6-2022

Scheduling for Space Tracking and Heterogeneous Sensor Environments

Gabriel H. Greve

SCHEDULING FOR SPACE TRACKING
AND HETEROGENEOUS SENSOR ENVIRONMENTS

DISSERTATION

Gabriel H. Greve, CTR, USAF

AFIT-ENG-DS-22-J-085

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright Patterson Air Force Base, Ohio

AFIT-ENG-DS-22-J-085

SCHEDULING FOR SPACE TRACKING AND HETEROGENEOUS SENSOR ENVIRONMENTS

DISSERTATION

Presented to the Faculty of the
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Gabriel H. Greve, B.S.C.S., M.S.C.S.
CTR, USAF

June, 2022

AFIT Graduate School of Engineering and Management
Thesis / Dissertation Committee Approval Signature Page

Designator Number: AFIT-ENG-DS-22-J-085

Title: **Scheduling for Space Tracking and Heterogeneous Sensor Environments**

Student's Name and Rank: CTR, USAF Gabriel H. Greve

Approved:

Committee Chair Signature	HOPKINSON.KENNETH.M.1275914858 Digitally signed by HOPKINSON.KENNETH.M.1275914858 Date: 2022.05.26 12:38:29 -04'00'	Date	May 26, 2022
Committee Member Signature	LAMONT.GARY.B.1231194467 Digitally signed by LAMONT.GARY.B.1231194467 Date: 2022.05.26 16:52:58 -04'00'	Date	May 26, 2022
Committee Member Signature	OXLEY.MARK.E.1230244746 Digitally signed by OXLEY.MARK.E.1230244746 Date: 2022.05.26 13:24:33 -04'00'	Date	May 26, 2022
Committee Member Signature		Date	
Committee Member Signature		Date	

For Dissertations Only

Dean's Acceptance: BADIRU.ADEDEJI.B.1293329580 Digitally signed by BADIRU.ADEDEJI.B.1293329580 Date: 2022.05.27 09:34:44 -04'00' — Date: May 27, 2022

ADEDEJI B. BADIRU, Ph.D.
Dean, Graduate School of Engineering and Management

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

List of Abbreviations

AFIT-ENG-DS-22-J-085

Abstract

This dissertation draws on the fields of heuristic and metaheuristic algorithm development, resource allocation problems, and scheduling to address key Air Force problems. The world runs on many schedules. People depend upon them and expect these schedules to be accurate. A process is needed where schedules can be dynamically adjusted to allow tasks to be completed efficiently. For example, the Space Surveillance Network relies on a schedule to track objects in space. The schedule must use sensor resources to track as many high-priority satellites as possible to obtain orbit paths and to warn of collision paths. Any collisions that occurred between satellites and other orbiting material could be catastrophic. To address this critical problem domain, this dissertation introduces both a single objective evolutionary tasker algorithm and a multi-objective evolutionary algorithm approach. The aim of both methods is to produce space object tracking schedules to ensure that higher priority objects are appropriately assessed for potential problems. Simulations show that these evolutionary algorithm techniques effectively create schedules to assure that higher priority space objects are tracked. These algorithms have application to a range of dynamic scheduling domains including space object tracking, disaster search and rescue, and heterogeneous sensor scheduling.

SCHEDULING FOR SPACE TRACKING AND HETEROGENEOUS SENSOR ENVIRONMENTS

1. *Introduction*

The way missions are conducted for our military depends on space. Satellites are used to communicate with troops, gather intelligence, fly drones, target weapons, etc. These satellites are not only vulnerable to attack, but also vulnerable to seemingly benign and/or small space debris [73]. When considering that the speed these objects can reach is 20,000 kilometers per hour or more, the potential destructive energy of even a very small object becomes very significant.

If a small object can be destructive, consider the old rocket bodies or defunct satellites of a greatly larger size that can cause catastrophic results. For example, on October 15, 2020, satellite tracking experts doing conjunction assessment of a defunct Russian satellite and a discarded Chinese rocket body watched their orbits at about 615 miles above Earth. They narrowly missed each other by 11 meters [64]. This is an example of Resident Space Objects (RSO) that, if they collide, could start a chain reaction known as the Kessler Syndrome [58]. The Kessler Syndrome is a concept of colossal cascading collisions of RSO in which the unstable condition of orbiting objects will eventually collide and break up into smaller pieces; therefore, increasing the collision rate. This scenario could result in a debris field so wide that it could inhibit space travel and block sunlight.

The field of scheduling is well studied; yet, in specific areas like satellite tracking and heterogeneous sensor environments are parts of continued interest for novel research [7][9][75]. These are the areas that are the focus of this dissertation, which will draw on the fields of heuristic and metaheuristic algorithm development, resource allocation problems, and scheduling to address key Air Force problems. The world runs on many schedules. People depend upon them and expect these schedules to be accurate. A process is needed where schedules can be dynamically adjusted to allow tasks to be completed efficiently. For example, the Space Surveillance Network relies on a schedule to track objects in space.

The schedule must use sensor resources to track as many high-priority satellites as possible to obtain orbit paths and to warn of collision paths. Any collisions that occurred between satellites and other orbiting material could be catastrophic. To address this critical problem domain, this dissertation introduces both a single objective evolutionary tasker algorithm and a multi-objective evolutionary algorithm approach. The aim of both methods is to produce space object tracking schedules to ensure that higher priority objects are appropriately assessed for potential problems. Simulations show that these evolutionary algorithm techniques effectively create schedules to ensure that higher priority space objects are tracked. These algorithms have application to a range of dynamic scheduling domains, including space object tracking, disaster search and rescue, and heterogeneous sensor scheduling.

1.1 Technical Motivation

A 24-hour schedule for sensors tracking satellites is created daily by the SP Tasker algorithm [8]. SP stands for special perturbations. Miller's article analyzes the SP Tasker algorithm [75]. Although the results for SP Tasker show that it does well on its evaluation metrics (unique track percentage and track response rate), it treats higher priority satellites the same as other satellites, like space junk. Moreover, it takes most of a 24-hour cycle to prepare a schedule for the next day. This dissertation further analyzes the algorithm and presents some novel alternative techniques that perform better with key measurable parameters.

As part of the Space Surveillance mission, the Geosynchronous Space Situational Awareness Program satellites will be able to collect pertinent information for more accurate tracking and characterization of man-made RSO [39]. These objects are launched by at least 11 countries. Of these countries combined, it is estimated that only about 23,300 of the 34,000 RSO >10 cm diameter are on orbit right now [17]. Only about 5,600 of these on orbit RSO are active payloads, because many of them are just space junk [38].

1.2 Contributions

The Research Contribution Hierarchy in Figure 1 works from the generic scheduling problem down to more specific problems and solutions that are novel in this dissertation. These are the 5 specific areas that are the major contributions to this research. Also briefly highlighted in this section, there are several minor contributions that are not listed in the hierarchy to avoid it being too cluttered.

The need to effectively catalog and track an increasing portfolio of RSO provides motivation to create more effective schedule algorithms geared towards space tracking using a heterogeneous mixture of sensor environments. This dissertation starts with the current state of scheduling theory and works towards solving specific real-world problems. The research presented in this effort concentrates on two main topics: Resource Allocation Problem (RAP) and the Multi-Objective RAP (MORAP). Specifically, the algorithms developed focus on applications involving the Satellite Sensor Allocation Problem SSAP and the Multi-Objective SSAP (MOSSAP).

The full formal mathematical definition for the SSAP is provided. Previously in the literature, the problem has only been partially defined. The novel Evolutionary Algorithm Tasker (EAT) solves the SSAP. Another contribution is that the EAT adds priority back into the tasking system. It does this by using the priority objective defined later. EAT also performs significantly better than the current system in two out of three key metrics.

The MOSSAP is mathematically defined. It is similar to the SSAP; however, instead of having a single objective that objective is split into competing objectives (probability and priority). The novel Multi-Objective EAT (MEAT) solves the MOSSAP. MEAT does well with the spread metric. Another contribution is the fact that is dissertation explores various MOEAs with respect to MOSSAP and benchmark problems. Also, the design of Chapter 3 allows the user to make the tradeoff between which algorithm to use, and which solution to choose from all the possible solutions provided by software.

Newman et al. [84] wrote an article comparing stochastic optimization approaches for scheduling satellite sensors. Improvements mentioned in Newman's article serve as an inspiration for the work in this dissertation. The algorithm improvements entail contri-

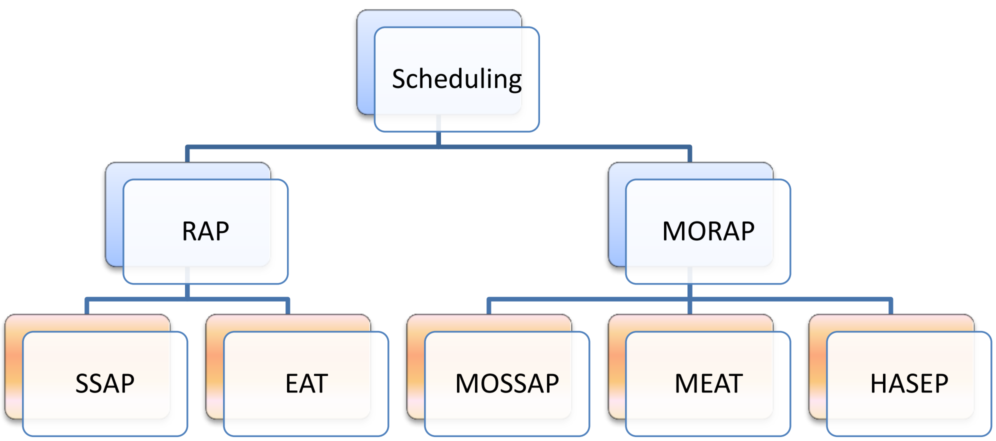

Figure 1 Research Contribution Hierarchy: Starting with a generic scheduling field, the research is narrowed to RAP and MORAP. The work is narrowed further to five specific contribution areas in the bottom row.

butions of single objective evolutionary algorithm (EA) and multi-objective evolutionary algorithm (MOEA) solutions to the SSAP. In this dissertation, new refinements are introduced to solve the SSAP problem more effectively according to key metrics of real-world significance.

Finally, there are a couple more contributions from this dissertation. This dissertation explores various MOEAs and does a scalability analysis. The experimentation also confirms that NSGA-II and SPEA2 do well with scalability. The scalability analysis is an important part of this research, because it is good step toward the overall research goal of solving the Heterogeneous Ariel Sensor Environment Problem (HASEP). The MOEAs are compared based on changing the number of decision variables. An additional contribution is the discussion of the HASEP. There are many differences between the SSAP problem and the HASEP. The objectives are different, and constraints are different.

1.3 Dissertation Outline

Chapter 2 explores the RAP inherent in Space Surveillance Network (SSN), which protects valuable space-based assets. This leads to the first research questions:

> *RQ1*: Will the EAT perform better, across key metrics, than existing algorithms on a full scale SSAP? When compared to the current algorithm that assigns the space tracking system, where does EAT perform better and where does EAT perform worse and by how much?

The second chapter seeks to allocate resources to track satellites to obtain the biggest yield that the present system can handle. This yield is based on three key metrics, which are track response rate, unique-track percentage, and not-tracked percentage. The SSAP is mathematically modeled in such a way that both the control and experiment are on a level playing field. The approaches to the SSAP add priority into the equation so that the most important satellites are tracked more often. The effectiveness of the approach is shown in the good results.

Chapter 3 seeks to build upon the key insights gained from developing the single objective algorithm into a multi-objective algorithm. It does this by splitting the single objective into two competing objectives. Chapter 3 explores the research questions:

RQ2: How well does the MEAT perform in comparison to well-known MOEAs that have a comparable time complexity? If it does well, what metrics did the MEAT perform well?

The third chapter develops the novel MEAT that features a hybrid genetic algorithm approach. Building on the novel algorithm developed in Chapter 3, Chapter 4 changes the application domain from primarily ground-based assets to aerial and space-based assets. The EAT and MEAT are designed and developed with specific evolutionary techniques to perform well on the SSAP. Further experimentation and analysis of the SSAP is conducted with MOEAs using the jMetal software framework [34]. Chapter 4 examines the research questions:

RQ3: Can increasing the decision variables provide good step towards addressing a real-world scaled problem? Which MOEA performs the best overall with the scalability analysis?

The fourth chapter provides key insights into understanding the capabilities of such systems both in the application domain and the computation domain. Three MOEAs are tested against benchmark problems and scaled to find out which performs better in large scale environments. Chapter 5 concludes the dissertation by summarizing the answers to the research questions that have just been outlined.

1.4 Background

This section provides a high-level background covering the most popular approaches to scheduling problems, as shown in Table 1. It shows an abridged summary of the literature reviewed. Much more literature is cited throughout this dissertation. For this list see the bibliography.

In the table there are 9 columns, each of which pertains to a different area of the hierarchy of scheduling approaches in the literature. The authors on the left side indicate publications they have written. The dots show the relationship between the authors' writings and their scheduling approaches. Baker and Pinedo have written books on scheduling, sequencing, and planning [7][90]. These books provide in-depth reference material for the interested reader.

Table 1 Scheduling Literature Overview

	Scheduling	RAP	EA	MOEA	SSAP	Dynamic Scheduling	Ad-hoc Arrival	EA Scheduling	SchedCAT
Baker (2009) [7]	•					•	•		
Brandenberg [13]	•								•
Cerqueira [19]	•								•
Chapman [21]	•								
Choi [24]					•				
Coello [26]			•	•					
Deb [30]			•	•					
Durillo (2011) [34]			•	•					
Elliott [37]	•								•
He (2004) [50]	•								
Honório [52]	•					•			
Kessler [58]					•				
Kiekintveld [59]		•							
Kleeman [60]	•		•	•				•	
Liu [67]			•	•					
Maheswaran [70]	•					•			
Malve [72]	•		•			•	•	•	
Miller (2007) [75]		•			•				
Nebro (2010)[82]			•	•					
Newman [84]	•		•		•				
Osman (2005) [85]		•	•	•					
Petrick [87]					•				
Pinedo (2012) [90]	•					•	•		
Pinedo (2005) [89]	•					•			
Rangsaritratsamee [92]	•					•	•	•	
Reynolds [93]		•							
Singh (2013) [100]	•								
Tan [104]			•	•					
Wang (1999) [110]	•								
Weeden [112]					•				
Wieder [114]	•								•
Zhang [122]			•	•					

1.4.1 Space Surveillance Network. With space debris and satellites adding to the congestion around the earth, the chances for objects to collide continues to increase. The sensors in the Space Surveillance Network track objects with the hope of avoiding collisions. However, with the current resources it is impossible to track all satellites, leaving the potential for unforeseen collisions. Many people around the world depend on satellite technology like the Global Positioning System (GPS) satellites, which are vulnerable to space object collisions. Information from GPS satellites need to be transmitted and received in a timely manner. Any collision could inhibit the ability of the system to work properly. EAs are used to assign the sensor in such a way to minimize this potential problem.

Like the SSAP, scheduling for real-time systems incorporates a set of tasks that need to be assigned to sensors. More specifically, it is a priority-based set cover scheduling problem where most of the information is off-line. Off-line refers to the fact that the information needed to compute a schedule is available before one begins to process the tasks. This creates an interesting problem. Namely, it raises the question of how-to best merge potentially conflicting events into the schedule. Many possibilities emerge as the algorithm considers a set of heterogeneous sensors to perform the assigned tasks to make a schedule.

1.4.2 Scheduling Cycle. A 24-hour scheduling cycle works for many systems, but others like the SSN [9][27][75]. For example, one schedule that has been well studied is the Air Tasking Order (ATO). The ATO is a schedule used to have joint control over airborne assets. While having a good plan is desired, with allocating resources ahead of time, sometimes plans do not work. In addition, plans change with new information. In a 24-hour plan cycle like the ATO, the variables have a tendency to change in the fast-paced world. Research exists to revise the ATO in real-time [120]. The vastness of space and the large number of orbital objects makes real-time scheduling an aspirational goal at best, but the standard ATO serves as an operational scheduling problem that is well understood and well executed [27].

Tactical space operations regularly change the common space picture, leading to the need for careful consideration when making assignment decisions. It is desirable to

have the ability to make a schedule based on sensor management decisions. The task of assigning sensor collections to associated tracking satellites is like other scheduling systems that incorporate sets of tasks that need sensors.

Likewise, in the aftermath of a disaster, initial schedules are made as part of the on-going disaster recovery operations. In these recovery situations, many organizations bring a variety of tools to help aid wounded people. For search and rescue operations to be successful, an organized schedule for proper resource allocation is critical. Schedule creation needs to be made carefully to best benefit the overall health and safety of individuals that are in danger. Tools, such as satellite information, images, robots, radar, etc. need to be carefully scheduled to provide the most help to everyone.

The No Free Lunch Theorem tells us that an algorithm tuned for a given problem and input distribution will likely work poorly if the input distribution and desired output distribution changes. In the context of this dissertation, we can interpret the No Free Lunch Theorem as a warning that caution is needed when applying a given approach to superficially similar problems, as the technique for one may not work well for another [116]. With that said, incorporating problem specific information into an algorithm can greatly improve its effectiveness.

In scheduling, it is often desirable to have a well-planned schedule. For example, search and rescue applications may require complex advanced scheduling, covering many multi-dimensional factors based on current requirements [3]. This is particularly true when robotics and other automated mechanisms are employed. In this dissertation, evolutionary algorithms serve as the basis for both types of multi-dimensional planning.

1.5 Dissertation Overview

Chapter 1 presents the state of space surveillance and reviews broad approaches to solving the problem. Chapter 2 provides an initial research approach by introducing the EAT as an algorithm, which outperforms a published tasking algorithm. Chapter 3 presents the novel MEAT as a hybrid genetic algorithm in combination with evolutionary strategy to create solutions to the MOSSAP. Chapter 4 continues to explore the MEAT

and applies it to a heterogeneous sensor environment. Chapter 5 concludes the dissertation by summarizing its contributions, summarizes each of the previous chapters, provides final thoughts, and provides ideas for future avenues of research.

2. Sensor Allocation for the Space Surveillance Network

2.1 Introduction

Rocket, spacecraft, and satellite builders continue to launch new satellites, which are causing the exosphere to become more congested. As space becomes more congested, tracking and knowing the orbits of space objects becomes more important. Once accurate orbits are determined, collisions and/or damage from space debris can be prevented. Many key assets, including expensive satellites, manned spacecraft, etc. need to be protected. Each collision can cause objects to break up into smaller pieces, which could cause future collisions. Even a small piece can cause significant damage because of the kinetic energy of these impacts. These are just a few reasons why space surveillance is such a critically important topic. The purpose of this chapter is to take the first step in an incremental research model. This step is to introduce the EAT and show its effectiveness.

On March 24, 2012, a piece of debris passed close enough to the International Space Station that the crew was ordered into escape capsules as a precaution [71]. The object was spotted too late to move the orbiting laboratory out of the way. The debris came from a collision between two satellites in 2009 that created 2,000+ pieces of orbital debris. The problem grows with every collision. Thus, the ability to avoid collisions is critical. Such a collision could cause what is known as the Kessler Syndrome. It is the idea that one collision could cause a chain reaction of cascading collisions [58]. This potential catastrophe could paralyze space exploration if not resolved. Current sensor networks have a limited ability to track satellites, which is causing problems such as the collision in 2009 and the more recent close call with the International Space Station [112][113].

The Satellite Sensor Allocation Problem (SSAP) is an issue that needs a good solution. The SSAP is a type of resource allocation problem (RAP) [75]. The Space Surveillance Network (SSN) is a global network of ground-based radar and optical sites, as well as space-based visible and other types of sensors [39]. These sensors are used to detect, track, and catalog artificial objects orbiting the Earth. A key task in satellite surveillance involves maintaining the satellite catalog, which has been maintained since 1957 [87]. The orbital information contained in the satellite catalog is used to chart the relative position

of objects and predict future orbits. These future orbits are used to anticipate collisions or near collisions between space objects.

The satellite catalog is updated daily with information from satellite sensors. A relatively small number of high-capacity ground-based sensors are used to track many Resident Space Objects (RSO). In fact, the space around Earth contains too many RSO for ground sensors to track. As a result, the sensor schedulers must decide which RSO to track, among the myriad of possibilities [75][76][77].

This chapter's development consists of several key parts. First, priority is added to the model and the algorithms, so all new approaches can benefit from priority. Priority has been used in a greedy allocation scheme, but with this model, priority is balanced with probability that limits starvation of lower priority RSO. This way the most important RSO are tracked more often. Second, the Evolutionary Algorithm Tasker (EAT) is designed and developed with specific evolutionary techniques to perform on the SSAP. It is a unique approach as a major extension of a traditional genetic algorithm by modifying and adding several novel techniques that Section 2.4.1 discusses. It includes an exclusive selection operator and hybrid Evolutionary Strategy/Genetic Algorithm. The EAT is the first time to our knowledge that an Evolutionary Algorithm (EA) has been applied to the full scale SSAP.

This chapter discusses the current space tasking situation and some possible approaches to solving the problem. Section 2.2 summarizes RSO and the sensors that track them, as well as related algorithms. Section 2.3 formally defines the SSAP and its relation to the more generic RAP. *These sections, except for the conclusion, are split into two parts: one discussing a single objective evolutionary algorithm and the other discussing multi-objective research on the SSAP.* Section 2.4 presents the evolutionary algorithmic approach and the implementation of each algorithm. Section 2.5 discusses the goals and measurable objectives of experiments. Section 2.6 covers the experimental output and performs a statistical analysis of the data. Finally, Section 2.7 summarizes, offers conclusions, and recommends ideas for future work.

2.2 Background

Knowing the present state of SSN is antecedent to discussing solutions. The SSN includes details on RSO and the sensors that track them. Previous solutions to the SSAP are also reviewed.

2.2.1 Resident Space Objects in the Earth's Orbit. *RSO* are active systems that communicate information or space junk (consisting of old rocket boosters, retired satellites, etc.). The SSAP only considers the RSO that are trackable, numbering in the tens of thousands, see Table 2. The table has three rows based on categories: trackable, potentially trackable, and untraceable. Each category has corresponding sizes that define general limits to each category. The estimated population is how many objects around the Earth exist in each category. The tiny, untraceable objects are estimated to be in the many millions to billions! However, it is hard to know a more exact amount. The last column is the potential risk to RSO. Even though these objects are small, they can cause significant damage. These numbers are updated as of May 2022, but they are expected to increase rapidly.

Table 2 Three Categories of RSO with Their Definition by Size, Estimated Population, and Potential Risk [38, 112]

Category	Definition (diameter)	Estimated Population	Potential Risk to RSO
Trackable	> 10 cm	36,500	Complete destruction
Potentially Trackable	> 1 cm	1 million	Complete to partial destruction
Untraceable	> 1 mm	130 million	Degradation, loss of sensors or subsystems

The image in Figure 2 provides a visualization depicting where the greatest orbital debris populations exist. About 95% of the objects in this image are non-functioning RSO as opposed to the relatively small number of active payloads and operational satellites [47]. The objects are scaled with respect to the image size so that they can be seen otherwise, they would be smaller. As displayed in the illustration, the most concentrated area for orbital debris is within 2,000 km of Earth. A ring of functioning satellites is about 35,785 km above the equator in geosynchronous orbit. Geosynchronous debris is a threat to these functioning satellites even at this great distance from Earth [10].

