and Beverly Turk. He is the first of four children: Rhonda, Chris and Shari. He attended several elementary and junior high schools in Maryland, Illinois and California, and finally graduated from Royal High School in Simi Valley, CA in 1982. After high school, he enlisted in the United States Navy, where he served for 25 years. He is a retired Naval Officer and was a direct commission officer from the enlisted rank of Electronics Technician, Chief Petty Officer. It was while on active duty in the Navy that Mr. Turk became interested in space operations. Mr. Turk earned numerous personal awards while on active duty and in subsequent government service, to include the Defense Meritorious Service Medal, Navy and Marine Corps Commendation Medal (six awards), Joint Service Achievement Medal (two awards), Navy and Marine Corps Achievement Medal (three awards), Good Conduct Medal (three awards), and various campaign, unit and deployment awards. In addition, he was selected, from a field of over 10,000 Sailors, as Commander, Anti-Submarine

Warfare Command Sailor of the Year in 1991. He was also selected as the 1991 Naval Air Station North Island Sailor of the Year and Air Traffic Control Technician of the Year in 1991. Mr. Turk served in government service for approximately three years following his Naval career and was selected as United States Cyber Command Civilian of the Quarter in 2012. He is presently employed by Booz, Allen Hamilton as lead Commercial Satellite Liaison. He has earned a Bachelor of Science in Electronics Engineering from Thomas Edison State University in Trenton, New Jersey (2005), a Master of Science in Space Systems Management and Master of Business Administration, both from Webster University in Colorado Springs, Colorado (2009 and 2011, respectively). Mr. Turk has been married to his high school sweetheart, Jeanne, since October 2nd, 1983 and they have been blessed with two children and seven grandchildren.