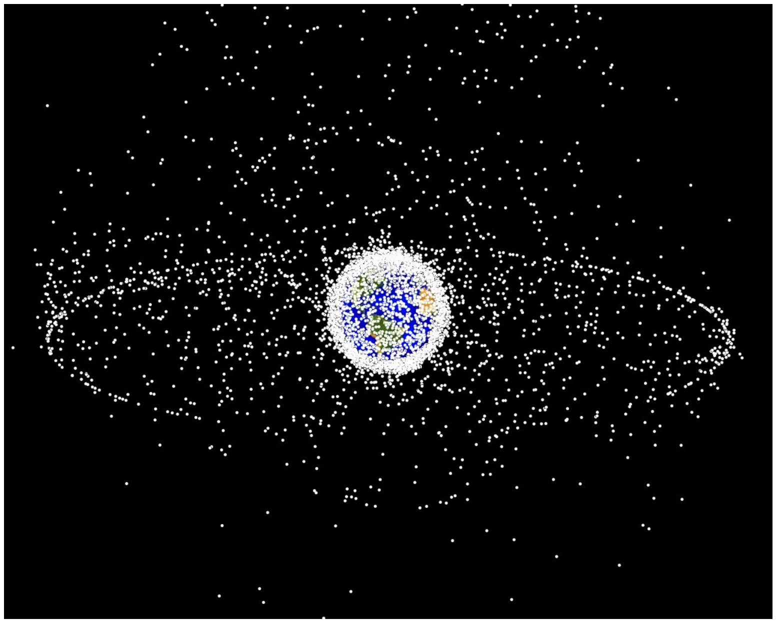

Figure 2 Depiction of Orbital Debris Surrounding the Earth [47]

2.2.2 Satellite Tracking Sensors. Satellite tracking sensors detect objects and gather positional information. A single point of positional data is called an observation. A track is a collection of observations obtained during one pass of a space object [87].

The tracking sensors come in many forms, including various radar systems, optical telescopes, laser ranging, the International Space Station, other satellites, etc. [39]. Ground radar sensors form the core of the SSN and perform most of the tracking, which most commonly use phased-array radar [22]. The second major type of sensor is the optical telescope. The satellite laser ranging sensors use ultra-short pulses of light to satellites equipped with reflectors [115].

The Eglin Air Force Base (AFB) phased-array radar is one of the main workhorse locations in the SSN, accounting for 30 percent of the total world-wide network capacity [22][91]. Most of the other radars track significantly less RSO per day. For example, Fylingdales, UK, can track 856 RSO per day, and Thule AFB can track 572 RSO per day [28][11]. Most, but not all, the sensors have published information about objects tracked per day. The ones that do not have public information were calculated based on the published number of active elements for the radar. For instance, if a radar has 7680

active elements, then a good estimate of how many objects it can track in a day is 900 because many elements are needed to obtain each observation of an object [22][28][11]. Furthermore, if radar has more elements, it will be able to track more RSO.

This chapter focuses on the assignable ground-based satellite sensors. Because the primary mission of most sensors is missile warning, tracking RSO is secondary to these sensors. However, a few dedicated sites have satellite tracking as their primary mission. The SSN has many sensor sites worldwide, totaling around 250 sensors [9]. Currently only 4 (2 U.S. and 2 Canada) are doing Space Based Space Surveillance (SBSS), that is satellites tracking other space objects, but Utzmann et al. plan to have more satellites in a network tracking separate areas of space [106][107].

2.2.3 Perturbed Motion. Perturbed motion is relevant because normal two-body (Earth-space object) motion theory is not enough to accurately predict an orbit propagation of satellites. Barker explained it well when he wrote, "The actual path will deviate from the two-body path due to perturbations caused by external mass bodies (e.g. the Sun, Moon, and planets) and internal forces not considered in Keplerian motion (e.g. due to the geopotential, atmospheric drag, etc.)" [8]. The two-body motion equation can be found in vector form in Equation (1), and the perturbed motion equation where the perturbing accelerations are added to the two-body motion in Equation (2). The perturbing accelerations, $\ddot{\vec{\boldsymbol{r}}}_P$, are a sum of influences like the Sun, Moon, atmospheric drag, etc. Both equations and more information about the differences between General Perturbations (GP) and Special Perturbations (SP) are in [8].

$$\ddot{\vec{\boldsymbol{r}}}_{2B} = -\frac{G_M}{\boldsymbol{r}^3}\vec{\boldsymbol{r}} \tag{1}$$

$$\ddot{\vec{\boldsymbol{r}}}_T = -\frac{G_M}{\boldsymbol{r}^3}\vec{\boldsymbol{r}} + \ddot{\vec{\boldsymbol{r}}}_P \tag{2}$$

GP models are analytical orbit models, while SP models are numerically based [95]. Some historic GP models are the Simplified General Perturbations model 4 and Position and Partials as functions of Time 3 [53][54]. It has been known for some time that SP

models are the preferred way to represent orbital motion [95]. GP models may have kilometers of inherent prediction error, but SP models have only meters of inherent error [8][95]. SP models are more accurate because the numerical nature of the model gives it that ability. From the late 1990's to early 2000's new computer techniques were able to overcome the immense processing power it took to produce the long lists of numbers that SP models required. Table 3 provides an alphabetized digest of all symbols for Equations (1) and (2). It also holds many other symbols used in the remaining formulas for the readers to conveniently reference.

Table 3 Symbol Summary

C	the set of capacities c_i
c_i	capacity of the i^{th} sensor
D	the set of opportunities or passes d_{ij}
d_{ij}	an opportunity for the i^{th} sensor to track the j^{th} satellite
G_M	the gravitational parameter
i	the index that refers to a specific sensor
j	the index that refers to a specific satellite
k	the index that refers to a specific daily pass
m	number of sensors
n	number of RSO
O	the set of priorities o_j
o_j	priority of the j^{th} satellite, where $1 \leq o_j \leq 5$
P	the set of probabilities p_{ijk}
p_{ijk}	probability of the i^{th} sensor to track the j^{th} satellite on the k^{th} daily pass, where $1 \leq k \leq d_{ij}$
R	the set of required tracks per day r_j
r_j	required tracks per day for the j^{th} satellite
$\boldsymbol{r}$	the distance between object centers
$\vec{\boldsymbol{r}}$	relative location vector
$\ddot{\vec{\boldsymbol{r}}}_{2B}$	the two-body acceleration vector acting on the satellite
$\ddot{\vec{\boldsymbol{r}}}_P$	the combined perturbing acceleration vector (sun, Moon, atmospheric drag, radiation, etc.)
$\ddot{\vec{\boldsymbol{r}}}_T$	the total acceleration vector acting on the satellite
s	the scaling factor for the EAT scoring value
T	the size of the neighborhood for MOEA/D-DE and pMOEA/D-DE
X	the output set that is an allocation of sensors tasked to track RSO
x_{ijk}	number of tracks allocated from the i^{th} sensor to the j^{th} satellite on the k^{th} daily pass

2.2.4 SP Tasker. James Miller developed the SP Tasker at the MITRE Corporation [75]. The SP Tasker is an algorithm designed to tell the sensors at the ground stations which RSO to track. The SP Tasker uses SP instead of a traditional general perturbations method because SP are more accurate. The tracked RSO are recorded in a database cataloging satellite movements [9]. The sensors cannot track all of the RSO every day. The system needs to calculate whether it will track or ignore each satellite. A sensor is tasked

when it is assigned an object to track. A satellite is tracked when the sensor receives the tracking information for the satellite.

The current space tracking algorithm, the SP Tasker, is more effective by several measures than its predecessor, the Space Defense Operations Center (SPADOC) tasking process [68]. However, the SP Tasker does not allow for tracking priorities to reflect the reality that tracking certain RSO is more important than tracking others.

The SP Tasker is compared to the SPADOC tasking process [75]. The SP Tasker experiments show that it is more effective than SPADOC because it can task more RSO to the same sensors. On November 9, 2005, the SP Tasker replaced the SPADOC tasking process to become the operational system tasking the SSN [75]. Since the SP Tasker has been operational, it has improved the performance of sensor tasking on two important metrics. The SP Tasker decreased the number of RSO uniquely tracked. The *unique-track percentage* is the percentage of tracked RSO that were tracked by only one sensor. If RSO are tracked by only one sensor, they have a single point of failure, allowing for more risk if some sensors were unable to track the RSO assigned to them. The SP Tasker increased the *track response rate*, which is the percentage of satellite tracks obtained over the number of tasks assigned to the sensors.

The SP Tasker algorithm for solving this problem is augmented from a marginal analysis algorithm in Denardo's *Dynamic Programming* book [32]. This algorithm solves a single RAP, and it is extended for the SP Tasker algorithm to solve the multi-RAP [75]. A version of the SP Tasker is implemented based on the information from the literature for comparison to the EAT [75][32]. Since the SP Tasker is currently in use, it is the standard for testing. For balanced testing the three original metrics (unique-track percentage, track response rate, and not tracked percentage) evaluating the SP Tasker are reevaluated in experimentation to compare the SP Tasker and EAT.

2.2.5 Evolutionary Algorithm. The choice to develop an EA, specifically the EAT, to solve the SSAP is because EAs have been shown to perform well on RAP. A couple examples, of related research where EA work well on RAP, are articles of Osman and Newman [86][84].

M. S. Osman uses an evolutionary algorithm approach to solve a multi-objective RAP [86]. Like the approach Miller uses with the SP Tasker, Osman also uses dynamic programming techniques to solve a multi-objective resource allocation problem. However, Osman concludes that the dynamic programming approach can have problems due to the potential for rapid state explosion. Since genetic algorithms are designed to efficiently search a large population of points, they should be better with the numerous states in such cases.

Newman compares a variety of stochastic optimizers to task sensors including an evolutionary algorithm (EA) approach like the method developed here, but with different parameters [84]. The EAT simulation size is also much larger and closer to actual sensor and satellite numbers than those employed by Newman. They compare their EA with a particle swarm optimizer, a combination of swarm and EA, and a perturbation based stochastic approach [84]. Their results show the hybrid combination of an EA, and a particle swarm optimizer did better than the other three approaches; however, these results are from a small scale test.

2.3 Problem Definition

The *goal* of the SSAP is to find an assignment of sensors to RSO given specific sensor and satellite constraints. The *first objective* is to have a higher accumulative probability of RSO being tracked and cataloged; the *second objective* is to have the higher priority RSO more likely to be tracked [75][84]. The *constraints*, such as sensor capacity, satellite daily passes, satellite track requirements, and priority serve as input for the problem and make it difficult to achieve these objectives. These constraints are based on real world physical restrictions like the relative location of the RSO and the sensors, the sensors field of view, and the capacity of each sensor. The model covers these and other physical restrictions in more detail. Also, the model shows how the complex restrictions reduce to their pertinent factors.

For a formal definition of the problem, the SSAP consists of a set of sensors and a set of RSO. The number of sensors is m, and n represents the number of RSO.

Each sensor can only track a certain number of RSO where $c_i \in C$ is a positive integer and denotes the capacity of the i^{th} sensor. The limited capacity of physical sensors prohibits the sensors from tracking all the RSO. Specifics on the actual capacity of the sensors are in Section 2.2.2.

Nonetheless, each object has a set number of tracks per day that are necessary for accurately determining orbital information [75]. The sensors can obtain complete (based on requirements) tracks for many of the RSO where $r_j \in R$ is a positive integer and denotes the required tracks per day for the j^{th} satellite. The required tracks per day are the number of tracks that the sensors must catalog for the satellite to be considered fully tracked. In order to obtain a full track, a set of observations must be taken to determine the current orbital path of the satellite.

This set of observations for a single track can be recorded during one pass of the satellite over the sensor. Each sensor's field of view is analyzed, resulting in a daily pass and corresponding track probability for each time the satellite enters a sensor's field of view. Each satellite has a set of daily passes or opportunities to be tracked where $d_{ij} \in D$, where d_{ij} is a positive integer and denotes an opportunity for the i^{th} sensor to track the j^{th} satellite. For example, if satellite, $j = 29$, passes into the sensor, $i = 6$, field of view four times in one day, the amount for that daily pass would be $d_{29\ 6} = 4$. Likewise, if the satellite does not pass over the sensor in that day, the corresponding daily pass value would be zero.

The daily pass value is how this model store the location of an object over a sensor. The exact location of each object cannot be predicted before the schedule or assignment is produced, but a reliable estimate location is necessary for planning purposes. *The model depends on this location to accurately plan and task the sensors.* Each sensor in the SSN has a specific location. For example, since the location of the Eglin sensor is in Florida, an object must be expected to be within the sensor's field of view somewhere close to the Southeastern United States to register in the model as a daily pass. There are many other sensors as well. Some ground-based sensors are Ascension, Clear, Cobra Dane, Diego Garcia, Fylingdales, Globus, and Thule to name a few. Also, space-based sensors are

GSSAP, NEOSSat, SBSS Block 10, Space Tracking and Surveillance System, Sapphire, and more [1][29].

Each pass has a track probability which is determined by range and radar cross section. When considering how the range influences probability, an object being further from the sensor corresponds to a lower probability of receiving a good signal. The radar cross section is the object's ability to reflect a radar signal back to the receiver. The ability to reflect the signal impacts track probability settings as well. The track probability $p_{ijk} \in P$ denotes the probability of the i^{th} sensor's ability to track the j^{th} satellite on the k^{th} daily pass where $1 \leq k \leq d_{ij}$.

The imagery in Figure 3 illustrates the relationship between daily passes and track probability. This pedagogical example has three sensors (1-3) and fourteen satellites (A-N). The satellites are labeled by letters to mark the current locations and the wavy lines to describe their next few hours of orbit. The sensors are dots on the map with concentric circles emanating from the middle. This image has many notable events. As the simulation progresses the first event is satellite C doing one of its daily passes over sensor two about eight minutes into simulation time. Then satellite J is over sensor one at fifty-three minutes, followed by M over two at sixty-one minutes. The last event easily seen in this two-hour window is D over one at ninety minutes. Satellite D clearly goes over sensor one, but it does not cross close to the center like the other satellites did. This pass will get a lower probability because there may not be enough time in the sensors field of view to obtain the observations necessary for a complete track. Note: Sensor three looks like it is all alone at this snapshot, but if the simulation continues even with this small number of satellites; E will pass over it at 126 minutes. It is also important to remember that each time a satellite is recorded to be over a sensor that implies the sensor can see the satellite in its field of view. Systems Tool Kit (STK) is a product of Analytical Graphics Inc. (AGI) [5]. STK includes, but is not limited to, simulating ground and space objects as they interact with each other. STK produced this image, but all the algorithms are implemented, and experiments completed outside of STK.

The track probability and the priority are both important aspects of this proposed approach. Every satellite has a priority where $o_j \in O$ denotes the priority of the j^{th}

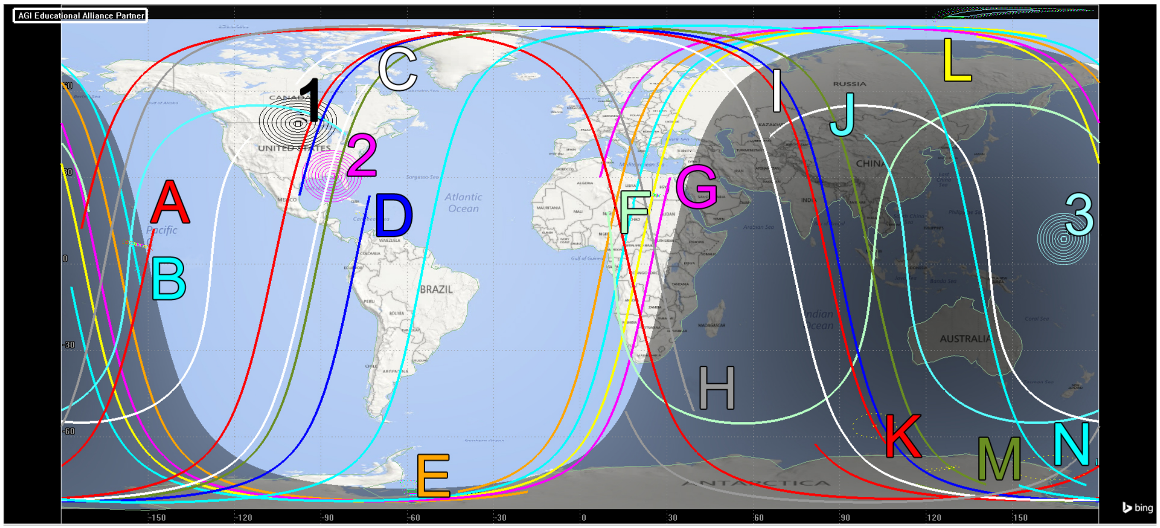

Figure 3 STK Example: Displays Three Sensors (1, Cavalier; 2, Eglin; 3, Kwajalein) as Concentric Rings and Fourteen Satellites (A-N) with Their Orbital Lines [5] (Traci Greve enlarged the labels for better visibility in Photoshop [2]).

satellite. The priority range is $1 \leq o_j \leq 5$ with 1 being the most important. Priority is mainly based on the significance of the satellite and potential loss in the event of a collision. For instance, active RSO are a higher category like 1, 2, or 3, and inactive RSO or debris are in categories such as 4 or 5. The five categories are adequate to address the diversity of importance among RSO [23][87].

The orbit accuracy of a satellite causes differing priority levels as well. These levels can be assigned based on calculations of the orbit error covariance, which is a good measure of orbit accuracy [51]. In order to reduce the error in estimating the orbit of a satellite, more error prone RSO are given slightly higher priority.

All of these parameters (C, R, D, P, O) serve as input to a tasking solver. The output is an allocation of sensors tasked to track RSO X where $x_{ijk} \in X$ is a Boolean value and denotes the track allocated from the i^{th} sensor to the j^{th} satellite on the k^{th} daily pass $(1 = true$ and $0 = false)$. A set X is part of the chromosome for the EAs.

2.3.1 Resource Allocation Problem. As previously noted, the SSAP is a specific form of RAP and can be categorized in the class of general theoretic scheduling problems. Since the SSAP involves multiple sensors, it is essentially a multi-RAP. Toshihide Ibaraki

and Naoki Katoh proved that this problem is NP-Hard [55][16]. No one can expect to find an optimum solution in polynomial time; therefore, a polynomial time heuristic algorithm is needed to provide a sub-optimal but good solution to this problem in a timely manner. The EAT is such a heuristic, and the EAT is also a precise mode-approximation solution to this specific form of resource allocation.

A RAP tries to distribute scarce resources among many activities and can be described using a linear programming model. In this case, the resources are the sensors, and the activity involves tasking the sensors to track RSO. A linear programming model can define explicitly both single and multi-objective formulations for the SSAP. This model incorporates all of the pertinent factors for the daily SSAP.

2.3.1.1 Single Objective Problem. To officially map the SSAP to an RAP, the following linear programming model is created, where the objective is to maximize Equation (3).

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{d_{ij}} p_{ijk}x_{ijk} \tag{3}$$

Subject to the following linear constraints:

$$\sum_{j=1}^{n}\sum_{k=1}^{d_{ij}} x_{ijk} \leq c_i \quad \forall i \tag{4}$$

$$0 \leq \sum_{k=1}^{d_{ij}} x_{ijk} \leq d_{ij} \quad \forall i\ \forall j \tag{5}$$

And soft constraint:

$$\sum_{i=1}^{m}\sum_{k=1}^{d_{ij}} x_{ijk} = r_j \quad \forall j \tag{6}$$

The first objective in Equation (3) is to maximize the summed *probability* of tracking as many RSO as possible. The limited sensor resources do not allow observers to track

all the RSO, so the objective to maximize the *probability* of cataloging as many RSO as possible. To put it another way, the objective maximizes the probability of tracking all the RSO that are scheduled to be tracked. The higher the probability, the greater likelihood additional RSO will be tracked.

If the algorithm assigns sensor i the task to track satellite j on its k^{th} pass, then p_{ijk} is added to the summed probability objective. Otherwise, if the sensor i is not assigned to track satellite j on pass k, then p_{ijk} is *not* added to the sum because $x_{ijk} = 0 = false$.

The constraints model the real-world limitations inherent to the space tracking problem. The first set of constraints in Equation (4) prevents the total number of allocated tracks for the i^{th} sensor from exceeding the capacity c_i of that sensor. This ensures that each sensor is not over assigned.

The second set of constraints in Equation (5) keeps the algorithm from allocating more tracks x_{ijk} than the daily passes d_{ij} or opportunities to track the satellite. The system cannot track an object more times than it physically exists above the target.

The final set of constraints in Equation (6) ensures that the algorithm's selected tracks meet the required number of tracks r_j for each object if possible. In order to obtain complete information about each object, the requirements must be met; however, since this is a soft constraint, the algorithm can obtain at least partial requirements on some RSO even if a full track requirement cannot be met. This partial information is better than nothing when the analysts are trying to determine risk of collision.

2.3.1.2 Multi-objective Problem Formation. The model defined in the previous section focused on single objective optimization, but this approach also uses multi-objective optimization on the SSAP. For a multi-objective problem, another objective is needed in addition to Equation (3). The goal is to maximize Equations (3) and (7) subject to the constraints in Equations (4)-(6). Therefore, the probability objective is defined in Equation (3); the priority objective is defined in Equation (7).

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{d_{ij}} \frac{x_{ijk}}{o_j} \tag{7}$$

This priority objective is to maximize the ratio of allocated RSO to the corresponding priority. The allocated tracks x_{ijk} are divided by the priority o_j. The summation is designed to ensure the highest priority RSO affect the objective value more than the lower priority RSO. For example, if $x_{ijk} = 1$ and $o_j = 2$, then values would translate into a larger impact on the total sum than other scenario of $x_{ijk} = 1$ and $o_j = 5$ (i.e. $\frac{1}{2} > \frac{1}{5}$). Accordingly, each of these values is summed up where the higher priority RSO have more significance.

2.4 Computation Domain

In the computation domain, the algorithms are developed and detailed. The first is the EAT single objective evolutionary algorithm. The other algorithms are a few Multi-Objective Evolutionary Algorithms (MOEA) using the two objectives of probability and priority. These heuristics are based on the mathematical model and parameters result in good resource allocation (i.e. an assignment of sensors to track RSO).

Genetic algorithms are search methods that emulate biological natural selection and survival of the fittest [4]. They start with a "population" of solutions to the problem. Then they use a series of selection, recombination, and mutation to potentially improve the solutions. The individual solutions can also be called a genotype or chromosome. A chromosome is a specification of the solution to the problem.

Table 4 Example Strand of Chromosome: Holds All the Assignment Values x_{ijk} for Sensor Zero.

$x_{000} = 0$	$x_{010} = 1$	$x_{020} = 0$	$x_{030} = 0$	$x_{040} = 0$	$x_{050} = 1$	$x_{060} = 1$	$x_{070} = 0$
$x_{001} = 1$	$x_{011} = 0$	$x_{021} = 1$	$x_{031} = 0$	$x_{041} = 1$	$x_{051} = 0$	$x_{061} = 0$	$x_{071} = 1$
$x_{002} = 0$	$x_{012} = 0$	$x_{022} = 1$	$x_{032} = 0$	$x_{042} = 0$	$x_{052} = 0$	$x_{062} = 0$	$x_{072} = 1$

Both single and multi-objective problems use the same chromosome definition for a solution to the SSAP. Each individual chromosome is a random assignment of RSO to be tracked by the sensors that is constructed with the capacity, probability of obtaining a track, and the daily passes. The initial assignments are the sensors being randomly assigned RSO to track that are feasible based on the constraints. For example, if the daily pass does not exist then that assignment cannot be made. The capacity is the length of

each strand. A strand is only part of a whole chromosome. The daily passes dictate the depth of each individual column. For a small example to illustrate one strand see Table 4. This tiny example strand shows only 8 satellites allowing for the whole strand to be displayed. The core of the strand is the x_{ijk} Boolean values because that is where the assignment decisions are delineated.

Table 5 Chromosome Structure: A strand for each sensor is combined to form a chromosome, where the varying width represent the different capacities.

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7

One strand for each sensor combines to make a full chromosome. The chromosome seen in Table 5 has a strand for each sensor, and the width of each column is based on the capacity of that sensor. The X_0 represents the entire set of x_{ijk} values sensor 0. The capacity values do not change for the experiments because each sensor capacity is set to correspond to the capabilities of that real-world sensor. Each sensor cannot exceed its capacity c_i set by constraint Equation (4). Each x_{ijk} is a solution of Boolean values summed for each sensor satellite pair. It must be greater than or equal to zero and less than or equal to the daily passes for each pair. The values of x_{ijk} are dictated by constraint Equation (5) in this manner.

2.4.1 Evolutionary Algorithm Tasker. The core operating principle for the EAT is that a "good" evolutionary algorithm should *explore* the solution space and *exploit* the good individuals it finds. In comparison with the SP Tasker approach, the EAT is in a separate category of optimization algorithms altogether. This means the algorithms are structured very differently. The SP Tasker is based on the idea of Marginal Analysis. For Marginal Analysis to work, the function must exhibit decreasing marginal return, which is synonymous with concavity [32]. Because the RAP has a concave function, the SP Tasker can work, knowing that if it tasks a satellite to be tracked after that satellite has already been tracked several times, each additional track will have decreased the value of return. More details on the SP Tasker and its Marginal Analysis approach is available in the literature [32][75].

The EAT is an evolutionary algorithm that is a hybrid Evolutionary Strategy (ES) and Genetic Algorithm (GA). Considering the parameters (capacities, daily passes, probabilities, requirements, and priorities) the EAT starts by generating an initial population of 64 possible individuals. The range is also limited to 64 children generated. Given the large size of the chromosome, the population and range of the ES are limited to aid computational efficiency. EAT generates a new population at each iteration based on the previous or initial population, as shown in Figure 4. A traditional GA does a single selection in which the entire selected population is treated the same, but this algorithm uses two different ways of selecting individuals to split the population into two parts and eventually four parts. The first part keeps the best 25% of the individuals from the previous generation like the elitist strategy of Goldberg [41]. This ensures a "healthy" portion of the best solutions already found are kept. Since priority is a part of the fitness metric, the EAT selects solutions with a higher likelihood of containing more important satellites. Other approaches like the SP Tasker and Newman's EA do not calculate fitness based on priority [75][84]. The second 25% is a distinctly random subset from the previous generation to explore the solution space stochastically. This keeps the population spread out over the solution area without much computing power. All individuals in the initial population are available for random selection to prevent loss of genetic diversity, which could result in the solution getting stuck at a local minima. Keeping the entire parent for the next generation is not like a traditional GA, because traditional GA would only keep a genetic recombination of the parent.

These two groups are then used to derive the other two groups. Some of the techniques used in the EAT are related to those found in research that combined Evolutionary Strategies (ES) and Genetic Algorithms (GA) [15][42], but the EAT has some different techniques. Typical ES only do the mutation step to introduce change [63] [96]. Like an ES, the EAT performs only a mutation on the best evaluated individuals. This way the algorithm tries to exploit the best individuals of the population. The third 25% of the new generation is the result of point mutations based on the best 25% already found. A point mutation is a small alteration to a good solution that potentially improves the individual. This small change is applied by removing one satellite assignment from a sensor

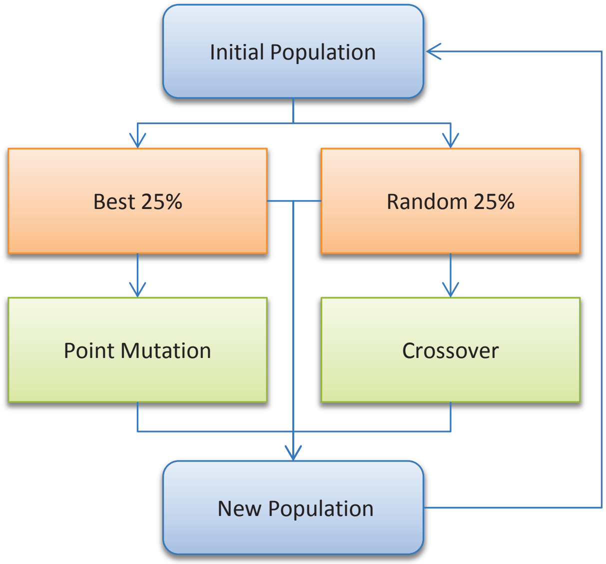

Figure 4 EAT Generation: Starting with an Initial Population the Best 25% individuals and a Random 25% are selected for the New Population. Then a Point Mutation is performed only on the Best Set, and a Crossover is performed only on the Random Set.

and replacing it with another satellite assignment. In early testing the mutation rate is a standard 0.05, but it is incremented while producing slightly better solutions until it peaked at 0.25. Thus, the EAT uses 0.25 as the mutation rate.

As well as exploiting the population, the EAT is designed to explore the solution space. The algorithm does this by using a standard two-point crossover [101]. However, it differs from most literature because the crossover is performed without a following mutation. The mutation is such a small change compared to the crossover recombination. The EAT performs a two-point crossover on the 25% random solutions already chosen to form the final 25% of the new generation. A new set of two random points are selected as crossover points for each generation, but the crossover points are fixed during the crossover operations for each generation. The next generation creates two new random points for crossover. The crossover itself is performed at a high rate of 100%, meaning that at every potential crossover point the crossover is performed. That sounds very high, but if should be remembered that the EAT is selecting only 25% of the overall population to make

```
FUNCTION generateNextGeneration (currentPopulation, scores)

newPopulation = emptySet; // initialize to empty set
bestSet = best25(currentPopulation, scores);
newPopulation = bestSet; // add 25 best individuals to new population
randomSet = random25(currentPopulation);
newPopulation += randomSet; // add 25 random indiv. to new population
// do point mutation and add them to new population
newPopulation += mutateParent(bestSet);
// take 2-point crossover and add them to new population
newPopulation += twoPointCross(randomSet);

return newPopulation
```

Figure 5 EAT Next Generation Pseudocode: The function *generateNextGeneration* takes the *currentPopulation* and forms a *newPopulation* through selection, mutation, and crossover.

this extreme change. The crossover is a stronger change with the goal of finding a good individual in an area not yet explored.

The EAT then evaluates the population by giving a score to each individual determined by its probability and coverage requirements. The individual with the best score is found and archived. Subsequent generations are likely to find a better individual who replaces the best solution each time until the algorithm converges. The algorithm terminates when it converges on a best solution, or 1000 iterations are reached. The count of 1000 is chosen from the varied initial parameter testing to save computational resources. The individual with the best score is the final output.

For more information the pseudocode is provided in Figure 5. This code shows that each of the four parts is created from the current population and is added in succession to make a new population.

2.4.2 Multi-Objective Evolutionary Algorithm. Not only does single objective algorithm work on the SSAP, but also MOEAs work on the SSAP as well by separating the single objective into both probability and priority parts. The well-known evolutionary algorithms are used like MOEA/Decomposition-Differential Evolution (MOEA/D-DE) and the parallel version, pMOEA/D-DE [65][82][122]. The DE represents an extended version of MOEA/D using differential evolution as the main search engine [67].

Nebro and Durillo further developed the MOEA/D-DE to create a parallel version pMOEA/D-DE [82]. In Figure 6, a flowchart of the parallel version shows which pieces of the algorithm execute in parallel and which parts must execute sequentially.

Both MOEA/D-DE and pMOEA/D-DE are initialized with the same four steps, which are to initialize the weight vectors, neighborhood, population, and ideal/reference point consecutively [122]. In any MOEA, the individuals in a population have their own weight vector that algorithm uses to compare potential individuals. Both algorithms use the weight vectors, so the initialization step creates a weight vector for each individual that is evenly spread across the solution space. Once the weight vectors are created, Euclidean distances can be calculated between them. These distances are analyzed to find the T closest weight vectors, where T is the predetermined size of the neighborhood. Each vector has an initial neighborhood that is the T closest vectors. The third initialization step is to generate a population based on problem specifications. Lastly, an ideal point or reference point is selected beyond the Pareto front to help drive the solution toward this goal [122].

At this point, the pMOEA/D-DE starts to differ from the MOEA/D-DE. Next the pMOEA/D-DE determines which set of indices or population group are run on each of the available threads. Then, the algorithm iterates until the termination condition is met.

At each iteration a probability is generated to select whether the neighborhood mating pool (local) or the entire population mating pool (global) is used for selection and recombination. After recombination, these algorithms perform a polynomial mutation that is detailed in their article [82]. The new individuals are evaluated to determine their fitness value. Finally, the pMOEA/D-DE updates the ideal point and solutions. More information on updating ideal/reference point and updating solutions can be found in Li and Zhang's work [65], and full pMOEA/D-DE pseudocode can be found in Nebro and Durillo's work [82]. The pMOEA/D-DE works similarly to other MOEAs by keeping/updating a set of solutions from which the decision maker picks. The main differences are the decomposition into subproblems with weighted vectors and the neighborhood mating.

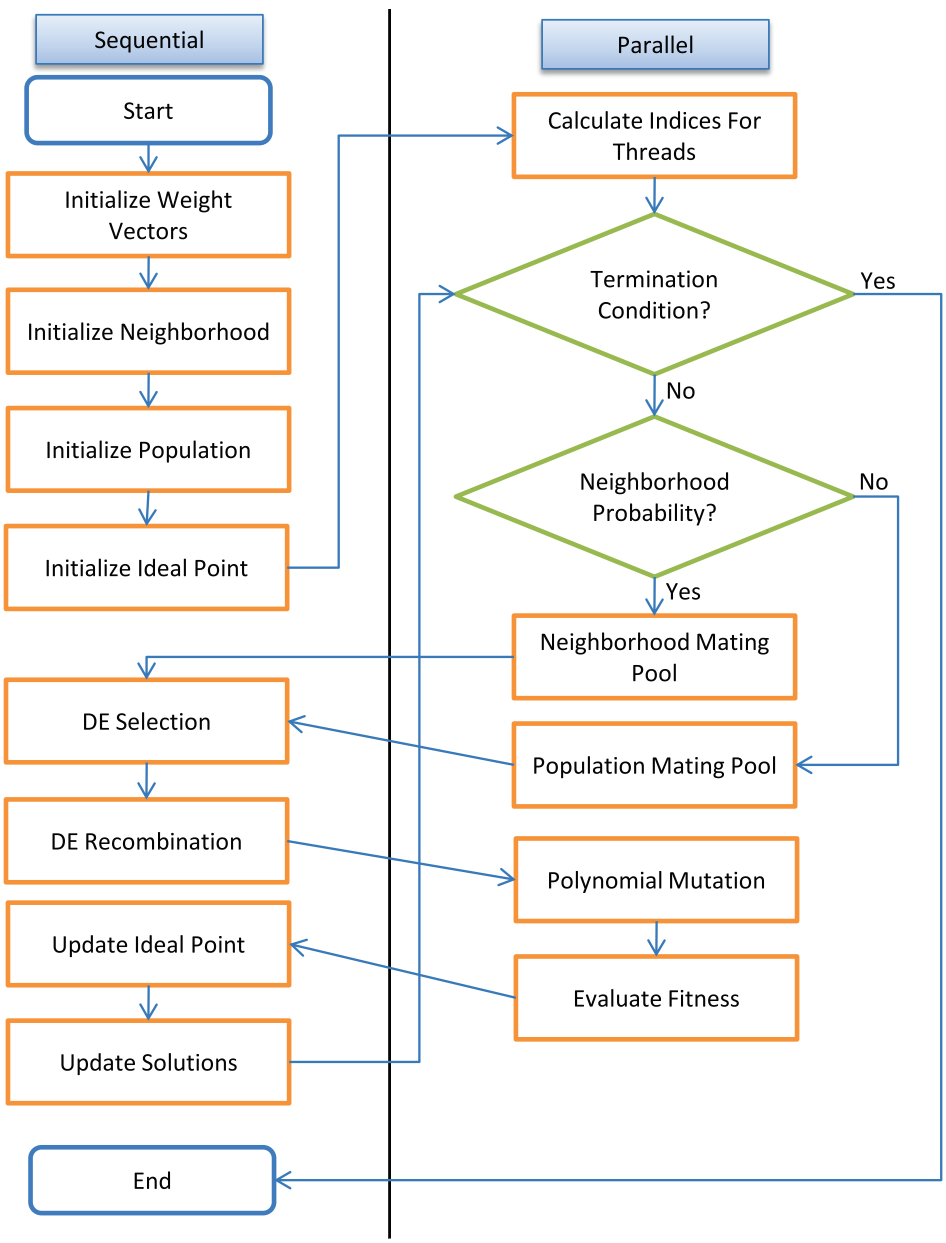

Figure 6 The pMOEA/D-DE Flowchart: Illustrating Sequential Components of the Left and the Parallel Components on the Right [82]

2.4.2.1 Algorithmic Complexity. MOEA/D-DE and pMOEA/D-DE are memetic algorithms. Memetic algorithms do not focus only on global search, but also use local search techniques to explore a neighborhood and reduce computational complexity [33]. MOEA/D-DE has computational complexity of $O(MNT)$ where M is the number of objectives, N is population size, and T is the number of neighbors [122]. Many common MOEA have computational complexity of $O(MN^2)$ [30][65][124]. Since T is less than N, MOEA/D-DE has a lower computational complexity. The smaller complexity is due to the MOEA/D-DE algorithms employing a local search instead of a global search. They only consider the neighborhood of subproblems to determine the mating pool.

Computational complexity is important because the SSN is on a 24-hour tasking cycle, meaning the sensors get a new assignment every day. Given this, the run-time should not take longer than one day. The dominate factor in the computational complexity of the metaheuristic selected is the number of RSO. Since simulation needs to track many RSO, the algorithmic time has to be a computationally efficient polynomial.

2.5 Experimental Design

Two sets of experiments are run. The first set relates to the single objective EAT, which has its own measurable objectives. The other set of generated measurements relates to the multi-objective approach implemented in jMetal. The jMetal framework is "an object-oriented Java based framework for multi-objective optimization with metaheuristics" [83]. The framework has implemented several multi-objective algorithms and benchmark MOEA problems ready for experiments. Users have many options to experiment with inside jMetal, some of which are to develop their own algorithms or solve their own multi-objective optimization problems. Another environment considered is the MOEA Framework, which would have been a good option, but jMetal has an easier implementation and well-planned design [44]. In fact, MOEA Framework makes use of some of the jMetal metaheuristic code.

2.5.1 Evolutionary Tasker Algorithm Design. The SP Tasker and the EAT are compared with four metrics. The first metric is the *not-tracked percentage*, which is the

percentage of RSO that did not receive any tracks at all. The lower the not-tracked percentage the better, because the target is to track as many RSO as possible.

Second, the *unique-track percentage*, previously mentioned in Section 2.2.4, should be minimized. For example, if a sensor is the only one assigned to track a satellite and the sensor cannot perform a full track for some reason; other sensors may be able to track the satellite. The fact that most sensors manage tracking as a lower priority task means that this activity is subject to interruptions (highest priority is missile warning). If that occurs, the impact on satellite tracking would be significant, especially if sensor 0 fails (X_0 and P_0 of the first strand of the chromosome representation in Table 5). If it was just the worst-case sensor that fails, thousands of satellites would not be tracked.

The third metric is the *track response rate* defined as number of tasked tracks received/number of tasked tracks, which should be maximized. In other words, the track response rate is the percentage of how often the satellite assigned to a sensor is being tracked. The track response rate should be maximized. The final metric is the *run time*, which should be minimized and should finish within the 24 hours.

Both implementations of the SP Tasker and the EAT are given the same data sets and are run 50 times to form a solid statistical base for analysis of variance testing. Also, the input data is generated from random seeds. The simulation tests are constructed so that they apply many factors that exist in the real-world SSAP. Some of these real-world factors are the number of sensors and RSO, sensors' fields of view, locations of both sensors and RSO, capacities of the sensors, etc. The application uses $m = 8$ sensor sites and use $n = 20,000$ RSO. The set of sensor capacities, C, is the known capacity of each sensor site ranging from $400 \leq c_i \leq 10,000$, which is detailed in Section 2.2.2. The SSN Optimization Study determined the required number of tracks per day, R, for the SSN to meet the US Strategic Command Capstone Requirements Document accuracy requirements [102]. The set of daily passes, D, and set of probabilities, P, are based on an STK simulation for a 24-hour period. The satellite priorities, O, are set based on mainly risk of loss, which is detailed in Section 2.3.

The EAT scoring is based on priority and probability, shown in Equation (8). The priority can be a partial measure of a sensor-satellite assignment, and the inverse probability can also be used as a partial measure of quality for a sensor-satellite assignment. Summing these two measures gives $(o_j) + (1 - p_{ijk})$, which does not yield good results. Because priority ranges from 1-5, the priority dominates the probability that ranges from 0-1. This can lead to starvation for the lower priority RSO. To limit starvation a better formula is developed by adding a scaling factor to give the probability more strength resulting in Equation (8). The scaling factor is four and is used in the experiments which are determined from initial testing.

$$(o_j) + [s \times (1 - p_{ijk})] \tag{8}$$

2.5.2 Multi-Objective Design. Both evolutionary algorithms have parameters that are set in the same manner to form a consistent baseline for testing and comparison. Each algorithm has a population size, 100; distance vectors, 100; maximum evaluations, 25,000; distribution index, 20; and crossover probability, 90%. The distribution index is the same for both crossover and mutation. If the EA allows too many evaluations or allows the population size to grow too large, the algorithm may not finish within the required 24-hour window mentioned in Section 2.4.2.1.

2.5.2.1 Implementation in the jMetal Framework. The jMetal4.3 software package is well designed from base components that allow for relatively easy problem implementation [35]. The goal is to implement SSAP to run against a couple of MOEAs and examine the differences. Figure 7 shows a UML (Unified Modeling Language) diagram illustrating how the SSAP Class is implemented in the jMetal Framework. The base of jMetal has abstract classes, such as `Algorithm` and `Problem`. The specific metaheuristic algorithms, like `MOEAD` or `pMOEAD`, extend from the abstract class `Algorithm`. Likewise, the implemented `SSAP` extends from the abstract `Problem`. Note: the jMetal code calls these algorithms MOEAD and pMOEAD even though they are MOEA/D-DE and pMOEA/D-DE, respectively.

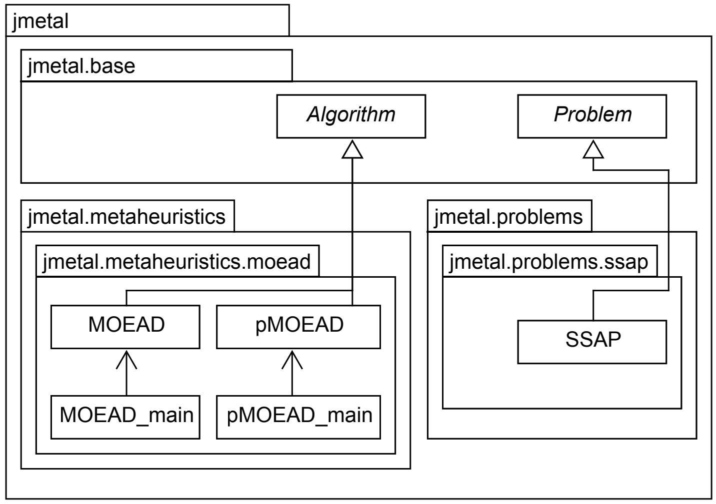

Figure 7 The jMetal UML Diagram with the SSAP Class Inside the "problems" Package, MOEAD and pMOEAD Classes Inside the "metaheuristics" Package

2.5.2.2 Performance Metrics. The mathematical model and performance metrics allow for the objective evaluation of various allocation algorithms. A performance metric is an evaluation measurement by which algorithms are compared and ranked. Some desired goals for measuring a Pareto front are generational distance (GD), spacing, and maximum spread [104]. However, the preferred quality indicators are Pareto dominance compliant because they are based on dominating individuals. The Pareto compliant quality indicators applied are hypervolume, epsilon indicator, R_2 indicator, and R_3 indicator [26].

The final utility indicators R_2 and R_3 are designed to measure the difference in the mean distance of the attainment surfaces, A and R [46]. The array A is an MOEA solution set, and R is the relaxed Pareto front. Simply put, these indicators reveal how far each solution is from the relaxed Pareto front. The closer the indicator is to zero, the closer the solution is to the front. The pMOEA/D-DE does better than MOEA/D-DE because its solution R_2 value is lower and R_3 value is the closest to zero.

Even though the true Pareto front (PF), PF_{true} is not obtainable, the relaxed Pareto front derived from simulations can serve as PF_{known}. The term "relaxed" refers to a Pareto front that is not the PF_{true} but can improve the evaluation metrics because PF_{known} is created in terms of true function evaluations. To obtain the relaxed Pareto front, it is better

to use a variety of MOEAs since no single MOEA has a proof of convergence to the true Pareto-optimal solutions. For an approach similar to the one used in Laumanns' approach, an archive of experimental solutions is kept from all multi-objective algorithms ran on the SSAP (e.g. SPEA2, NSGA-II, MOEA-D/DE, pMOEA-D/DE, PAES, OMOPSO, etc.) [65][30][124][62][61][99]. Unfortunately, SPEA2, NSGA-II, PAES, and OMOPSO were not able to produce enough points for full experimental testing; however, they are able to produce enough points to improve experiments involving the relaxed Pareto front, PF_{known}. This archive is used as the best relaxed Pareto front of all the experiments run in the tests.

2.6 Results and Analysis

All the experiments are run on AFIT's Linux cluster called Nordic. The computer architecture of this is listed in Table 6. The table depicts the number of nodes, the number and speed of the processors, the amount of memory per node, and the speed of the communication back-plane between nodes. It also displays the sum total of these attributes. The experiments are run with jMetal which supports parallel metaheuristics. Since this jMetal version can evaluate solutions in parallel, even traditionally non-parallel algorithms like MOEA/D-DE can take advantage of some parallel processing [83].

Table 6 Computer Architecture: Showing the Number of Nodes, the Number and Speed of the Processors, the Amount of Memory per Node, and the Speed of the Communication Backplane of AFIT's Nordic Linux Cluster

	Nodes	**Processors**	**Memory**	**Back-plane**
	10	16 x 2.3 GHz	4 GB	10 Gigabit
	2	32 x 2.2 GHz	16 GB	10 Gigabit
Total	12	224 processors	72 GB	-

2.6.1 Evolutionary Algorithm Tasker Results. In this section, the results of both the SP Tasker and the EAT approaches are compared. Table 7 provides a statistical analysis of the data. The experimental variability is very low, but each of the 50 runs has a unique seed that generates close, but unique, results as seen in Table 8. The first column of Table 8 is the seed used for that run. The remaining column headers start with SP or EA for the SP Tasker and EAT, respectively. Each column header also ends with a metric

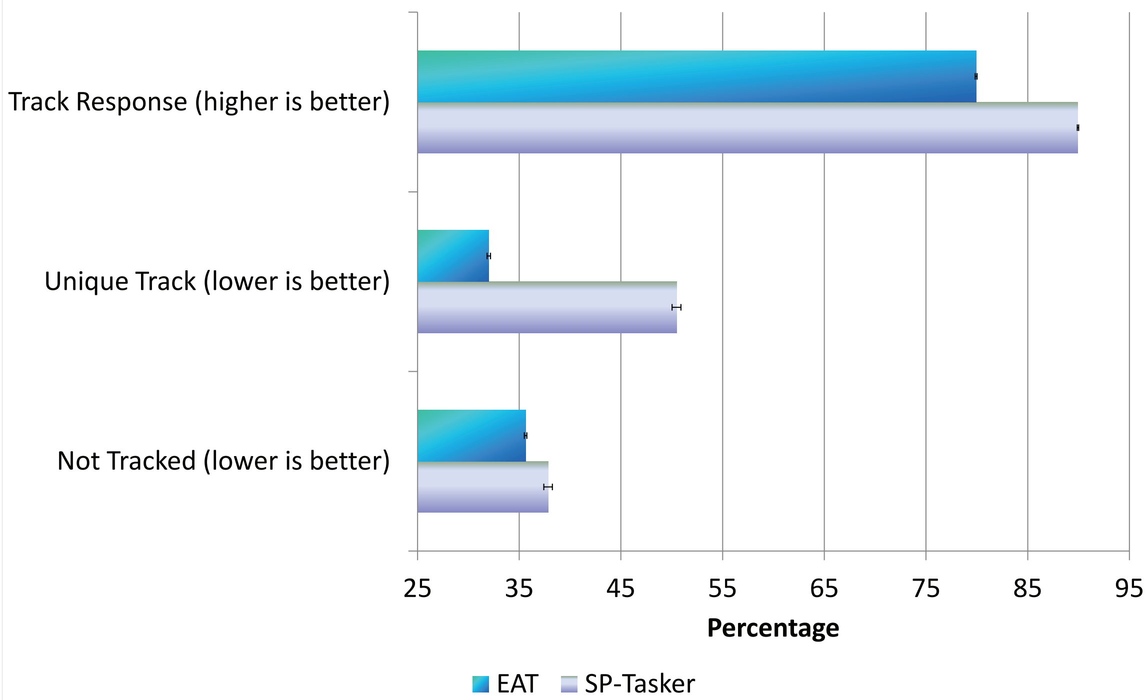

Figure 8 SP-Tasker and EAT: The Track Response Rate Mean (higher is desired), Unique-Track Percentage Mean (lower is desired), and Not-Tracked Percentage Mean (lower is desired) are shown as bars of fifty runs, potential range 0-100.

identifier NT (Not-Tracked Percentage), UT (Unique-Tracked Percentage), and TR (Track Response Rate).

Table 7 SP & EAT Statistical Data: The Average, Standard Deviation, Median, Maximum, Minimum, and T-Test (two sets significantly different)

	Not Tracked		Unique Track		Track Response	
	SP	EAT	SP	EAT	SP	EAT
Average	37.839	35.628	50.466	32.008	89.943	79.930
Std. Dev.	0.853	0.250	0.882	0.342	0.140	0.188
Median	37.888	35.640	50.408	32.010	89.930	79.918
Max.	39.315	36.135	52.360	33.015	90.261	80.410
Min.	35.950	35.195	48.980	31.315	89.623	79.462
T-Test	1.984E-22		1.393E-63		8.319E-82	

The average for all those runs is presented with 99.9% confidence intervals in Figure 8. A confidence level of 99.9% indicates a corresponding significance level of 0.01% or p-value under 0.01. This confidence level has been used in all cases comparing EAT to SP Tasker, which means that the differences are unlikely to have occurred by chance with a probability of 99.9%. The t-test and box plots reinforce the confidence intervals conclusion

Table 8 EAT and SP Tasker Raw Data Table: The Full Result of All Fifty Runs

SEED	SPNT	EANT	SPUT	EAUT	SPTR	EATR
484486	39.315	35.720	48.980	31.940	90.050	80.119
939958	38.310	35.360	49.935	32.255	89.851	80.109
87429	37.795	35.780	50.685	32.270	89.920	80.142
82013	36.355	36.040	52.025	31.455	89.861	80.122
12619	37.425	35.715	51.080	31.810	89.923	79.965
549597	37.485	35.755	50.875	31.795	89.939	79.877
69095	37.865	35.735	50.520	32.240	89.938	79.811
317481	37.645	35.590	50.770	31.815	89.808	80.117
940194	39.135	36.010	49.270	31.705	90.100	79.895
757261	36.115	35.735	51.900	31.995	89.982	80.072
122379	38.460	35.540	49.925	31.930	89.919	80.080
818529	37.530	35.545	50.890	31.945	90.096	80.112
521407	37.800	35.440	50.360	32.395	89.903	80.002
17554	35.950	36.135	52.355	31.315	89.768	79.784
761961	39.140	35.675	49.025	32.105	89.980	80.102
240309	38.185	35.985	49.925	31.675	89.709	79.880
957083	38.805	36.020	49.560	31.320	89.989	80.215
924668	37.955	36.020	50.230	31.575	90.045	79.716
679060	38.300	35.195	50.170	32.515	89.876	79.770
550037	38.850	35.735	49.375	32.140	89.969	79.761
475376	39.060	35.710	49.105	32.365	89.994	80.147
716207	37.595	35.355	50.530	31.990	90.247	79.860
362016	38.410	35.195	50.075	33.015	89.770	79.816
324448	37.825	35.655	50.425	31.955	90.084	80.160
9673	37.905	35.480	50.620	31.935	89.906	79.735
323681	36.495	35.345	51.840	32.220	89.910	79.837
602484	38.985	35.570	49.260	32.145	89.900	79.462
196921	37.815	35.695	50.365	32.075	89.781	79.929
707985	36.860	35.545	51.445	32.095	90.149	79.602
335060	37.455	35.315	50.560	32.205	90.170	80.064
643501	37.645	35.735	50.610	31.955	90.020	79.801
876769	38.895	35.990	49.150	31.650	89.996	80.128
278613	38.020	35.400	50.555	31.940	89.623	79.939
638099	38.495	35.245	49.895	32.375	90.092	79.571
574294	38.470	35.455	49.660	32.005	90.261	80.099
769655	38.420	35.230	49.985	32.895	89.795	79.852
353646	36.860	35.605	51.550	32.035	89.880	79.790
572707	37.280	36.030	51.050	31.460	89.759	79.945
331625	38.265	35.255	50.155	32.440	89.711	79.724
990813	36.805	35.415	51.720	32.020	89.941	80.012
634839	37.870	35.415	50.400	32.185	89.759	79.915
136460	36.715	35.485	51.610	32.080	89.918	80.173
328564	38.230	35.750	50.055	32.110	90.089	79.895
481641	36.005	35.625	52.360	32.015	89.918	79.783
575691	36.880	35.660	51.410	31.625	89.836	80.410
159425	37.930	35.990	50.415	31.550	89.835	79.922
298980	38.110	35.825	50.060	31.895	90.042	79.691
15498	38.675	35.465	49.575	32.145	89.943	79.900
749041	37.240	35.720	51.215	31.750	90.043	79.942
392783	38.305	35.495	49.800	32.080	90.162	79.744

that these data sets are significantly dissimilar. The raw data in Table 8 shows how the data has very low variability, which results in high confidence.

Each set of data is tested for normality to prepare for the significance tests, and the normality tests show that all data are normally distributed [97]. For each significance test or t-test, the null hypothesis is equality between the SP and EAT data sets. For example, when comparing SP Tasker Not-Tracked Percentage (SPNT) with EAT Not-Track Percentage (EANT), the null hypothesis would be SPNT=EANT. The null hypothesis is tested to determine if there is a statistically significant difference between the two sets of data. Since the value is very low ($<< 0.05$) for all three pairs of data, each null hypothesis is rejected. Therefore, the t-test is evidence that each set of data are different, for comparison of the SP Tasker and EAT. To enumerate: the SP Tasker track response rate data is statistically different from the EAT track response rate; the SP Tasker unique-track percentage data is statistically different from the EAT unique-track percentage; and the SP Tasker not-tracked percentage data is statistically different from the EAT not-tracked percentage.

The SP Tasker had a higher track response rate than the EAT, as seen in Figure 9. The track response rate is the only metric where the SP Tasker performed better than the EAT. The SP Tasker mean is 89.943 compared to 79.930 for the EAT. This is to be expected, because the SP Tasker only focuses on the probability, while the EAT strives toward a greater goal of balancing probability and priority.

When it comes to redundancy and accuracy, the EAT does a “better” job at the unique-track percentage as seen in Figure 10. The EAT mean is 32.008, and the SP Tasker mean is 50.466. A greater percentage of RSO are tracked by more than one sensor. This ensures that more RSO are tracked in case a sensor fails or must yield to the higher priority of the missile warning system.

The EAT has a lower not-tracked percentage, which is better than the SP Tasker, as seen in Figure 11. The EAT mean is 35.628 compared to 37.839 for the SP Tasker. The EAT shows the ability to obtain at least one track for more RSO.

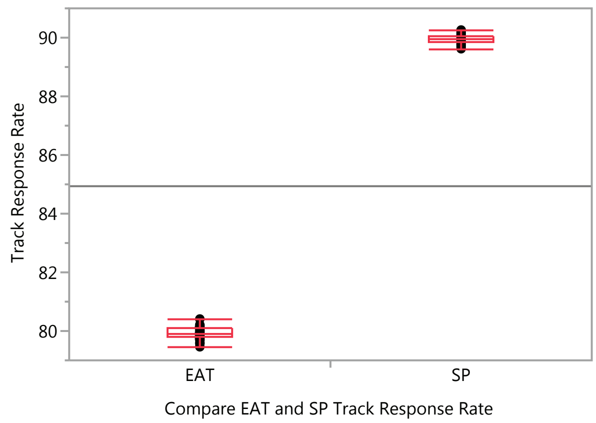

Figure 9 Track Response Rate Box Plot: The plot presents the difference in data sets where the SP performs better.

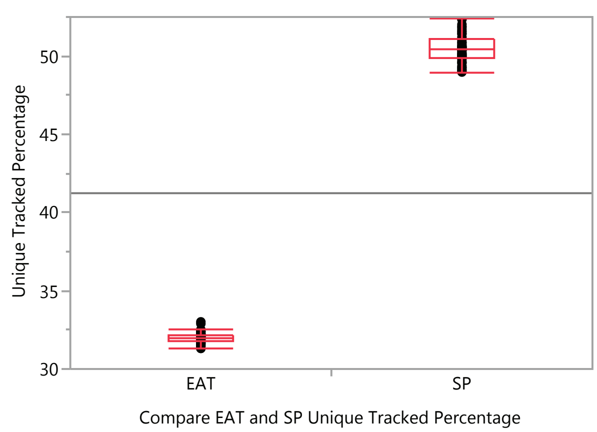

Figure 10 Unique-Track Percentage Box Plot: The plot presents the difference in data sets where the EAT performs better.

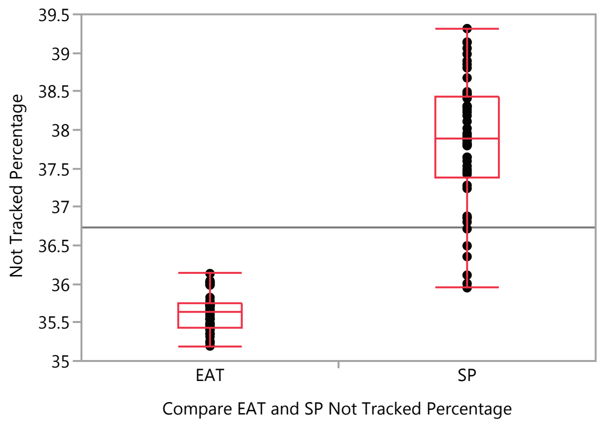

Figure 11 Not-Tracked Percentage Box Plot: The plot presents the difference in data sets where the EAT performs better.

The EAT ran on average approximately fifty times faster than the SP Tasker. The longest single run time of the SP Tasker is almost nineteen hours. This time is like the SP Tasker Performance experiment by Concetto Giuliano and Francis Chun et. al. [40]. Their results on a cluster of sixteen CPUs achieved a time of just less than twenty-four hours; their results on a cluster of fifty CPUs achieved a time of approximately twelve hours. The run time measurement does not matter as much as the other measurements unless the time violates the 24-hour cycle threshold, but as the amount of RSO grows in the coming years the EAT has a large margin under the 24-hour threshold.

2.6.2 Multi-Objective Results. For experimentation in jMetal, the SSAP implementation has both the previously defined objectives and constraints from the programming model defined in Section 2.3.1. The two MOEAS (MOEA/D-DE, pMOEA/D-DE) solve the SSAP, producing the following results.

2.6.2.1 Additional Variables for Constraints. For problems like the SSAP that are restricted by several constraints, it is hard to find feasible solutions because the constraints restrict the problem for small sets of variables [69]. In initial experimentation with a small number of RSO (< 4000), the algorithms struggle to form a Pareto front of solutions. To produce a highly populated Pareto front, more variables need to be added. In Figure 12 the additional variables of more RSO allow the EAs to find more feasible

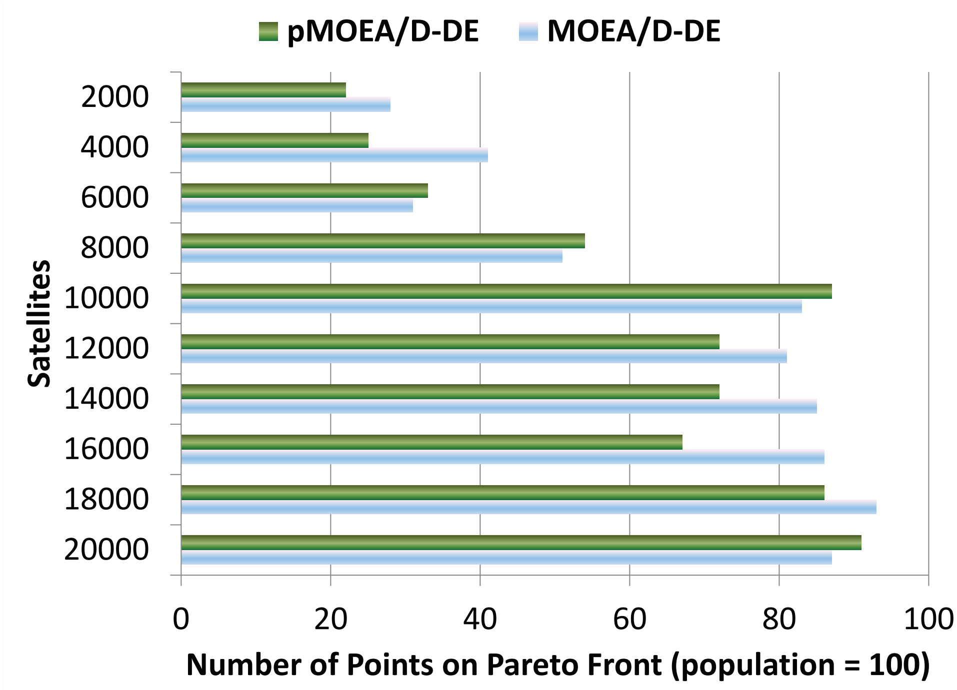

Figure 12 MOEA/D-DE and pMOEA/D-DE Points on Pareto Front: The points on the Pareto Front increase as the number of input satellites increases.

solutions resulting in more points on the Pareto front. The problem lends itself to many RSO. For the 20,000 trackable RSO, the EAs can find solutions where nearly all of their points are on the Pareto front!

2.6.2.2 Pareto Front. Figure 13 shows the two approximate Pareto fronts from each algorithm's solution. The points shown are the best non-dominated points generated by more than forty runs of each algorithm. The decision makers must choose a solution from the Pareto front to use. The scatter plot of points shows that the pMOEA/D-DE can generate better points for most of the area, but not the lower-right side of the plot where MOEA/D-DE achieves better points. If the decision maker wants to favor the probability of tracking the most RSO, they would choose a point from the upper-left side. Conversely, if the decision maker wants to favor the priority of tracking more important RSO, they would choose one of the points in the lower-right side. For a balanced approach, the decision can be one of the solutions in the middle.

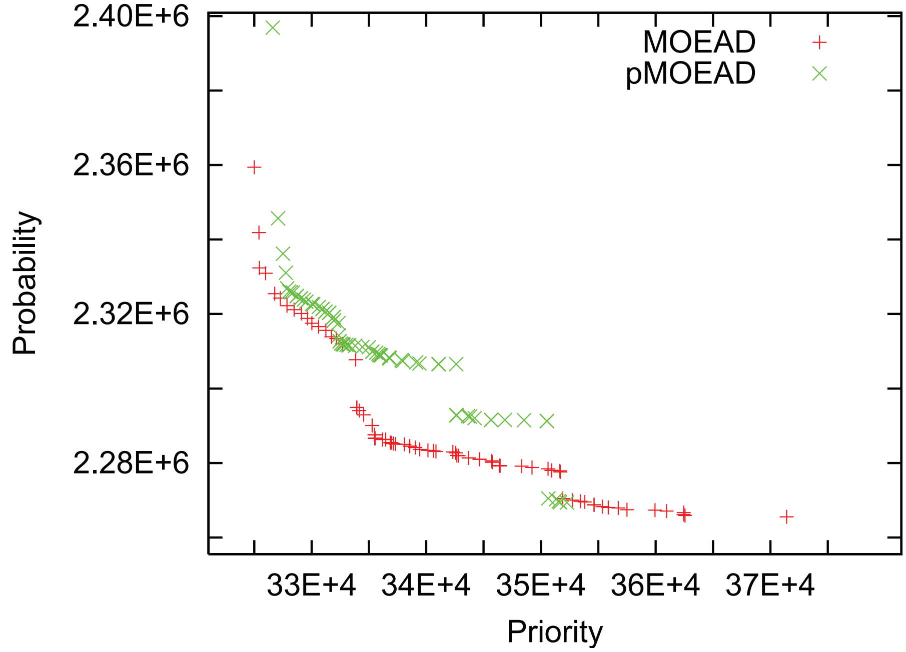

Figure 13 MOEA/D-DE and pMOEA/D-DE Pareto Front: Probability on the vertical axis is competing with priority on the horizontal axis for the SSN Resources while the MOEAD and pMOEAD solutions are on the plot.

Table 9 Quality Indicators: Mean and Standard Deviation for Each Indicator (Epsilon, Hypervolume, Generational Distance, and Spread) and Each Algorithm (MOEA/D-DE & pMOEA/D-DE)

Indicator		MOEA/D-DE	pMOEA/D-DE
Epsilon	*Mean*	$3.05e+05$	$3.42e+05$
	Std. Dev.	$7.5e+05$	$7.9e+05$
Hypervolume	*Mean*	0.8767	0.8907
	Std. Dev.	0.1052	0.0959
GD	*Mean*	0.0172	0.0173
	Std. Dev.	0.0099	0.0085
Spread	*Mean*	0.9563	0.9490
	Std. Dev.	0.0591	0.0471

2.6.2.3 Quality Indicators.

The quality indicators used to measure each algorithm's performance are epsilon, the hypervolume, the generational distance (GD), and the spread. These quality indicator values can be found in Table 9.

The epsilon indicator measures the translation distance between two approximate sets. The epsilon indicator calculates the smallest amount that must be used to translate

set, A, so that every point in set B is covered. If the mean epsilon indicator is greater than one, then both contain solutions not dominated by the other set [26]. If the epsilon is one, then both sets are the same Pareto front approximation. If the epsilon less than one, then all points in set B are dominated by a point in set A. However, the desired outcome is still the smallest epsilon value, which is achieved by the MOEA/D-DE that has the smallest epsilon.

The hypervolume measures the volume, or in this case, the area of the two-dimensional dominated portion of the objective space. The pMOEA/D-DE has a slightly higher hypervolume than MOEA/D-DE, but the numbers are within one standard deviation of each other.

The GD is the average distance of the known algorithmic front to the relaxed Pareto front. The MOEA/D-DE produced a slightly larger value for GD, meaning that it is further away from the relaxed Pareto front.

Finally, the spread or spacing is a metric that describes how the vectors in the known front are spaced. It measures the distance between neighboring vectors in the known front. The spread value is zero where all vectors are evenly spaced. The pMOEA/D-DE value is closer to zero, or more evenly spaced than the MOEA/D-DE value, but again they are still within one standard deviation of each other.

The R_2 and R_3 indicators from the SSAP experiments are in Table 10. As explained earlier in Section 2.5.2.2, the R_2 and R_3 values are desired to be closer to zero, which means the pMOEA/D-DE performed better.

Table 10 R_2 and R_3 Indicators: For MOEA/D-DE and pMOEA/D-DE

	MOEA/D-DE	pMOEA/D-DE
R_2	174552	85407
R_3	0.0683	0.0334

The pMOEA/D-DE slightly outperformed MOEA/D-DE for five out of six of these multi-objective quality indicators (epsilon, hypervolume, GD, spread, R_2, and R_3), but the values are not different enough to be statistically significant.

2.7 Conclusion

This chapter adds priority into the SSAP model and presents a novel solution technique. This solution is the novel EAT which is a single objective evolutionary algorithm. Both the SP Tasker algorithm and the EAT are implemented to solve the SSAP. The EAT can assign more RSO to be tracked and have more RSO tracked by multiple sensors. On the other hand, the SP Tasker has slightly higher success at tracking the RSO that receive sensor tracking assignments. The EAT run time is much quicker, which could allow for a reduced cataloging cycle and/or have room for expansion of the current catalog.

The EAT algorithm holds great promise to update the space tracking system. Potentially the EAT or a similar evolutionary algorithm can be the next tasking algorithm. With further refinement, the EAT can conceivably improve its track response rate results to become better than the SP Tasker.

The second solution technique is applying two MOEAs to the SSAP. The two MOEAs are pMOEA/D-DE and MOEA/D-DE. They did well when tested against the SSAP. Empirical results suggest the highly restrictive constraints of the SSAP can be met by additional variables. In Section 2.6.2.3, the pMOEA/D-DE is shown to be slightly better than the MOEA/D-DE in all but one of the tested quality metrics. Examination of the quality indicators shows the difference between the MOEAs. The MOEA experiments are useful because they show a computationally efficient and quality MOEA approach to the multi-objective SSAP. They also give the decision maker more flexibility in deciding which solution to choose.

The experimental results provide a quantitative basis for improved tracking, leading to decreased risk of collision. Further experimental testing could measure the not-tracked percentage, unique-track percentage, and track response rate of the multi-objective algorithms. This information could validate the case for a new evolutionary algorithm approach.

Further research could be done to develop new approaches, using both single objective and multi-objective solutions that use priority to track more important RSO. The current system considers expensive operating payloads and broken pieces of debris equally. A

new system that increases the track locations and orbits of higher priority RSO would better serve the purpose of protecting space assets. With any new development, priority can be a key piece of measurement. A metric such as the number of priority satellites not tracked would be a good addition to the assessment. Another multi-objective option using a commercial off the self product called ACE Premier Intelligent Resource Optimizer (AceIRO) should be a branch of future work. Triet Tran used this commercial off the self product, AceIRO, to run a multi-objective resource optimization to task sensors in the SSN [105]. Their experiment only used hundreds of tasks, but further research could scale this up to a more realistic size.

The results from this study and further research could provide a more an effective way to detect collisions like the one in 2009 and the near miss with the International Space Station in 2012. With each collision the problem grows worse and closer to a catastrophic Kessler Syndrome situation. Evolutionary algorithms can be the answer to avoid such a catastrophe. A novel solution called EAT shows improved sensor allocation performance for all but one metric, and the SSAP model improves on current/previous systems.

3. *Multi-Objective Evolutionary Algorithm Tasker*

3.1 Introduction

Many government agencies are trying to tackle the problem of space debris. Coordination is key because the limited assets of separate states are better when added to a multiple pronged approach to address the issues [12]. Additional satellite constellations should not be added to Low Earth Orbit (LEO) unless careful planning and resources are used to mitigate the space debris. Companies like SpaceX, OneWeb, Boeing, and others are planning large satellite constellations in the already congested LEO region. To get an idea of how much this would impact the current state of the space catalog here are the numbers: The United States Space Surveillance Network (SSN) is currently tracking about 44,000 objects with about 19% being operational satellites, 14% being old rocket bodies, and 67% being debris. The set of operational satellites is dramatically larger in the last decade from about 5% because of massive increase in satellite launches. SpaceX plans to add at least 4,425 satellites into LEO by 2024 [14, 18]. In total, the number of additional satellites proposed by these companies is between 14,041 and 15,601. This all increases the chance of a costly collision between Resident Space Objects (RSO).

Although these companies have plans to minimize their space debris, there will be a very large increase in the catalog in a short amount of time. Some of these large-scale satellite constellations have already been approved by the Federal Communications Commission. These companies all have plans for minimizing space debris. The literature indicates that ground or space-based lasers could be used to knock debris into a lower orbit or even a decaying orbit [88][98]. Although this presents an interesting approach, currently the application of such technology is prohibitively expensive. Limiting the number of RSO is the best practice [113].

These issues are not limited to LEO. Satellites in Geosynchronous Equatorial Orbits (GEO) are at risk of collision from debris as well, even though these satellites tend to have deep space orbits. Space assets are in international space. Yes, there are assets in GEO over specific ground locations belonging to that territory, but most of the time even geosynchronous satellites can service more than one continent, let alone multiple countries

with its area of service. Currently, many countries have interests and property in space. Data-sharing can help improve Space Situational Awareness (SSA) by taking advantage of all the global resources [94]. Global leaders have worked together on some international efforts like the European Space Agency, Space Data Association, Secure World Foundation, etc. These organizations have differing technical abilities. In an environment where radar sensors, optical sensors, etc. are present as SSA resources, it is important to take advantage of each sensor's unique abilities and strategic locations. The hybrid sensor situation lends itself to evolutionary optimization because of the vast array of states to explore.

There are many hybrid MOEAs [15, 117, 121]. The algorithm design shown in this chapter follows the same pattern as that of this algorithmic class. Characteristic techniques include point mutation and/or capitalizing on the best individuals. The novelty of the new algorithm starts with the application space operations, especially the SSN and other combined capabilities of the algorithm. Satellite applications are important to the future, and systems based on satellites must be dependable. Work has been done to make satellite applications more reliable [118]. While there are many hybrid MOEAs, few use an evolutionary strategy or genetic algorithm as a part of their approach. The closest to Multi-objective Evolutionary Algorithm Tasker (MEAT) is an algorithm which uses an evolutionary strategy one part design [121]. Altogether the approach used in the MEAT is a novel line of research.

3.1.1 Short Survey. The previous chapter developed the Evolutionary Algorithm Tasker (EAT) for experimentation and research. This chapter augments the EAT into the Multi-objective Evolutionary Algorithm Tasker (MEAT) to further this line of research. In preview, this chapter discusses the model and implementation of multi-objective methods to specific scenarios of RAP for orbital information. First is the background knowledge in Section 3.2 that includes basic definitions of the SSN, genetic algorithms, and summarizes the sensor resources. Section 3.3 categorizes the multi-objective model as a RAP. Section 3.4 presents the optimal approach and application of the problem, as well as the goals and measurable objectives of experiments. Section 3.5 covers the experimental output and performs statistical analysis of the data. Finally, Section 3.6 summarizes the key

points, offers concluding remarks, and advocates ideas for future work. For the reader's ease of understanding and quick look up, the following Table 11 provides a list of acronym definitions.

Table 11 List of Acronym Definitions Alphabetically

AFSIM	Advanced Framework for Simulation Integration and Modeling
DM	Decision Maker
EAT	Evolutionary Algorithm Tasker
ESSS	European Space Surveillance Sensors
GD	Generational Distance
GEO	Geosynchronous Equatorial Orbit
HV	Hyper-Volume
ISON	International Scientific Optical Network
LEO	Low Earth Orbit
MOEA	Multi-Objective Evolutionary Algorithm
MEAT	Multi-Objective Evolutionary Algorithm Tasker
MOES	Multi-Objective Evolutionary Strategy
MOEA/D	Multi-Objective Evolutionary Algorithm Based on Decomposition
MOGA	Multi-Objective Genetic Algorithm
MOGLS	Multi-Objective Genetic Local Search
NP-Hard	Nondeterministic Polynomial-time Hard
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PF	Pareto optimal Front
RSO	Resident Space Objects
RAP	Resource Allocation Problem
SSAP	Sensor Satellite Allocation Problem
SSA	Space Situational Awareness
SSN	Space Surveillance Network
SPEA2	Strength Pareto Evolutionary Algorithm
ZDT	Zitzler, Deb, Thiele

3.2 Background

The current environment in space surveillance is complicated. Being aware is important. When considering possible solutions understanding the environment in space is key. Figure 14 demonstrates the congestion in the exosphere and lower regions of space [81, 48]. Knowing the objects orbiting is only part of the battle, because people are depending on the information gained from satellite systems. With many international players and many possible algorithmic solutions to the SSAP, it is best to know the history and current research in these areas.

The focus of this chapter is two-fold. First, the SSAP is discussed regarding the development and use of a jMetal study including a problem set and test suite. Second, a fundamental comparison is made between MOEAs with benchmarks and quality indicators. This image shows a global problem that needs a global solution.

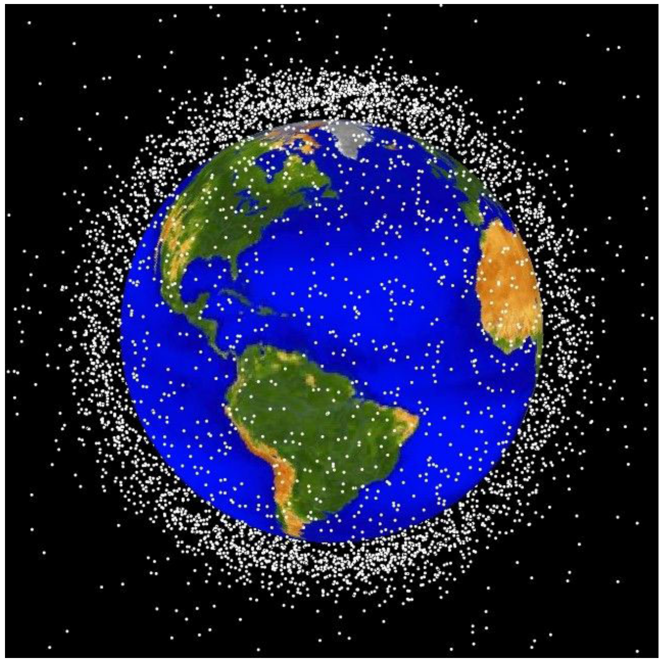

Figure 14 Image showing low earth region where the debris is the most concentrated.

3.2.1 Overview of Space Surveillance Networks. Much more collaboration could be done without compromising each country's security concerns. Some countries have many space surveillance assets, while others have few or even none. Without the ability to see into space well, these states need to rely on others to launch satellites or risk the potential for collision before the satellite even gets into an operational orbit. Since 2016 Australia has really increased their capability of tracking space assets by conducting experiments and using various sensors track RSO [78]. RSO are all objects that are orbiting earth, including active systems and space junk (consisting of orbital debris, old rocket boosters, etc.) The Canadian Space Surveillance System has a satellite, Sapphire, that adds space-based, sensor-tracking abilities [74]. Sapphire is also feeding information to the United States SSN. The SSN is the most extensive space surveillance system with ground-based sensors in all four quadrants of the global. Europe, China, and Russia have their respective systems as well [78]. India, Japan, Kazakhstan, Korea, and Ukraine have smaller networks also [108, 20].

Table 12 summarizes open and available data for the sensors that could be tasked worldwide, and the table is not meant to be a comprehensive diagram of all sensors [108].

Table 12 Satellite Tracking Sensors

Organization	Phased Array	Optical	Radar	Other
China	3	9	-	-
ESSS	-	12	8	1
ISON	-	28	-	-
Russia	15	2	-	-
United States	20	24	16	11
Other	3	13	3	-

It shows a large set of passive sensors. The total count of passive sensors is probably over 250 [9]. Passive sensors track RSO that are not actively participating in tracking. Another group of sensors are called active sensors. These active sensors track satellites that are designed to aid in the tracking by reflecting the signal or generating a response signal. A few examples of these are satellites equipped with transponders or equipped with mirrors to reflect laser-ranging signals. In Table 12, ESSS stands for European Space Surveillance Sensors, and ISON stands for International Scientific Optical Network. Nearly all these sensors do many other important tasks besides observing orbital debris.

The SSN tracks current RSO and catalogs them by recording the state of orbital objects. The current Space-Track catalog has a current listing showing the present state of RSO [18]. Historically, the catalog holds tens of thousands of items, but many have already decayed into the atmosphere and burned up. The catalog only contains objects that are trackable, generally greater than 10cm in diameter [112]. Many millions of objects are smaller than the generally observable size. The small objects still have the potential to damage assets despite their size.

3.2.2 Metaheuristic Techniques. Metaheuristics is a class of approximation algorithms. They come into play often because many problems are to computationally complex to obtain an optimal answer in a reasonable amount of time. Metaheuristics provide "acceptable" solutions in a timely manner [103]. As formerly noted, the SSAP is a specific form of the RAP. Because the SSAP involves multiple sensors, it is a multi-RAP. Toshihide Ibaraki and Naoki Katoh proved that this problem is nondeterministic polynomial-time hard (NP-Hard) [55]. This means that no one can expect to find an optimum solution in polynomial time; therefore, a polynomial time heuristic algorithm is needed.

One of many possible heursitics is the Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D). Decomposition is a simple strategy in multi-objective optimization where the problem is broken up into sub-problems. The MOEA/D uses such a strategy. Once the magnitude optimization sub-problems are created, each problem is solved and optimized concurrently [67, 122]. Each sub-problem is improved by using information from its neighboring sub-problems. The neighborhood technique instead of a global technique allows the MOEA/D to have lower computational complexity at each generation than Multi-objective Genetic Local Search and non-dominated sorting genetic algorithm II (NSGA-II) [30, 56]. The neighborhood size of MOEA/D was experimentally investigated considering scalability and the sensitivity. Zhang and Li found that the computational cost linearly scaled up when the number of decision variables increased. However, the regular MOEA/D does not work well with highly constrained problems like the SSAP [31].

A special version of Constraint-Handling NSGA-III works toward the purpose of maintaining a population with more feasible solutions [31]. Focusing on constraints is good because the infeasible solutions are useless in the end. Many MOEAs consider Pareto domination during selection, but the constraint-domination principle values feasible solutions, while still considering the usual domination principle [124, 30].

NSGA-II has a smaller computational complexity than NSGA-III [31]. The NSGA-II procedure is run later for the results and analysis [30]. NSGA has many versions like NSGA-II$_{ss}$, aNSGA-II, and rNSGA-II [36, 45, 83]. NSGA-II$_{ss}$ is an augmentation to use steady-state selection instead of generational selection which the original applies [36]. After evaluating all these versions in the literature, the NSGA-II$_{ss}$ was chosen as the most likely to do well on the SSAP specifications.

Strength Pareto Evolutionary Algorithm 2 (SPEA2) has a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method [124]. SPEA2 focuses on two main aspects of the algorithm the mating selection and environmental selection. In mating selection, the SPEA2 employs a partial elitist strategy of keeping individuals with the best fitness values and incorporating the density information to avoid groups all packed in the same general area of the search space. For the environ-

mental selection, SPEA2 also relies on making sure that the boundary solutions are kept. SPEA2 reaches better results on all considered problems than several other MOEA [124]

A basic MOGA does a single selection in which the entire selected population is treated the same, but this algorithm has two different ways of selecting individuals. MOES performs a mutation on only the best individuals. [6]

Many real-world optimization problems are multi-objective, meaning they have two or more competing objectives. In this case, no single solution can optimize all the objectives simultaneously. Instead, Pareto optimal solutions are achievable for reasonably sized problems that try to optimize all objectives at once, given the constraints. The Pareto optimal front (PF) is the set of all the Pareto optimal solutions in the objective space. The PF spread of solutions are interesting to the decision maker (DM) for practical purposes [26]. The DM chould choose which solution to use. MOEAs are a good fit for multi-objective problems because they can produce an approximate PF very quickly in a single round [109]. They are frequently strong to latent objective function traits [26, 66]. Evolutionary algorithms approaches have been found to be advantageous for automatic processing applications with an abundance of data. MOEAs do well at refining turbulent data via excellent parameter selection, allowing significant information to be found [80, 79].

3.3 Problem Definition

The EAT is a single objective algorithm that has shown promising results with the specific RAP called SSAP [43]. Some of the same techniques in the EAT are used in the multi-objective version called MEAT, such as the ES and GA strategies. This chapter investigates why the MEAT performs well on the RAP and investigates whether it will work well on other problems. In order to figure out this inquiry, MEAT is compared to other well-known MOEAs and run against the same set of benchmark problems.

The specific benchmark problems selected are the Zitzler, Deb, Thiele (ZDT) problem set [123]. The following are reasons why the ZDT benchmarks are selected. The originators of this problem set designed them to present MOEAs with different problem features. This way it is possible to identify whether the MEAT is successful or not on each kind of problem.

These benchmark problems are common in the literature. They are readily available in the jMetal framework to be implemented straight away.

The results only reflect that the MEAT works effectively on the benchmarks tested. To know if the MEAT works well on other problems beyond the ones tested here, they will have to be tested individually because of the No Free Lunch Theorem [116]. The MEAT is developed to present a novel approach to solving MOEAs with the combination MOGA and MOES techniques.

To give a brief context to the problem, the SSAP consists of a set of sensors and a set of RSO. The number of sensors is m and n represents the number of RSO. The set of observations for a single track can be recorded during one pass of the satellite over the sensor. Each sensor's field of view is analyzed, resulting in a daily pass and corresponding track probability for each time the satellite enters a sensor's field of view. Each satellite has a set of daily passes or opportunities to be tracked where $d_{ij} \in D$, and where d_{ij} is a positive integer and denotes an opportunity for the i^{th} sensor to track the j^{th} satellite. For example, if satellite, $j = 18$, passes into the sensor, $i = 4$, field of view four times in one day, the amount for that daily pass would be $d_{18\ 4} = 3$. Likewise, if the satellite does not pass over the sensor in that day, the corresponding daily pass value would be zero.

Each pass has a track probability, which is determined by range and radar cross section. When considering how the range influences probability, an object being further from the sensor corresponds to a lower probability of receiving a good signal. The radar cross section is the object's ability to reflect a radar signal back to the receiver. The ability to reflect the signal impacts track probability settings as well. The track probability $p_{ijk} \in P$ denotes the probability of the i^{th} sensor's ability to track the j^{th} satellite on the k^{th} daily pass where $1 \leq k \leq d_{ij}$.

Every satellite has a *priority* where $o_j \in O$ denotes the *priority* of the j^{th} satellite. The *priority* range is $1 \leq o_j \leq 5$ with 1 being the most important. *Priority* is mainly based on the significance of the satellite and potential loss in the event of a collision. For instance, active RSO are a higher category, such as 1, 2, or 3, and inactive RSO or debris

are in categories such as 4 or 5. The five categories are adequate to address the diversity of importance among RSO [87][23].

The goal is to maximize both Equations 3 and 7 from Chapter 2 subject to the constraints. They are copied to Equations 9 and 10 below for convenience.

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{d_{ij}} p_{ijk}x_{ijk} \tag{9}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{d_{ij}} \frac{x_{ijk}}{o_j} \tag{10}$$

The model defined in the previous literature has the full formal problem definition including the long list of constraints [43]. The single objective SSAP is transformed into the mutli-objective SSAP (MOSSAP) by Equations (9) and (10). The first objective in Equation (9) is to maximize the summed *probability* of tracking as many RSO as possible. The limited sensor resources do not allow observers to track all the RSO, so the objective to maximize the *probability* of cataloging as many RSO as possible. To put it another way, the objective maximizes the probability of tracking all the RSO that are scheduled to be tracked. The higher the probability, the greater likelihood additional RSO will be tracked. The priority objective in Equation (10) is to maximize the ratio of allocated RSO to the corresponding priority. The allocated tracks x_{ijk} are divided by the priority o_j. The summation is designed to ensure that the highest priority RSO affect the objective value more than the lower priority RSO. For example, if $x_{ijk} = 1$ and $o_j = 2$, then values would translate into a larger impact on the total sum than the other scenario of $x_{ijk} = 1$ and $o_j = 5$ (i.e. $\frac{1}{2} > \frac{1}{5}$). Accordingly, each of these values is summed up where the higher priority RSO has more significance.

3.4 Experimental Design

The approach of using MOEAs on the SSAP is preferred because MOEAs have been shown to perform well on RAP [84, 86]. The well-known algorithms like SPEA2, and

NSGA-II are used as a baseline comparison. Both algorithms have been tested and verified to perform well on many problems including RAP [111].

Many research efforts only used terrestrial observers which was a safe assumption because there was only one orbital observer, the International Space Station, taking a few observations [43]. However, recent space-based surveillance is not as easily ignored. Now that there are more orbital observers and they are growing in number quickly, the simulation considers both terrestrial and orbital observers.

3.4.1 Multi-Objective Evolutionary Tasker Algorithm Design. MEAT mating selection is important because those are the individuals used for offspring production. Mating selection needs to be done carefully to avoid problems. For example, if mates are selected based on fitness alone, then the search could get stuck at local maximum instead of exploring the entire search area.

The main principle of the MEAT is that a good search algorithm should explore the solution space and exploit the good individuals it finds. The MEAT is designed with an MOES part to exploit the best solutions and a MOGA part to explore the solution space. The combination results in the ability to find good solutions while avoiding the drawbacks of these techniques used separately. The selection operator employs these elitist and exploration methods.

3.4.2 Performance Metrics and Quality Indicators. Pareto Compliant Quality Indicators consist of the error ratio, hyperarea ratio (hyper-volume), epsilon indicator, and utility R_1 and R_2 indicators [26].

Generational Distance (GD) [119] measures the closeness of the solutions to the relaxed PF. The closer the individuals are to the PF the better. Of course, finding solutions on the optimal true PF would be ideal, but that is not always possible. GD will calculate how close the overall solution approaches the PF.

The Hyper-Volume (HV) measures both closeness and diversity. It measures how close the resulting solutions are to the true PF, or in this case, the relaxed PF. HV also measures the diversity of the solutions by analyzing the individuals to see their similarities

and differences. The fact that both closeness and diversity are measured with HV makes it a Pareto compliant quality indicator. Thus, HV is the preferred measure.

The *spread* is a metric concerning the spacing of solutions. A PF is less useful if the solutions are all bunched together. Preferably the solutions would be spread out to cover the objective space uniformly. The spread is a measure quantifying how well the PF is evenly spaced.

Regarding the decision maker (DM) and the process of selecting a solution, it is a probability versus priority decision. For example, if the DM wants higher priority satellites tracked more often on a given day, the DM will choose a solution lower on the PF. If the DM wants a balanced solution, the DM will select a solution in the middle. If the DM prefers tracking a larger number of RSO, then the DM would pick a solution toward the left side of the PF.

3.5 Results and Analysis

Table 13 SPREAD. Mean and Standard Deviation

	NSGAII	SPEA2	MEAT
ZDT1	$3.30e-01_{2.2e-02}$	$3.20e-01_{1.5e-02}$	$3.79e-01_{4.2e-02}$
ZDT2	$3.75e-01_{4.9e-02}$	$3.45e-01_{3.9e-02}$	$3.28e-01_{5.3e-02}$
ZDT3	$7.46e-01_{1.3e-02}$	$7.28e-01_{2.0e-02}$	$6.76e-01_{2.0e-02}$
ZDT4	$3.28e-01_{2.8e-02}$	$8.76e-01_{5.1e-01}$	$4.37e-01_{3.4e-02}$

Table 14 SPREAD. Median and Interquartile Range

	NSGAII	SPEA2	MEAT
ZDT1	$3.39e-01_{4.2e-02}$	$3.21e-01_{2.9e-02}$	$3.64e-01_{8.1e-02}$
ZDT2	$3.78e-01_{9.8e-02}$	$3.63e-01_{7.3e-02}$	$3.22e-01_{1.1e-01}$
ZDT3	$7.39e-01_{2.4e-02}$	$7.39e-01_{3.5e-02}$	$6.71e-01_{3.8e-02}$
ZDT4	$3.17e-01_{5.2e-02}$	$9.70e-01_{1.0e+00}$	$4.27e-01_{6.5e-02}$

Table 15 GD. Mean and Standard Deviation

	NSGAII	SPEA2	MEAT
ZDT1	$2.18e-04_{1.4e-05}$	$4.01e-04_{8.9e-05}$	$3.62e-04_{1.4e-04}$
ZDT2	$1.42e-04_{5.2e-05}$	$2.40e-04_{4.5e-05}$	$4.46e-04_{3.2e-04}$
ZDT3	$1.30e-04_{1.2e-05}$	$2.11e-04_{4.8e-05}$	$4.38e-03_{6.8e-03}$
ZDT4	$2.18e-04_{9.1e-05}$	$4.44e-02_{5.8e-02}$	$5.32e-03_{5.4e-03}$

Table 16 GD. Median and Interquartile Range

	NSGAII	SPEA2	MEAT
ZDT1	$2.19e-04_{2.8e-05}$	$3.76e-04_{1.7e-04}$	$4.04e-04_{2.6e-04}$
ZDT2	$1.58e-04_{1.0e-04}$	$2.42e-04_{9.0e-05}$	$4.37e-04_{6.3e-04}$
ZDT3	$1.33e-04_{2.4e-05}$	$1.90e-04_{8.9e-05}$	$5.74e-04_{1.2e-02}$
ZDT4	$2.41e-04_{1.8e-04}$	$2.32e-02_{1.1e-01}$	$3.64e-03_{1.0e-02}$

The benchmarks used for comparison and analysis of the results are from ZDT. These benchmark test problems are processed by the MEAT, NSGA-II and SPEA2. The results

Table 17 HV. Mean and Standard Deviation

	NSGAII	SPEA2	MEAT
ZDT1	$6.60e-01_{2.2e-04}$	$6.56e-01_{4.5e-04}$	$6.25e-01_{2.4e-03}$
ZDT2	$3.27e-01_{6.4e-04}$	$3.24e-01_{7.9e-04}$	$2.97e-01_{1.6e-03}$
ZDT3	$5.15e-01_{1.9e-04}$	$5.11e-01_{2.0e-03}$	$4.80e-01_{2.5e-02}$
ZDT4	$6.59e-01_{1.1e-03}$	$6.58e-01_{4.1e-04}$	$6.00e-01_{2.5e-02}$

Table 18 HV. Median and Interquartile Range

	NSGAII	SPEA2	MEAT
ZDT1	$6.60e-01_{4.1e-04}$	$6.56e-01_{9.0e-04}$	$6.26e-01_{4.5e-03}$
ZDT2	$3.27e-01_{1.2e-03}$	$3.23e-01_{1.4e-03}$	$2.98e-01_{2.9e-03}$
ZDT3	$5.15e-01_{3.7e-04}$	$5.12e-01_{4.1e-03}$	$4.93e-01_{4.4e-02}$
ZDT4	$6.58e-01_{2.0e-03}$	$6.58e-01_{7.6e-04}$	$6.08e-01_{4.8e-02}$

are displayed in Tables 13-18. The HV and GD show us that the NSGA-II is the strongest algorithm with the chosen ZDT problem set, because it performed well in the majority of the metrics. However, research has shown that NSGA-II is a robust algorithm, and the interest lies in how the novel MEAT compares with other algorithms. Regarding the spread, the MEAT did better than both NSGA-II and SPEA2. Since the spread is a judge of spacing, the MEAT obtains better uniformity of solutions. Specifically, the MEAT did better on problems ZDT2 and ZDT3.

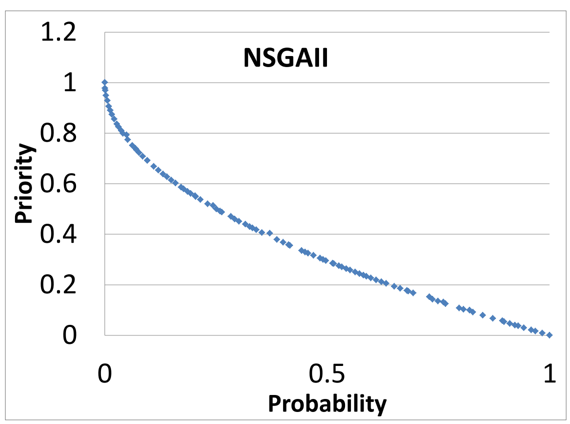

Figure 15 Pareto Front of NSGAII

3.6 Conclusion

MEAT is a novel algorithm that can solve multi-objective problems with comparable numbers, especially with the spread of solutions. MEAT works well for some specific application areas like the SSAP, ZDT2, and ZDT3. More research is needed to figure out

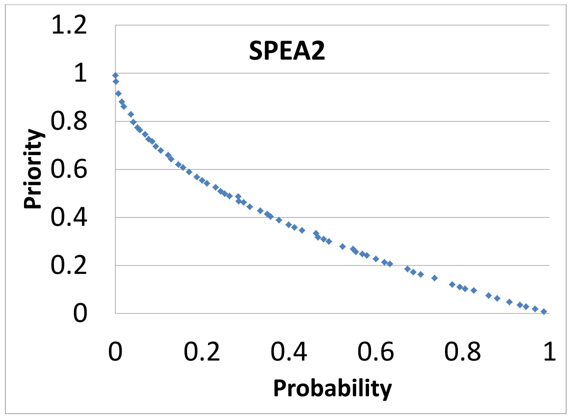

Figure 16 Pareto Front of SPEA2

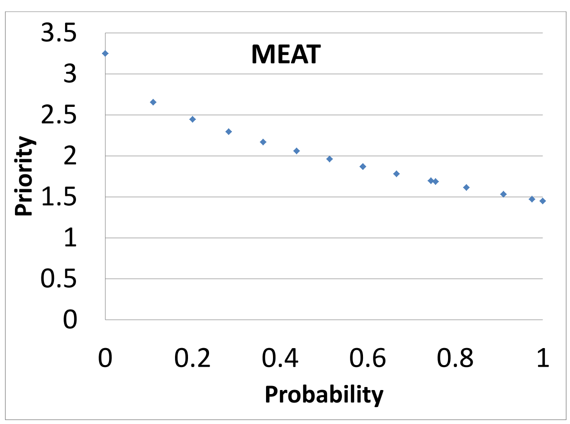

Figure 17 Pareto Front of MEAT

what characteristics of these three problems contributes to the MEAT's high performance. Every application of MEAT is going to be different because of the No Free Lunch Theorem. That said many RAP applications have similar characteristics. MEAT is built on these characteristics. The purpose is to have a system that can be tuned and tweaked as needed to meet the requirements of a new application. In future research it would be beneficial to have a visualization to show what would happen based on the results for the SSAP.

Future work, Advanced Framework for Simulation Integration and Modeling (AF-SIM) is a framework for simulating many different missions and has visions of being a comprehensive simulation domain for many important operations [25].

4. *Tasking for Sensors in Space with Hybrid GA and ES Algorithm*

4.1 Introduction

In Chapters 2 and 3, the research revolved around looking up at satellites for resource allocation. From this point on, the viewpoint considered is flipped to satellites looking down. Especially in a disaster search and recovery scenario, the heterogeneous sensors used to do satellite imaging could help first responders. If a picture is cropped in and does not have the entire context of the situation, then important items are missed. In Figure 18, water, coastline, and a few objects can be viewed, but not much else.

Figure 18 This screenshot of an AFSIM project that shows a cropped view of a few ships in the water.

To see the big picture in Figure 19, much more information can be gleaned. When not seeing the big picture the act of zooming out could change everything. More information or context can make a huge difference. Responsible managing of resources to get more information could change the whole decision-making process and focus resources on where it is headed most. It is desirable to gain to get the context and know the information that makes a difference. The better the resource allocation, the less likely to miss the important information.

This chapter develops a few key parts. First, the application is implemented and tested in the AFSIM framework to provide imagery like seen in Figure 18 and 19 [25].

Figure 19 This screenshot of an AFSIM project shows two satellites and many ships in the water for the search and rescue mission.

Additionally and most importantly, a scalability analysis is performed on the MOEAs to show the results of changing the decision variables. Because real-world problems are typically large scale including many computation variables, this chapter's analysis expands the problems to a larger number of variables. The MEAT, NSGA-II, and SPEA2 are run to figure out how well they perform with these adjustments.

In preview, this chapter discusses satellite imaging and resource allocation. Section 4.2 reviews the literature and other approaches, and formally defines the SSAP and its relation to the more generic RAP. Section 4.3 covers the experimental output and performs a statistical analysis of the data. Finally, Section 4.4 summarizes and offers conclusions.

4.2 Background

The goal of this RAP is to find an allocation for the satellite constellation given specific heterogeneous sensor constraints [86]. The first objective is to have a higher performance by completing the most jobs, balanced with priority of that job. The second objective is to lower the cost of acquiring the necessary imagery. The constraints, such as satellite capacity, location, priority, cost, time, etc. serve as input. Each day comes with a list of points of interest (POI). These POI consist of quality requirements, priority, and

location information. The quality requirements detail the resolution required. The priority is split into four bins evenly distributed.

In the heterogeneous sensor in environment there are satellite sensors and aircraft sensors. According to the Cost Study, the operation and support of the Global Hawk high altitude aircraft is $4,500 per hour [57]. Figure 20 shows an example of one of this aircraft. These sensor platforms can carry a variety of sensors, but typically utilize a multi-spectral camera which can record visual light images and infrared heat signatures.

Figure 20 Global Hawk Research Unmanned Aircraft

The previous approach of the hybrid genetic algorithm and evolutionary strategy will not directly apply to this problem, but an augmentation has potential. The previous approach was good; it had no guarantees, but the proper tuning and adjustments make the algorithm work on this problem.

4.2.1 Algorithmic Complexity. A quick way to find the algorithmic complexity of any algorithm is to evaluate the worst-case scenario. MEAT needs M comparisons in the worst case to determine the ranking of best solutions. This is needed to find the "Best 25%" necessary for the crossover calculation, which will require $O(MN^2)$ computations. Therefore, the time complexity of MEAT is $O(MN^2)$, where M is the number of objectives and N is the population size. In Figure 21, the flowchart describes where the determination of the best solutions is made in the process. It also shows all the other steps as the algorithm works through initialization, evaluation, selection, mutation, crossover, combination, and convergence.

4.2.2 Resource Allocation Problem. Given the cost and completion objectives the algorithm must solve the problem subject to the list of constraints: type, signal, resolution,

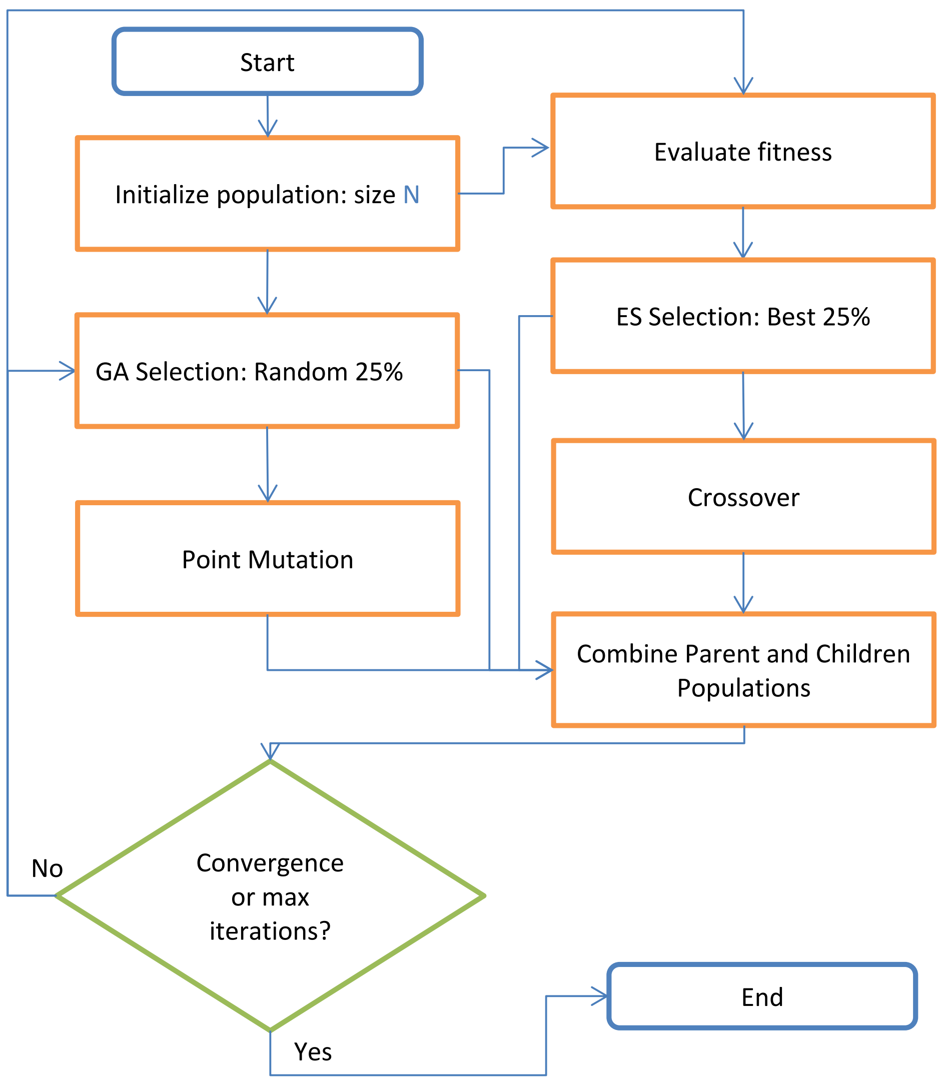

Figure 21 Multi-objective Evolutionary Algorithm Tasker Flowchart: From start to end the flowchart shows the process of the algorithm.

time, capacity, location, etc. Resource allocation or resource management is the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability and the project time. The task is to assign the sensors jobs. The weight of each job completed, or priority, adds an importance factor to each job that could potentially be accomplished. The completion objective would be to sum the jobs completed. The solution should be to complete as many jobs as possible with respect to the constraints. Given real world constraints, the application needs to find the right type of sensor to do the job at the proper location. It also must account for the capacity to store all the data necessary. In Figure 22 the resource allocation problem is diagrammed.

Specifically in this chapter, the discussion is about a specialized RAP called Heterogeneous Aerial Sensor Environment Problem (HASEP). It might be simple to say that the HASEP is like the SSAP versions from the previous chapters, but it is not the same. While it may have some similarities like the purpose of assigning sensors tasks, the HASEP has many differences. The objectives are based on weight of the job and completion of the job, the previous SSAP uses priority and probability. There is not probability that factors into the HASEP objectives. These objectives are based on true or false answer to whether a job was completed or not. The constraints are also different for the new aerial imaging problem. HASEP uses satellite capacity, location, priority, cost, time, type, signal, and resolution, while SSAP uses priority, daily pass probability, capacity, number of daily passes, and track requirements. There are a few similarities there, but for the most part they are different.

4.2.3 Advanced Framework for Simulation Integration and Modeling. The Air Force Research Laboratory's Approach to a high-fidelity simulation environment. The idea behind the Advanced Framework for Simulation Integration and Modeling (AFSIM) is a common modeling framework, using common models in a common environment. AFSIM is ideal for mission-level simulations on the order of a few hours. AFSIM is a framework for simulating many different missions and has visions of being a comprehensive simulation domain for many important operations. Figure 19 shows an instant in the simulation where there are a couple satellites tracking targets. AFSIM will help achieve realistic constraints.

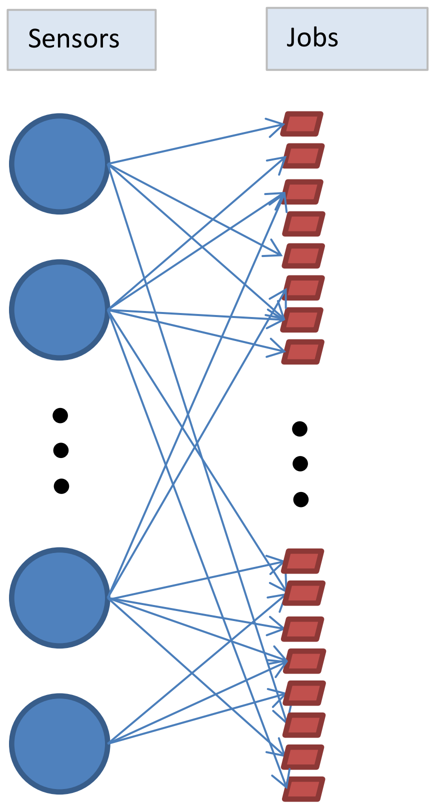

Figure 22 Resource Allocation Problem: Sensors are the resources, and jobs are the activities.

For example, it will help to obtain answers questions such as: Is that sensor able to capture the target at that time? Is the sensor at the proper location to obtain the data? Is the POI in the sensor's field of view? As shown in Figure 23, a given satellite typically has a cone to represent its field of view. The Max Imaging Range is shown as a dashed line with targets that may or may not be in range. AFSIM does a good job of considering targets in range and ignoring targets out of range.

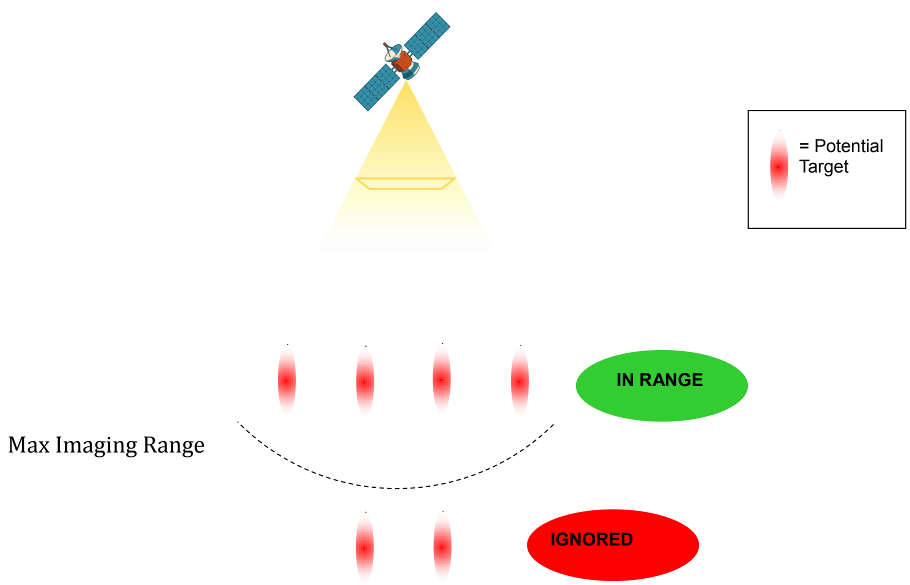

Figure 23 AFSIM gives more accurate and realistic scenario to determine targets in range and ignore targets out of range.

4.3 Results and Analysis

The experiments focus on the dimensionality of the selected MOEAs. Dimensionality is the number of decision variables. This is important because in the real-world engineering problems tend to have thousands of variables. This section assesses the MOEAs which are NSGA-II, SPEA2, and MEAT. The standard or default for these benchmark problems is usually thirty variables, and sometimes as low as ten decision variables. The goal is to study how these MOEAs perform with the quality indicators (EP, SPREAD, and IGD+). In this study, the ZDT problem family is used. During the tests the Pareto front is the

Table 19 Ten Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$1.26e-02_{2.2e-03}$	$1.77e-02_{1.9e-03}$	$8.31e-01_{2.7e-01}$
ZDT2	$1.21e-02_{2.1e-03}$	$1.94e-02_{5.2e-03}$	$2.53e+00_{5.5e-01}$
ZDT3	$7.51e-03_{1.3e-03}$	$1.18e-02_{2.2e-03}$	$6.34e-01_{2.6e-01}$
ZDT4	$2.01e-02_{1.2e-02}$	$1.77e-02_{3.1e-03}$	$5.19e+01_{1.8e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.50e-01_{3.5e-02}$	$3.32e-01_{3.5e-02}$	$7.37e-01_{6.9e-02}$
ZDT2	$3.60e-01_{3.1e-02}$	$3.34e-01_{4.4e-02}$	$1.03e+00_{2.8e-02}$
ZDT3	$7.42e-01_{1.2e-02}$	$7.27e-01_{2.0e-02}$	$7.77e-01_{5.8e-02}$
ZDT4	$3.54e-01_{3.5e-02}$	$7.31e-01_{4.6e-01}$	$1.01e+00_{4.7e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$3.05e-03_{1.3e-04}$	$4.81e-03_{2.5e-04}$	$7.32e-01_{2.2e-01}$
ZDT2	$2.73e-03_{7.8e-05}$	$4.22e-03_{1.6e-04}$	$1.94e+00_{5.1e-01}$
ZDT3	$1.76e-03_{1.3e-04}$	$3.02e-03_{2.5e-04}$	$4.47e-01_{1.5e-01}$
ZDT4	$6.19e-03_{2.1e-03}$	$4.58e-03_{2.5e-04}$	$5.16e+01_{1.8e+01}$

same for each problem while the number of decision variables is changed. Experiments range from ten to one-hundred decision variables.

In Tables 19-28 are the results from the three algorithms tested by the four benchmarks. The EP is for Epsilon indicator [26]. Both epsilon and spread metrics are explained in previous chapters. The new indicator for this chapter is the IGD+. The IGD+ stand for Inverted Generational Distance plus. It can measure quality with specialized Pareto fronts. For these tables the darker the gray shading the better.

The EP, epsilon, value for this set in Table 19 is dominated by the NSGAII solutions on all but the ZDT4. In ZDT4, SPEA2 has 1.77e-02 as shown in the table [45, 65, 30, 124]. NSGAII performs significantly better on benchmarks 1-3 while SPEA2 does better on 4, but it is close. In spread section of the table, it is the opposite. SPEA2 does better in all but the ZDT4 while NSGAII took that one. SPEA2 and NSGAII where within one standard deviation for ZDT1-3, but not for ZDT4. The last four lines of the table are for IGD+ where the first three problems went to the SPEA2 with NSGAII taking the best for only ZDT4. All these four where significant differences.

In Table 20, the EP results show that NSGAII is significantly better in on ZDT1 and ZDT2. SPEA2 is better for ZDT3 and ZDT4 but only significantly with ZDT4. For the spread metric there was not statically significant winner. For ZDT4 specifically, MEAT, NSGAII, and SPEA2 all produced numbers within single standard deviation. In regards

Table 20 Twenty Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$1.25e-02_{1.7e-03}$	$1.88e-02_{3.2e-03}$	$1.15e+00_{4.0e-01}$
ZDT2	$1.30e-02_{3.2e-03}$	$1.96e-02_{4.0e-03}$	$3.06e+00_{4.3e-01}$
ZDT3	$1.31e-02_{3.1e-02}$	$1.27e-02_{2.8e-03}$	$8.39e-01_{3.2e-01}$
ZDT4	$6.70e-01_{2.6e-01}$	$1.82e-02_{3.5e-03}$	$1.75e+02_{4.8e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.49e-01_{2.7e-02}$	$3.26e-01_{3.2e-02}$	$7.95e-01_{5.2e-02}$
ZDT2	$3.51e-01_{2.7e-02}$	$3.24e-01_{3.2e-02}$	$1.04e+00_{2.3e-02}$
ZDT3	$7.45e-01_{1.5e-02}$	$7.30e-01_{1.9e-02}$	$7.82e-01_{7.7e-02}$
ZDT4	$9.44e-01_{9.6e-02}$	$7.10e-01_{4.8e-01}$	$1.00e+00_{5.0e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$3.24e-03_{1.2e-04}$	$5.25e-03_{2.5e-04}$	$1.02e+00_{3.5e-01}$
ZDT2	$2.89e-03_{1.2e-04}$	$4.47e-03_{2.4e-04}$	$2.45e+00_{4.1e-01}$
ZDT3	$2.20e-03_{2.0e-03}$	$3.30e-03_{3.0e-04}$	$6.15e-01_{2.0e-01}$
ZDT4	$4.06e-01_{2.3e-01}$	$4.68e-03_{9.1e-04}$	$1.74e+02_{4.8e+01}$

Table 21 Thirty Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$1.33e-02_{2.4e-03}$	$1.85e-02_{2.8e-03}$	$1.27e+00_{2.2e-01}$
ZDT2	$1.33e-02_{2.0e-03}$	$1.98e-02_{7.1e-03}$	$3.49e+00_{4.1e-01}$
ZDT3	$1.37e-02_{3.1e-02}$	$1.28e-02_{2.1e-03}$	$8.64e-01_{2.6e-01}$
ZDT4	$2.99e+00_{8.2e-01}$	$1.93e-02_{4.1e-03}$	$2.81e+02_{4.2e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.53e-01_{2.7e-02}$	$3.17e-01_{3.2e-02}$	$8.09e-01_{3.2e-02}$
ZDT2	$3.50e-01_{3.3e-02}$	$3.28e-01_{2.6e-02}$	$1.04e+00_{2.5e-02}$
ZDT3	$7.41e-01_{9.6e-03}$	$7.34e-01_{2.0e-02}$	$7.96e-01_{6.9e-02}$
ZDT4	$9.36e-01_{3.5e-02}$	$6.77e-01_{4.7e-01}$	$1.01e+00_{4.4e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$3.80e-03_{1.9e-04}$	$5.41e-03_{2.3e-04}$	$1.12e+00_{2.0e-01}$
ZDT2	$3.44e-03_{1.9e-04}$	$4.61e-03_{2.2e-04}$	$2.87e+00_{3.9e-01}$
ZDT3	$2.44e-03_{2.0e-03}$	$3.45e-03_{2.8e-04}$	$6.25e-01_{1.3e-01}$
ZDT4	$2.66e+00_{8.2e-01}$	$5.15e-03_{2.7e-03}$	$2.81e+02_{4.2e+01}$

to the IGD+, the output shows NSGAII is better for ZDT1-3 but only significantly for the first two. SPEA2 is significantly better for ZDT4.

For Table 21, the EP indicator shows that NSGAII is significantly better in on ZDT1 and ZDT2 while SPEA2 is significantly better for ZDT4. They are statistically the same for ZDT3. The only winner on the spread metric is SPEA2 on ZDT1. NSGAII and SPEA2 are close for ZDT2-3. MEAT, NSGAII, and SPEA2 are in a statistical tie for ZDT4. For the MEAT spread, the value 7.96e-01 at first glance seems like it is close, but it is not close enough to matter at this point. The IGD+ measurement shows NSGAII is better for ZDT1-3 but is withing one standard deviation for ZDT3. In this context, the SPEA2 is statistically better for ZDT4.

Table 22 Forty Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$1.37e-02_{1.7e-03}$	$1.89e-02_{2.6e-03}$	$1.44e+00_{2.3e-01}$
ZDT2	$1.43e-02_{2.6e-03}$	$2.00e-02_{4.7e-03}$	$3.53e+00_{3.2e-01}$
ZDT3	$1.49e-02_{3.1e-02}$	$1.40e-02_{3.0e-03}$	$8.99e-01_{2.0e-01}$
ZDT4	$8.49e+00_{2.1e+00}$	$3.43e-02_{7.8e-02}$	$4.40e+02_{7.1e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.41e-01_{2.9e-02}$	$3.15e-01_{3.8e-02}$	$8.27e-01_{2.9e-02}$
ZDT2	$3.45e-01_{3.2e-02}$	$3.14e-01_{3.2e-02}$	$1.04e+00_{2.3e-02}$
ZDT3	$7.49e-01_{1.6e-02}$	$7.32e-01_{1.4e-02}$	$7.78e-01_{7.3e-02}$
ZDT4	$9.70e-01_{2.1e-02}$	$4.77e-01_{3.0e-01}$	$1.00e+00_{4.0e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$4.84e-03_{2.8e-04}$	$5.72e-03_{3.9e-04}$	$1.27e+00_{2.1e-01}$
ZDT2	$4.66e-03_{3.2e-04}$	$4.72e-03_{2.3e-04}$	$2.91e+00_{3.1e-01}$
ZDT3	$3.06e-03_{2.1e-03}$	$3.78e-03_{4.9e-04}$	$6.56e-01_{8.8e-02}$
ZDT4	$8.16e+00_{2.1e+00}$	$1.67e-02_{6.2e-02}$	$4.40e+02_{7.1e+01}$

Table 23 Fifty Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$1.63e-02_{2.7e-03}$	$1.93e-02_{1.7e-03}$	$1.43e+00_{1.5e-01}$
ZDT2	$2.39e-02_{4.4e-02}$	$2.16e-02_{6.0e-03}$	$3.92e+00_{3.6e-01}$
ZDT3	$2.14e-02_{4.3e-02}$	$1.35e-02_{2.3e-03}$	$9.37e-01_{2.5e-01}$
ZDT4	$1.76e+01_{4.2e+00}$	$5.50e-02_{1.9e-01}$	$5.62e+02_{7.7e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.46e-01_{3.1e-02}$	$3.27e-01_{2.8e-02}$	$8.19e-01_{2.7e-02}$
ZDT2	$3.48e-01_{4.5e-02}$	$3.23e-01_{3.5e-02}$	$1.03e+00_{2.8e-02}$
ZDT3	$7.47e-01_{1.2e-02}$	$7.25e-01_{1.7e-02}$	$7.83e-01_{6.7e-02}$
ZDT4	$9.75e-01_{1.8e-02}$	$8.50e-01_{4.6e-01}$	$1.01e+00_{4.0e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$6.49e-03_{5.5e-04}$	$6.10e-03_{3.5e-04}$	$1.26e+00_{1.3e-01}$
ZDT2	$7.33e-03_{2.8e-03}$	$4.95e-03_{3.0e-04}$	$3.29e+00_{3.4e-01}$
ZDT3	$4.35e-03_{2.7e-03}$	$3.82e-03_{3.2e-04}$	$7.03e-01_{1.3e-01}$
ZDT4	$1.72e+01_{4.2e+00}$	$3.68e-02_{1.7e-01}$	$5.61e+02_{7.7e+01}$

In Table 22, the first four lines show the result for the epsilon indicator. The NSGAII is best in two out of four problems. The SPEA2 is best in the other two, but only significantly for problem ZDT4. With respect to the spread these algorithms are very similar. Most instances there is a statistical tie, but for ZDT4 where SPEA2 is better. Finally, the IGD+ metric shows that NSGAII is better in three out of the four problems, but not significantly in ZDT2. The SPEA2 does significantly better in this instance on ZDT4.

In Table 23, the EP results show that SPEA2 is significantly better in only ZDT3. For the spread metric, most of these results are close. ZDT3 is only problem where SPEA2 is statistically better. With the spread and ZDT4, the MEAT is in a statistical tie with the others. In regard to the IGD+, the output shows SPEA2 dominates all four problems in a big way, and this is the start of a pattern of sorts among the higher decision variables.

Table 24 Sixty Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$1.81e-02_{2.3e-03}$	$1.99e-02_{2.8e-03}$	$1.58e+00_{1.8e-01}$
ZDT2	$2.06e-02_{2.9e-03}$	$1.95e-02_{4.7e-03}$	$3.96e+00_{3.9e-01}$
ZDT3	$2.33e-02_{4.3e-02}$	$1.47e-02_{3.9e-03}$	$9.70e-01_{2.2e-01}$
ZDT4	$3.30e+01_{6.5e+00}$	$1.23e-01_{5.6e-01}$	$7.18e+02_{9.6e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.51e-01_{2.7e-02}$	$3.27e-01_{3.0e-02}$	$8.36e-01_{2.1e-02}$
ZDT2	$3.54e-01_{2.5e-02}$	$3.19e-01_{2.6e-02}$	$1.03e+00_{2.7e-02}$
ZDT3	$7.54e-01_{1.3e-02}$	$7.35e-01_{1.8e-02}$	$8.02e-01_{6.3e-02}$
ZDT4	$9.80e-01_{1.1e-02}$	$6.39e-01_{4.3e-01}$	$1.00e+00_{3.9e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$9.30e-03_{1.1e-03}$	$6.49e-03_{4.1e-04}$	$1.39e+00_{1.6e-01}$
ZDT2	$1.04e-02_{1.4e-03}$	$4.98e-03_{2.5e-04}$	$3.33e+00_{3.7e-01}$
ZDT3	$5.92e-03_{2.7e-03}$	$3.94e-03_{3.5e-04}$	$7.25e-01_{1.1e-01}$
ZDT4	$3.27e+01_{6.5e+00}$	$1.01e-01_{5.1e-01}$	$7.18e+02_{9.6e+01}$

Table 25 Seventy Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$2.29e-02_{2.6e-03}$	$1.92e-02_{2.4e-03}$	$1.49e+00_{2.1e-01}$
ZDT2	$4.72e-02_{8.2e-02}$	$1.94e-02_{4.0e-03}$	$4.13e+00_{4.1e-01}$
ZDT3	$1.52e-02_{2.2e-03}$	$1.40e-02_{2.6e-03}$	$9.53e-01_{2.1e-01}$
ZDT4	$5.05e+01_{7.9e+00}$	$2.75e-01_{1.4e+00}$	$8.57e+02_{8.4e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.54e-01_{2.2e-02}$	$3.29e-01_{2.9e-02}$	$8.29e-01_{2.6e-02}$
ZDT2	$3.67e-01_{5.4e-02}$	$3.14e-01_{3.0e-02}$	$1.03e+00_{2.7e-02}$
ZDT3	$7.55e-01_{1.5e-02}$	$7.29e-01_{1.6e-02}$	$8.02e-01_{5.3e-02}$
ZDT4	$9.83e-01_{9.0e-03}$	$6.31e-01_{4.3e-01}$	$1.01e+00_{3.8e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$1.34e-02_{1.3e-03}$	$6.44e-03_{2.8e-04}$	$1.31e+00_{1.9e-01}$
ZDT2	$1.79e-02_{1.1e-02}$	$5.25e-03_{3.1e-04}$	$3.49e+00_{3.9e-01}$
ZDT3	$7.30e-03_{8.9e-04}$	$3.95e-03_{3.3e-04}$	$7.17e-01_{1.1e-01}$
ZDT4	$5.02e+01_{7.9e+00}$	$2.52e-01_{1.3e+00}$	$8.57e+02_{8.4e+01}$

For Table 24, the EP indicator shows that SPEA2 is significantly better than NSGAII in on ZDT3-4. They are statistically the same for ZDT1-2. The only winner on the spread metric is SPEA2 on ZDT2-3. NSGAII and SPEA2 are close for ZDT1. MEAT, NSGAII, and SPEA2 are in a statistical tie for ZDT4. For the second time in as many tables, the SPEA2 dominates all four problems with the IGD+ measurement.

In Table 25, the first four lines show the result for the epsilon indicator, where the SPEA2 is significantly best in three out of four problems. It ties with SPEA2 on only the ZDT3 here. With respect to the spread these algorithms and these instances show there is a statistical tie expect for ZDT2-3 where SPEA2 is better. The MEAT is in statistically tie with the other algorithms for the ZDT4 problem. Finally, the IGD+ metric shows that NSGAII is better in all four problems when considering seventy decision variables.

Table 26 Eighty Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$2.78e-02_{3.1e-03}$	$2.00e-02_{2.7e-03}$	$1.57e+00_{1.5e-01}$
ZDT2	$1.40e-01_{1.9e-01}$	$1.99e-02_{4.6e-03}$	$4.07e+00_{3.2e-01}$
ZDT3	$3.11e-02_{3.1e-02}$	$1.42e-02_{2.6e-03}$	$1.02e+00_{1.6e-01}$
ZDT4	$7.18e+01_{1.7e+01}$	$4.91e-01_{2.4e+00}$	$9.82e+02_{1.0e+02}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.59e-01_{2.7e-02}$	$3.20e-01_{2.3e-02}$	$8.38e-01_{1.7e-02}$
ZDT2	$4.33e-01_{1.1e-01}$	$3.14e-01_{3.4e-02}$	$1.04e+00_{2.7e-02}$
ZDT3	$7.55e-01_{2.0e-02}$	$7.29e-01_{2.1e-02}$	$8.12e-01_{3.7e-02}$
ZDT4	$9.88e-01_{8.0e-03}$	$7.44e-01_{4.5e-01}$	$1.00e+00_{3.6e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$1.86e-02_{2.5e-03}$	$6.60e-03_{3.1e-04}$	$1.38e+00_{1.4e-01}$
ZDT2	$3.75e-02_{3.3e-02}$	$5.41e-03_{2.9e-04}$	$3.44e+00_{3.0e-01}$
ZDT3	$1.20e-02_{2.9e-03}$	$4.04e-03_{3.5e-04}$	$7.72e-01_{8.4e-02}$
ZDT4	$7.14e+01_{1.7e+01}$	$4.62e-01_{2.3e+00}$	$9.82e+02_{9.9e+01}$

Table 27 Ninety Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$3.19e-02_{3.8e-03}$	$1.99e-02_{2.9e-03}$	$1.61e+00_{1.7e-01}$
ZDT2	$2.30e-01_{2.5e-01}$	$2.10e-02_{5.4e-03}$	$4.17e+00_{3.6e-01}$
ZDT3	$3.57e-02_{5.9e-03}$	$1.36e-02_{2.4e-03}$	$1.01e+00_{1.7e-01}$
ZDT4	$9.79e+01_{1.3e+01}$	$7.93e-01_{3.9e+00}$	$1.14e+03_{9.9e+01}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.61e-01_{2.0e-02}$	$3.22e-01_{3.4e-02}$	$8.40e-01_{2.3e-02}$
ZDT2	$4.88e-01_{1.6e-01}$	$3.15e-01_{2.6e-02}$	$1.03e+00_{2.6e-02}$
ZDT3	$7.55e-01_{1.5e-02}$	$7.39e-01_{1.7e-02}$	$8.17e-01_{5.1e-02}$
ZDT4	$9.89e-01_{6.6e-03}$	$6.99e-01_{4.5e-01}$	$1.00e+00_{3.4e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$2.58e-02_{4.0e-03}$	$6.96e-03_{4.2e-04}$	$1.42e+00_{1.6e-01}$
ZDT2	$6.47e-02_{5.6e-02}$	$5.49e-03_{2.6e-04}$	$3.53e+00_{3.4e-01}$
ZDT3	$1.67e-02_{2.5e-03}$	$4.11e-03_{3.7e-04}$	$7.72e-01_{9.3e-02}$
ZDT4	$9.76e+01_{1.3e+01}$	$7.60e-01_{3.8e+00}$	$1.14e+03_{9.9e+01}$

At this point, SPEA2 is starting to perform even dominate nearly every measurement. In Table 26, the EP results show that SPEA2 is significantly better on all problems. For the spread metric, most of these results are close for ZDT4 among the three algorithms, NSGAII, SPEA2, and MEAT. ZDT1-3 are the problems where SPEA2 is statistically better. In regard to the IGD+, the output shows SPEA2 dominates all four problems yet again.

For Table 27, the EP indicator shows that SPEA2 is significantly better all four problems. The only winner on the spread metric is SPEA2 on ZDT1-2. NSGAII and SPEA2 are close for ZDT3. MEAT, NSGAII, and SPEA2 are in a statistical tie for ZDT4 once again. For the fifth time in as many tables, the SPEA2 dominates all four problems with the IGD+ measurement.

Table 28 One Hundred Decision Variables

	NSGAII	SPEA2	MEAT
EP. Mean and Standard Deviation			
ZDT1	$3.92e-02_{4.8e-03}$	$2.01e-02_{2.3e-03}$	$1.60e+00_{1.2e-01}$
ZDT2	$2.05e-01_{1.9e-01}$	$2.04e-02_{4.0e-03}$	$4.23e+00_{3.0e-01}$
ZDT3	$5.09e-02_{8.1e-03}$	$1.46e-02_{2.8e-03}$	$1.06e+00_{1.8e-01}$
ZDT4	$1.25e+02_{1.7e+01}$	$1.30e+00_{6.3e+00}$	$1.29e+03_{1.1e+02}$
SPREAD. Mean and Standard Deviation			
ZDT1	$3.69e-01_{2.3e-02}$	$3.17e-01_{2.9e-02}$	$8.39e-01_{2.0e-02}$
ZDT2	$4.87e-01_{1.2e-01}$	$3.20e-01_{3.3e-02}$	$1.02e+00_{2.8e-02}$
ZDT3	$7.53e-01_{1.3e-02}$	$7.30e-01_{1.7e-02}$	$8.25e-01_{3.9e-02}$
ZDT4	$9.91e-01_{7.2e-03}$	$6.83e-01_{4.6e-01}$	$1.00e+00_{3.3e-03}$
IGD+. Mean and Standard Deviation			
ZDT1	$3.39e-02_{5.0e-03}$	$7.06e-03_{3.4e-04}$	$1.41e+00_{1.1e-01}$
ZDT2	$6.65e-02_{3.1e-02}$	$5.59e-03_{2.7e-04}$	$3.59e+00_{2.8e-01}$
ZDT3	$2.40e-02_{3.7e-03}$	$4.17e-03_{4.1e-04}$	$8.12e-01_{1.0e-01}$
ZDT4	$1.25e+02_{1.7e+01}$	$1.27e+00_{6.2e+00}$	$1.29e+03_{1.1e+02}$

In Table 28, the SPEA2 did significantly better in all but one of the twelve lines. The only line where is statistically tied is the hard ZDT4 problem on the spread metric. The MEAT is in statistically tie with the other algorithms for the ZDT4 problem.

In summary of the MOEA exploration tables. The SPEA2 shows a pattern of simply being better in nearly all respects especially as the number of decision variables grows. The MEAT was able to remain close to the other algorithm on the hard and pesky problem ZDT4 specifically with the spread. Overall, the three algorithms had various points of performing well.

4.4 Conclusion

In conclusion, The MEAT successfully completed the benchmark tests and matched previous techniques in the literature in terms of the spread of the Pareto optimal front. The MEAT performed worse in terms of the epsilon and inverted generational distance plus metrics. The MOEAs used in this chapter were tested for scalability. The number of decision variables of real-world problems is usually much larger than these benchmark test problems. Increasing is the number of decision variables is one way to get closer to the solving the desired problem. The NSGA-II and SPEA2 are both performed well on the Pareto compliant indicators. SPEA2 specifically performed well as the decision variables grew larger. These characteristics can help a user of the algorithms to choose an algorithm that is the best fit for them.

Much of this chapter discussed the on the HASEP, and unfortunately the research must have an ending point. That is where future work comes in. The MEAT or other algorithms might do well in the new sensor environment which is called HASEP. A researcher could test and augment the MEAT to address resource allocation for the specific disaster recovery scenarios. In previous chapters the MEAT performed space tracking tasks. The MEAT could be used to task space sensors with surveillance of ground objects. While this scheduling domain has some similarity to the ones in the previous chapters, the nature of the assigning and associating resource constraints changes when moving from a paradigm where sensors on the ground point up vs the case where satellite sensors point down at the ground.

5. Conclusion

This dissertation advances the state of the art in approaches to allocate space-based assets in order to effectively utilize sensor resources through the application of multiple meta-heuristic algorithm approaches. The solutions in each of the previous chapters, show the potential to improve current approaches to scheduling techniques. In the experiments run, the introduced novel scheduling algorithms produced better results or similar results in less time. This chapter summarizes the dissertation's main contributions through research questions.

5.1 Satellite Tracking

The SSAP can be generalized to a RAP. From there it can be further generalized to a Scheduling problem. A major contribution from this dissertation is the introduction of an Evolutionary Algorithm Tasker (EAT), which produces better results in most of the measured categories. The EAT solves the SSAP which is mathematically defined in Section 2.3.1.1. For example, it runs in a significantly shorter amount of time when compared with a published scheduling algorithm in the literature. The dissertation addressed key research questions.

> *RQ1*: Will the EAT do well on a full scale SSAP when compared to the published algorithm from the literature in the space object tracking domain? In what ways is the EAT better and in which ways is it worse than the previous approach?

Chapter 2 develops the EAT to solve the SSAP, which is a specific RAP for the SSN to protect valuable space-based assets. This is a promising approach to the SSAP. The chapter seeks to allocate resources that track satellites resulting obtain the biggest yield that the present system can handle. The SSAP is mathematically modeled such that both the control and experiment are on a level playing field. Approach to the SSAP adds priority into the equation, so that the most important satellites are tracked more often. The effectiveness of the approach is shown in the results Section 2.6. To answer the first question, "Will the EAT do well on a full scale SSAP when compared to the published algorithm from the literature in the space object tracking domain?" The simple answer is

yes. The EAT does well on the full scale SSAP when compared to the SP Tasker. Full scale is also an important item to note, because many approaches in the literature only handle a small-scale problem. Chapter 2 schedules tracking for tens of thousands of RSO.

To answer the second part directly, "In what ways is the EAT better and in which ways is it worse than the previous approach?" Briefly the EAT is better in two out of three metrics. For satellite tracking, the two new solution techniques with improved sensor allocation showed improved performance over the pre-existing SP Tasker in the SSN. The first solution is the EAT, which is a single objective evolutionary algorithm. Both the SP Tasker algorithm and the EAT were implemented to solve the SSAP. The EAT can assign more satellites to be tracked and have more satellites tracked by multiple sensors in Section 2.6.1. To be exact, the results show that the EAT is 2.211% better and 18.458% better in Not Tracked Percentage Mean and Unique-Track Percentage Mean, respectively. The actual metric value percentages are obtained by a simple difference between percentages (50.466%-32.008%) and (37.839%-35.628%). In contrast, the SP Tasker has a slightly higher success rate tracking the satellites that receive sensor tracking assignments. Specifically, the SP Tasker is 10.013% (89.943%-79.930%) better with Not-Tracked Percentage Mean. The EAT runs much faster, based on the experiments conducted, which has the potential to allow for a reduced cataloging cycle.

The second solution technique applies two MOEAs to the SSAP. The two MOEAs are the pMOEA/D-DE and MOEA/D-DE. These two approaches did well when tested against the SSAP. Empirical results suggest the highly restrictive constraints of the SSAP can be met by introducing additional variables. In Section 2.6.2, the pMOEA/D-DE is better than the MOEA/D-DE in all the tested quality metrics, but the differences are not large. Examination of the quality indicators shows the difference between the MOEAs. Both single objective and multi-objective solutions use priority to track more important satellites more often. Both solutions produce good sensor allocations for the Space Surveillance Network given the input constraints.

Chapter 3 seeks to build upon the key insights gained from developing the single objective algorithm in Chapter 2 into a multi-objective algorithm. This is done by splitting

the single objective into two competing objectives. The following research questions are explored:

> *RQ2*: How well does the MEAT perform in comparison to well-known MOEAs that have a comparable time complexity? If it does well, what metrics did the MEAT perform well?

This dissertation develops the novel MEAT that features a hybrid genetic algorithm approach. A minor contribution is formally defining the MOSSAP in Section 3.3. The MEAT solves the MOSSAP and provides a Pareto front for the user to choose which solution along the Pareto front is preferred. To answer the research questions directly the first says, "How well does the MEAT perform in comparison to well-known MOEAs that have a comparable time complexity?" The MEAT does a respectably well by doing better in the spread metric while doing worse in the GD and HV measurements. The second question is, "If it does well, what metrics did the MEAT perform well?" The MEAT does well with one metric, that is the spread metric. Tables 13 and 14 show this metric. The other MOEAs used for comparison performed better on the Pareto compliant metrics. A specific characteristic where the MEAT does well, in comparison with competing algorithms, is with the spread metric. This dissertation offers an exploration of MOEAs. Three MOEAs are run against the MOSSAP and benchmark problems. Their results are shown for analysis. Another minor contribution is that the decision maker can pick the algorithm to run. The decision maker can also pick the solution from the approximation front of solutions available.

5.2 Heterogeneous Sensors

Based on the new algorithm developed in Chapter 3, Chapter 4 changes the application domain from primarily ground-based assets to aerial and space-based assets. This full implementation in this new application domain is the goal, and this dissertation takes steps toward the goal. Building on previous research, further experimentation and analysis is made in Chapter 4 with MOEAs using jMetal [34]. Chapter 4 examines the following research questions.

RQ3: Can increasing the decision variables provide good step towards addressing a real-world scaled problem? Which MOEA performs the best overall with the scalability analysis?

Chapter 4 answers RQ3 by performing a scalability analysis with various algorithms. Specifically, the MEAT does provide a computationally effective approach, because it has similar time complexity to the other MOEAs selected. The decision maker can then look at the results presented and make an informed decision on which algorithm and solution to choose. Section 4.3 provides key insights into understanding the capabilities of such systems both in the application domain and the computation domain. Unfortunately, the MEAT does not do well on in most of the experimental cases. It does come in tie for the best statistical result with the other algorithms on the spread metric for nine of ten times for ZDT4. The SPEA2 is the best MOEAs tested in Chapter 4 coming in with the best measurements on 66 out of the 120 lines of testing. Another contribution is the exploration of the MOEAs. The number of decision variables of realistic problems is usually very large, such as hundreds or more. The scalability study of the ZDT problems makes the analysis increasing the number of decision variables helpful. To specifically answer the question that says, "Can increasing the decision variables provide good step towards addressing a real-world scaled problem?" Yes, of course since real-world problems usually have many variables, increasing these small problems will make it closer to a real-world problem. Also, with, "Which MOEA performs the best overall with the scalability analysis?" The short answer is the SPEA2, but NSGA-II is an honorable mention.

5.3 Future Avenues of Research

One avenue for future research is to explore the possibility of using an artificial immune system (AIS) to solve the SSAP. AISs are known for solving multi-objective scheduling problems [49][52]. Although the satellite tracking problem is essentially a resource allocation problem, the resource allocation problem and the scheduling problem have some strong similarities. In satellite tracking, the goal is to track n satellites with m sensors. Likewise in the generic job shop scheduling problem, the purpose is to take n jobs of varying sizes and schedule them on m identical machines with the goal of minimizing the time

it takes to complete the jobs. The commonalities between these problems allows for similar solution approaches. As an example, an AIS for the SSAP has many similarities with EAs, including the respective problem representations. Both algorithms often use whole number vectors to indicate which vector index (sensor) should track which whole number (satellite).

In terms of ad-hoc event handling, the proposed algorithm for disaster rescues dynamic scheduling showed promise, based on the experiments run, for dynamic scenarios that required the ability to incorporate new on-line events into the existing schedule when they arise. The proposed algorithm could incorporate such on-line events and indicates why a new event cannot be added to the existing schedule when failures occur. If the constraints are too restrictive, then further research may need to relax those constraints. In the scenario where a new event is introduced to the schedule, the scheduling algorithm has demonstrated the ability to quickly decide a good course of action based on the experiments conducted. The algorithm can suggest whether to make a simple mutation to the schedule in order to add the new task, to ignore a lower priority old task in order to accommodate the needs of the new task, or to ignore the new task due to its lower overall priority. Whatever decision is made, the aim is to produce a schedule that will perform well according to effective and efficient metrics. The EA solution technique has shown promise in the experiments for creating schedules well-adapted to the domain of satellite tracking, including the ability to make changes to the schedule when on-line and ad-hoc events dictate a need to do so.

The goal for future research would be to create a scheduling algorithm that can take complex priorities and constraints in an almost real-time manner and produce effective schedules according to emerging requirements. Such a system would drastically improve the current long lead times in crafting and implementing schedules for resource allocation systems. A research group at MITRE has just recently made the first step along these lines [16].

Bibliography

1. Abbasi, Viqar, et al. "The NEOSSat Experience: 5 Years in the Life of Canada's Space Surveillance Telescope." *1st NEO and Debris Detection Conference22*. 2019.
2. "Adobe Photoshop CS 4.0," 2008.
3. Ahmadi, Ghazaleh, et al. "A Decision Support Model for Robust Allocation and Routing of Search and Rescue Resources after Earthquake: A Case Study," *Operational Research*, 1–43 (2020).
4. Albadr, Musatafa Abbas, et al. "Genetic Algorithm Based on Natural Selection Theory for Optimization Problems," *Symmetry*, *12*(11):1758 (2020).
5. Analytical Graphics, Inc., "Systems Tool Kit (STK)," 2014. STK 11.
6. Baggetta, Jeff S and Brian E Skahillb. "Hybrid Optimization using an Evolutionary Strategy and Surrogate Assisted Local Search," *Improving Environmental Model Calibration and Prediction* (2010).
7. Baker, Kenneth R and Dan Trietsch. *Principles of Sequencing and Scheduling*. John Wiley & Sons, 2009.
8. Barker, Bill, "Astrodynamics Concepts and Terminology," October 2012. Omitron Inc.
9. Barker, Bill and Richard Leap, "ASW Training," 2012. Omitron Inc.
10. Blodger, Ian. "Reclassifying Geostationary Earth Orbit as Private Property: Why Natural Law and Utilitarian Theories of Property Demand Privatization," *Minn. JL Sci. & Tech.*, *17*:409 (2016).
11. Bollinger, James, "Ballistic Missile Early Warning System (BMEWS)." `http://www.radomes.org/museum/BMEWS.php`, 1996. Accessed on: 18 August 2016.
12. Booker, Cory and Dan Sullivan, "Booker, Sullivan Press FCC on Space Debris," Oct 2017.
13. Brandenburg, B, "Schedcat: the Schedulability Test Collection and Toolkit," 2013.
14. Brodkin, Jon, "SpaceX and OneWeb Broadband Satellites Raise Fears about Space Debris," Oct 2017.
15. Bull, Lawrence and Terence C. Fogarty. *An Evolution Strategy and Genetic Algorithm Hybrid: An Initial Implementation and First Results*, 95–102. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994.
16. Cardin, R, et al. "SNARE (Sensor Network Autonomous Resilient Extensible): Decentralized Sensor Tasking Improves SDA Tactical Relevance." *Advanced Maui Optical and Space Surveillance Technologies Conference*. 2021.
17. "Celestrak SATCAT Boxscore." Celestrak: `https://celestrak.com/satcat/boxscore.php`, April 2022.

18. Center, Joint Space Operations. "Space-Track. org," *The Source for Space Surveillance Data,[www. space-track. org], USSTRATCOM* (2018).

19. Cerqueira, F., et al. "Linux's Processor Affinity API, Refined: Shifting Real-Time Tasks Towards Higher Schedulability." *Real-Time Systems Symposium (RTSS), 2014 IEEE*. 249–259. Dec 2014.

20. Chandrashekar, S. "Space, War, and Deterrence: A Strategy for India," *Astropolitics*, *14*(2-3):135–157 (2016).

21. Chapman, Stephen N. *The Fundamentals of Production Planning and Control*. Prentice Hall, 2006.

22. Chatters, Edward P. and Brian J. Crothers. *AU-18 Space Primer*. 2009.

23. Cherry, Paul R. "Sensor Tasking by the Space Defense Operations Center." *Proceedings of the Space Surveillance Workshop*. 93–98. 1993.

24. Choi, Charles Q. "Earth-based Lasers Could Zap Space Junk Clear From Satellites," (Mar 2011).

25. Clive, Peter D, et al. "Advanced Framework for Simulation, Integration and Modeling (AFSIM)(Case Number: 88ABW-2015-2258)." *Proceedings of the International Conference on Scientific Computing (CSC)*. 73. 2015.

26. Coello Coello, Carlos A., et al. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.

27. Compton, Matthew, et al. "1— The Network Tasking Order (NTO)." *Military Communications Conference, 2008. MILCOM 2008. IEEE*. 1–7. 2008.

28. "The Crankshaft Publishing: Strategic Warning Radars." `http://www.the-crankshaft.info/2010/04/strategic-warning-radars.html`, April 2010. Accessed on: 15 July 2010.

29. Crosier, Maj Gen Clinton and Plans Director. "United States Strategic Command Space Situational Awareness Sharing Program Update," *United Nations Committee on the Peaceful Uses of Outer Space Scientific and Technical Subcommittee 54th Session Vienna, Austria, February*, *3* (2016).

30. Deb, K., et al. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, *6*(2):182 –197 (apr 2002).

31. Deb, Kalyanmoy and Himanshu Jain. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints.," *IEEE Trans. Evolutionary Computation*, *18*(4):577–601 (2014).

32. Denardo, Eric V. *Dynamic Programming : Models and Applications*. Mineola, N.Y.: Dover Publications, 2003. Originally published: Englewood Cliffs, N.J. : Prentice-Hall, c1982.

33. Du, Mengyu and Yan-Fu Li. "An Investigation of New Local Search Strategies in Memetic Algorithm for Redundancy Allocation in Multi-State Series-Parallel Systems," *Reliability Engineering & System Safety*, *195*:106703 (2020).

34. Durillo, Juan J. and Antonio J. Nebro. "jMetal: A Java Framework for Multi-Objective Optimization," *Advances in Engineering Software*, *42*:760–771 (2011).

35. Durillo, Juan J. and Antonio J. Nebro. "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, *42*:760–771 (2011).

36. Durillo, Juan J, et al. "On The Effect of the Steady-State Selection Scheme in Multi-Objective Genetic Algorithms." *International Conference on Evolutionary Multi-Criterion Optimization*. 183–197. 2009.

37. Elliott, G.A. and J.H. Anderson. "Exploring the Multitude of Real-Time Multi-GPU Configurations." *Real-Time Systems Symposium (RTSS), 2014 IEEE*. 260–271. Dec 2014.

38. "The European Space Agency." `https://www.esa.int/`, February 2020. Accessed on: 16 October 2020.

39. "Geosynchronous Space Situational Awareness Program (GSSAP)." Air Force Space Command: `https://www.spaceforce.mil/About-Us/Fact-Sheets/Article/2197772/geosynchronous-space-situational-awareness-program/`, Oct 2020.

40. Giuliano, Concetto R, et al. "High Performance Computing Software Applications for Space Situational Awareness." *The Advanced Maui Optical and Space Surveillance Technologies Conference1*. 89. 2006.

41. Golberg, David E. "Genetic Algorithms in Search, Optimization, and Machine Learning," *Addison Wesley*, *1989*:102 (1989).

42. Gonçalves, José Fernando, et al. "A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem," *European journal of operational research*, *167*(1):77–95 (2005).

43. Greve, Gabriel H, et al. "Evolutionary Sensor Allocation for the Space Surveillance Network," *The Journal of Defense Modeling and Simulation* (2017).

44. Hadka, Dave and others, "MOEA Framework," 2016.

45. Hamdy, Mohamed, et al. "Implementation of Pareto-Archive NSGA-II Algorithms to a Nearly-Zero-Energy Building Optimisation Problem." *Proceedings of the building simulation and optimization conference*. 2012.

46. Hansen, Michael Pilegaard, et al., "Evaluating the Quality of Approximations to the Non-Dominated Set," 1998.

47. Harrington, J.D. and Nilufar Ramji, "NASA Orbital Debris Program Office Website." `https://orbitaldebris.jsc.nasa.gov/photo-gallery/`, April 2022. Accessed on: 21 April 2022.

48. Harrington, J.D., et al., "NASA Orbital Debris Program Office Website." `https://orbitaldebris.jsc.nasa.gov/photo-gallery/`, April 2022. Accessed on: 5 May 2022.

49. Hart, Emma and Peter Ross. "An Immune System Approach to Scheduling in Changing Environments." *Genetic and Evolutionary Computation Conference*. 1559–1566. 1999.

50. He, Dong-Zhi, et al. "Hybrid Earliest Deadline First/preemption Threshold Scheduling for Real-time Systems." *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on1*. 433–438 vol.1. Aug 2004.

51. Hill, K., et al. "Dynamic Tasking of Networked Sensors Using Covariance Information." *Advanced Maui Optical and Space Surveillance Technologies Conference*. September 2010.

52. Honório, Leonardo M., et al. "Dynamic Polymorphic Agents Scheduling and Execution Using Artificial Immune Systems." *Proceedings of the 7th international conference on Artificial Immune Systems*. ICARIS '08. 166–175. Berlin, Heidelberg: Springer-Verlag, 2008.

53. Hoots, Felix R and Ronald L Roehrich. *Models for Propagation of NORAD Element Sets*. Technical Report, DTIC Document, 1980.

54. Hoots, Felix R, et al. "History of Analytical Orbit Modeling in the US Space Surveillance System," *Journal of Guidance, Control, and Dynamics*, *27*(2):174–185 (2004).

55. Ibaraki, Toshihide and Naoki Katoh. *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, 1988.

56. Jaszkiewicz, Andrzej. "On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem-A Comparative Experiment," *IEEE Transactions on Evolutionary Computation*, *6*(4):402–412 (2002).

57. Kenul, Philip Matthew, et al. "Sensing Hazards with Operational Unmanned Technology: Cost Study of Global Hawk Unmanned Aircraft System Operations for High Impact Weather Observations, Final Report," (2018).

58. Kessler, Donald J, et al. "The Kessler Syndrome: Implications to Future Space Operations," *Advances in the Astronautical Sciences*, *137*(8):2010 (2010).

59. Kiekintveld, Christopher, et al. "Computing Optimal Randomized Resource Allocations for Massive Security Games." *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 689–696. 2009.

60. Kleeman, Mark P and Gary B Lamont. "Solving the Aircraft Engine Maintenance Scheduling Problem Using a Multi-objective Evolutionary Algorithm." *Evolutionary Multi-Criterion Optimization*. 782–796. 2005.

61. Knowles, Joshua D and David W Corne. "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evolutionary computation*, *8*(2):149–172 (2000).

62. Laumanns, Marco, et al. "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization," *Evol. Comput.*, *10*(3):263–282 (September 2002).

63. Lee, Kwang Y and Frank F Yang. "Optimal Reactive Power Planning Using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Strategy, Genetic Algorithm, and Linear Programming," *IEEE Transactions on Power Systems*, *13*(1):101–108 (1998).

64. "LeoLab, Inc. Website." `https://www.leolabs.space/`, October 2020. Accessed on: 16 October 2020.

65. Li, Hui and Qingfu Zhang. "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II," *Evolutionary Computation, IEEE Transactions on*, *13*(2):284–302 (2009). ID: 1.

66. Li, Miqing, et al. "Shift-Based Density Estimation for Pareto-Based Algorithms in Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, *18*(3):348–365 (2014).

67. Liu, Bo, et al. "An Enhanced MOEA/D-DE and Its Application to Multiobjective Analog Cell Sizing." *Evolutionary Computation (CEC), 2010 IEEE Congress on*. 1–7. July.

68. Luce, Rick, et al. *Joint Space Operations Center Mission System Application Development Environment*. Technical Report, DTIC Document, 2012.

69. Lukasiewycz, M., et al. "SAT-decoding in Evolutionary Algorithms for Discrete Constrained Optimization Problems." *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. 935–942. 2007.

70. Maheswaran, Muthucumaru, et al. "Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems." *Heterogeneous Computing Workshop, 1999. (HCW '99) Proceedings. Eighth*. 30–44. 1999.

71. Malik, Tariq, "Space Junk Forces ISS Astronauts to Take Shelter in 'Lifeboat' Capsules." `https://www.huffpost.com/entry/space-junk-iss-astronauts_n_1376963`, March 2012. Accessed on: 30 May 2022.

72. Malve, Sujay and Reha Uzsoy. "A Genetic Algorithm for Minimizing Maximum Lateness on Parallel Identical Batch Processing Machines with Dynamic Job Arrivals and Incompatible Job Families," *Computers & Operations Research*, *34*(10):3016 – 3028 (2007).

73. Martin, David, "The Battle Above." `https://www.cbsnews.com/`, April 2015. Accessed on: 16 October 2020.

74. Maskell, Paul and Lorne Oram. "Sapphire: Canada's Answer to Space-Based Surveillance of Orbital Objects." *Advanced Maui Optical and Space Surveillance Conference*. 2008.

75. Miller, James G. "A New Sensor Allocation Algorithm for the Space Surveillance Network," *Military Operations Research - Alexandria*, *12*(1):57–69 (2007).

76. Moomey, Daniel. *Aiding Geostationary Space Situational Awareness Using Small Aperture Commercial Telescopes*. PhD dissertation, MS thesis, Department of Aero-

nautics and Astronautics, AFIT, Wright-Patterson AFB, OH, 2015), 3. 10. Headquarters Air Force Space Command/A3CD, Enabling Concept, 12, 22–24, 2015.

77. Moomey, Daniel. “A Call to Action,” *Air & Space Power Journal*, *29*(6) (2015).

78. Morreale, Brittany, et al. “Australian Space Situational Awareness Capability Demonstrations.” *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2017.

79. Mukhopadhyay, Anirban, et al. “Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II,” *IEEE Transactions on Evolutionary Computation*, *18*(1):20–35 (2013).

80. Mukhopadhyay, Anirban, et al. “A Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I,” *IEEE Transactions on Evolutionary Computation*, *18*(1):4–19 (2013).

81. “NASA: What Is Orbital Debris?.” `https://www.nasa.gov/`, June 2010. Accessed on: 30 May 2022.

82. Nebro, Antonio J. and Juan J. Durillo. “A Study of the Parallelization of the Multi-Objective Metaheuristic MOEA/D.” *Proceedings of the 4th international conference on Learning and intelligent optimization*. LION’10. 303–317. Berlin, Heidelberg: Springer-Verlag, 2010.

83. Nebro, Antonio J. and Juan J. Durillo, “jMetal 4.3 User Manual,” 2013.

84. Newman, Andrew J, et al. “A Comparison of Stochastic Optimizers Applied to Dynamic Sensor Scheduling for Satellite Tosracking.” *SPIE Defense, Security, and Sensing*. 76970M–76970M. 2010.

85. Osman, M. S., et al. “An Effective Genetic Algorithm Approach to Multiobjective Resource Allocation Problems (MORAPs),” *Applied Mathematics and Computation*, *163*(2):755 – 768 (2005).

86. Osman, M. S., et al. “An Effective Genetic Algorithm Approach to Multiobjective Resource Allocation Problems (MORAPs),” *Applied Mathematics and Computation*, *163*(2):755 – 768 (2005).

87. Petrick, Beth L. *Weighting Scheme For the Space Surveillance Network Automated Tasker*. MS thesis, Air Force Institute of Technology, December 1994. ADA289285.

88. Phipps, Claude R. “A Laser-Optical System to Re-Enter or Lower Low Earth Orbit Space Debris,” *Acta Astronautica*, *93*:418–429 (2014).

89. Pinedo, Michael. *Planning and Scheduling in Manufacturing and Services*, *24*. Springer, 2005.

90. Pinedo, Michael L. *Scheduling: Theory, Algorithms, and Systems*. Springer Science & Business Media, 2012.

91. “Premier Space Surveillance Squadron Located at Eglin.” `http://www.eglin.af.mil/News/Article-Display/Article/391473/`

`premier-space-surveillance-squadron-located-at-eglin`, December 2011. Accessed on: 30 May 2022.

92. Rangsaritratsamee, Ruedee, et al. "Dynamic Rescheduling that Simultaneously Considers Efficiency and Stability," *Computers & Industrial Engineering*, *46*(1):1 – 15 (2004).

93. Reynolds, M. Brent, et al. "Service Oriented Infrastructure Resource Provisioning Optimization." AFIT Technical Report.

94. Rovetto, Robert J and TS Kelso. "Preliminaries of a Space Situational Awareness Ontology.(2016) 26th AIAA." *AAS Space Flight Mechanics meeting, Napa, California*. 2016.

95. Schumacher, PW and Felix R Hoots. "Evolution of the NAVSPACECOM Catalog Processing System Using Special Perturbations." *Proceedigns of the Fourth US/Russian Space Surveillance Workshop, US Naval Observatory, Washington, DC*. 2000.

96. Schwefel, Hans-Paul, "Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series," 1995.

97. Shapiro, S. S. and M. B. Wilk. "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, *52*(3/4):pp. 591–611 (1965).

98. Shen, Shuangyan, et al. "Cleaning Space Debris with a Space-Based Laser System," *Chinese Journal of Aeronautics*, *27*(4):805–811 (2014).

99. Sierra, Margarita Reyes and Carlos A Coello Coello. "Improving PSO-Based Multi-Objective Optimization Using Crowding, Mutation and Epsilon-Dominance." *International Conference on Evolutionary Multi-Criterion Optimization*. 505–519. 2005.

100. Singh, Lovejit and Sarbjeet Singh. "A Survey of Workflow Scheduling Algorithms and Research Issues," *International Journal of Computer Applications*, *74*(15):21–28 (July 2013). Full text available.

101. Soon, Gan Kim, et al. "A Comparison on the Performance of Crossover Techniques in Video Game." *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on*. 493–498. 2013.

102. "Space Control Capstone Requirements Document," 2000.

103. Talbi, El-Ghazali. *Metaheuristics: from Design to Implementation*, *74*. John Wiley & Sons, 2009.

104. Tan, K.C., et al. "An Evolutionary Artificial Immune System For Multi-Objective Optimization," *European Journal of Operational Research*, *187*(2):371 – 392 (2008).

105. Tran, Triet. "Application of a COTS Resource Optimization Framework to the SSN Sensor Tasking Domain–Part I: Problem Definition." *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, held in Wailea, Maui, Hawaii, September 15-18, 2014, Ed.: S. Ryan, The Maui Economic Development Board, id. 511*. 51. 2015.

106. Utzmann, Jens, et al. “SBSS Demonstrator: A Design for Efficient Demonstration of Space-based Space Surveillance End-to-end Capabilities.” *40th COSPAR Scientific Assembly. Held 2-10 August 2014, in Moscow, Russia, Abstract PEDAS. 1-27-14.40*. 3449. 2014.

107. Utzmann, Jens, et al. “A System Design for Space-based Space Surveillance.” *Proceedings of Small Satellites Systems & Services Symposium, Porto Petro*. 26–30. 2014.

108. Vallado, David A and Jacob D Griesbach. “Simulating Space Surveillance Networks.” *Paper AAS 11-580 presented at the AAS/AIAA Astrodynamics Specialist Conference. July*. 2011.

109. Wang, Rui, et al. “Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods,” *IEEE Transactions on Evolutionary Computation*, *20*(6):821–837 (2016).

110. Wang, Yun and M. Saksena. “Scheduling Fixed-priority Tasks with Preemption Threshold.” *Real-Time Computing Systems and Applications, 1999. RTCSA ’99. Sixth International Conference on*. 328–335. 1999.

111. Wang, Zai, et al. “Multi-objective approaches to optimal testing resource allocation in modular software systems,” *IEEE Transactions on Reliability*, *59*(3):563–575 (2010).

112. Weeden, Brian, “The Numbers Game.” The Space Review: `http://www.thespacereview.com/article/1417/1`, July 2009.

113. Weeden, Brian, “Going Blind: Why America Is on the Verge of Losing Its Situational Awareness in Space and What Can Be Done About It,” 2012.

114. Wieder, A. and B.B. Brandenburg. “On Spin Locks in AUTOSAR: Blocking Analysis of FIFO, Unordered, and Priority-Ordered Spin Locks.” *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*. 45–56. Dec 2013.

115. Wilkinson, Matthew, et al. “The Next Generation of Satellite Laser Ranging Systems,” *Journal of Geodesy*, *93*(11):2227–2247 (2019).

116. Wolpert, D.H. and W.G. Macready. “No Free Lunch Theorems for Optimization,” *Evolutionary Computation, IEEE Transactions on*, *1*(1):67–82 (Apr 1997).

117. Xu, Ying, et al. “Hybrid Multi-Objective Evolutionary Algorithms Based On Decomposition for Wireless Sensor Network Coverage Optimization,” *Applied Soft Computing*, *68*:268–282 (2018).

118. Yang, Erfu, et al. “Multi-Objective Evolutionary Optimizations of a Space-Based Reconfigurable Sensor Network under Hard Constraints,” *Soft Computing*, *15*(1):25–36 (2011).

119. Yen, Gary G and Zhenan He. “Performance Metric Ensemble for Multiobjective Evolutionary Algorithms,” *IEEE Transactions on Evolutionary Computation*, *18*(1):131–144 (2014).

120. Zacherl, Brian. *Weapon-Target Pairing; Revising an Air Tasking Order in Real-Time*. Technical Report, DTIC Document, 2006.

121. Zăvoianu, Alexandru-Ciprian, et al. "DECMO2: A Robust Hybrid and Adaptive Multi-Objective Evolutionary Algorithm," *soft Computing*, *19*(12):3551–3569 (2015).

122. Zhang, Qingfu and Hui Li. "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *Evolutionary Computation, IEEE Transactions on*, *11*(6):712 –731 (dec. 2007).

123. Zitzler, Eckart, et al. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary computation*, *8*(2):173–195 (2000).

124. Zitzler, Eckart, et al., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," 2001.

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
16-06-2022	Doctoral Dissertation	Jan 2011 - June 2022

4. TITLE AND SUBTITLE
Scheduling for Space Tracking and Heterogeneous Sensor Environments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Greve, Gabriel, H. CTR

5d. PROJECT NUMBER
19G130

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT-ENG-DS-22-J-085

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Space Vehicles Directorate, Air Force Research Laboratory, Kirtland AFB
Brian McBee Chief at Simulation & Technology Assessment Branch
3350 Aberdeen Ave SE
Kirtland AFB, NM 87117
brian.mcbee.1@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A. Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This dissertation draws on the fields of heuristic and meta-heuristic algorithm development, resource allocation problems, and scheduling to address key Air Force problems. The world runs on many schedules. People depend upon them and expect these schedules to be accurate. A process is needed where schedules can be dynamically adjusted to allow tasks to be completed efficiently. For example, the Space Surveillance Network relies on a schedule to track objects in space. The schedule must use sensor resources to track as many high-priority satellites as possible to obtain orbit paths and to warn of collision paths. Any collisions that occurred between satellites and other orbiting material could be catastrophic. To address this critical problem domain, this dissertation introduces both a single objective evolutionary tasker algorithm and a multi-objective evolutionary algorithm approach. The aim of both methods is to produce space object tracking schedules to ensure that higher priority objects are appropriately assessed for potential problems. Simulations show that these evolutionary algorithm techniques effectively create schedules to assure that higher priority space objects are tracked. These algorithms have application to a range of dynamic

15. SUBJECT TERMS
Genetic Algorithms, Space Domain Awareness, Evolutionary Computation, Scalability Analysis, Resource Allocation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Kenneth M. Hopkinson, AFIT/ENG
U	U	U	UU	102	**19b. TELEPHONE NUMBER** *(Include area code)* (937)255-6565x4579 Kenneth.Hopkinson@afit.edu

Reset