

**OPTIMAL ALLOCATION OF  
SATELLITE NETWORK RESOURCES**

Sarah E. Burrowbridge



Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
In partial fulfillment of the requirements of the degree of

Master of Science  
In  
Mathematics

Jeff Borggaard  
John A. Burns  
Hamilton Hagar  
Terry L. Herdman

17 December 1999  
Northern Virginia Campus

Keywords: Greedy Algorithm, Scheduler, Satellites,  
Mission Planning, Resource Allocation

Copyright 1999, Sarah E. Burrowbridge

# Optimal Allocation of Satellite Network Resources

**Sarah Burrowbridge**

## **(ABSTRACT)**

This work examines a straightforward solution to a problem of satellite network resource allocation by exploring the application of an optimal algorithm to a subset of the general scheduling problem. Making some basic simplifications, such as the limitation of the mission scenarios to Low Earth Orbiting satellites, allows the effective application of the rigorous methods of the Greedy Activity-Selector algorithm. Tools such as Analytical Graphic's Satellite Tool Kit and MATLAB 5.0 are integrated to assist in the implementation of the algorithm. Several examples are presented in order to illustrate the effectiveness of the process.

## **Acknowledgements**

I would like to acknowledge the hard work and valuable assistance that many people have contributed to the completion of this work. I would like to thank my advisor Dr. Terry Herdman for the constant guidance and support he has provided me throughout my academic career. His understanding of both academia and industry not only initially sparked my interest in the field of applied mathematics, but has continually provided great insight and motivation for me as well.

Special thanks, also, to Hamilton Hagar who encouraged and supported my interest in space applications. He is a fabulous mentor and I have appreciated the chance to learn from his knowledge and expertise. Our valuable discussions and his comments and insight have provided the basis of this investigation.

Finally I would like to thank my parents Jenny and Bill Burrowbridge for their unconditional love and support in all that I do.

# Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	BACKGROUND .....	1
1.2	PREVIOUS EFFORTS.....	1
1.3	SCOPE AND APPROACH .....	2
<b>2</b>	<b>MODELS AND METHODOLOGY.....</b>	<b>3</b>
2.1	SATELLITE ORBITS.....	3
2.2	NATURE OF THE SCHEDULING PROBLEM .....	3
2.3	PROCEDURE.....	4
2.4	GREEDY ACTIVITY-SELECTOR.....	4
2.5	PROVING THE GREEDY ALGORITHM CORRECT .....	7
2.5.1	Theorem.....	7
2.6	RELATIONSHIP OF THE GREEDY ACTIVITY-SELECTOR TO THE GENERAL SATELLITE SCHEDULING PROBLEM.....	8
2.6.1	Simple Example.....	9
2.7	IMPLEMENTATION OF THE GREEDY ALGORITHM .....	11
2.7.1	Opportunity/Demand Count Ratio .....	11
2.7.2	Scheduler Performance Ratio.....	11
2.7.3	Schedule/Demand Count Ratio .....	11
2.7.4	Site Loading / Availability Count Ratio.....	12
<b>3</b>	<b>EXAMPLES AND SCENARIOS .....</b>	<b>13</b>
3.1	BASIC CASE 4 SATELLITES, 1 TRACKING SITE.....	13
3.1.1	Let the contact requirements be 2 contacts per day for each satellite. ....	14
3.1.2	Let the contact requirements be 4 contacts per day for each satellite. ....	15
3.1.3	Let the contact requirements be 5 contacts per day for each satellite. ....	15
3.2	SATELLITE FORMATIONS .....	15
3.2.1	Satellites in Formation .....	17
<b>4</b>	<b>CONCLUSIONS .....</b>	<b>19</b>
<b>5</b>	<b>MATLAB AND STK DATA AND GRAPHS.....</b>	<b>20</b>
5.1	CHARTS AND GRAPHS FOR CASE 1 – 4 SATELLITES, 2 TRACKING STATIONS .....	20
5.1.1	Satellites (EOS Chem-1, QuickSCAT, Landsat-7, ICESat).....	21
5.1.2	Accesses of 4 Satellites with respect to Hawaii and Norway Tracking Sites.....	22
5.1.3	Satellite visibilities calculated in STK.....	23
5.1.4	Satellite Visibilities with Respect to Hawaii Tracking Site .....	24
5.2	SCHEDULER RESULTS FOR CASE 1 – 4 SATELLITES, 1 TRACKING STATION; .....	25
5.2.1	Successfully Scheduled Contacts.....	25
5.2.2	Total Demand, Opportunity and Loading Count per Satellite.....	26
5.2.3	Total Demand, Opportunity and Loading Time per satellite.....	27
5.2.4	Site Utilization Count by Satellite.....	28
5.2.5	Site Utilization Time by Satellite.....	29
5.2.6	Total Site Availability Count and Total Site Loading Count.....	30
5.2.7	Total Site Availability Time and Total Site Loading Time.....	31
5.3	SCHEDULER RESULTS FOR CASE 1 – 4 SATELLITES, 1 TRACKING STATION; .....	32
5.3.1	Successfully Scheduled Contacts.....	32
5.3.2	Total Demand, Opportunity and Loading Count per Satellite.....	33
5.3.3	Total Demand, Opportunity and Loading Time per Satellite.....	34
5.3.4	Site Utilization Count by Satellite.....	35

5.3.5	Site Utilization Time by Satellite.....	36
5.3.6	Total Site Availability Count and Total Site Loading Count.....	37
5.3.7	Total Site Availability Time and Total Site Loading Time.....	38
5.4	SCHEDULER RESULTS FOR CASE 1 – 4 SATELLITES, 1 TRACKING STATION; .....	39
5.4.1	Successfully Scheduled Contacts.....	39
5.4.2	Total Demand, Opportunity and Loading Count per Satellite.....	40
5.4.3	Total Demand, Visibility and Loading Time per Satellite .....	41
5.4.4	Site Utilization Count by Satellite.....	42
5.4.5	Site Utilization Time by Satellite.....	43
5.4.6	Total Site Availability Count and Total Site Loading Count.....	44
5.4.7	Total Site Availability Time and Total Site Loading Time.....	45
5.5	COMPARE NETWORK SUPPORT PERFORMANCE RATIOS FOR CASE 1 .....	46
5.6	7 SATELLITES, 1 TRACKING STATION.....	47
5.6.1	Satellite Visibilities of Seven Satellites with Respect to Hawaii Tracking Site.....	47
5.6.2	Total Demand, Visibility and Loading Count per Satellite.....	48
5.7	CHARTS AND GRAPHS FOR CASE 2 – 3 SATELLITE FORMATION .....	49
5.7.1	Satellites (EOS AM-1, EO-1, Landsat-7).....	49
5.7.2	Accesses of 3 Satellites with respect to Hawaii Tracking Site.....	50
5.7.3	Satellite visibilities calculated in STK.....	51
5.7.4	Satellite Visibilities with Respect to Hawaii Tracking Site .....	52
5.8	SCHEDULER RESULTS FOR CASE 2 - 3 SATELLITES IN FORMATION; .....	53
5.8.1	Successfully Scheduled Contacts.....	53
5.8.2	Total Demand, Visibility and Loading Count per Satellite.....	54
<b>6</b>	<b>MATLAB PROGRAMS.....</b>	<b>55</b>
6.1	SCHEDULER – GREEDY ALGORITHM.....	55
6.2	SCHEDULER – QUERY USER FOR SCENARIO .....	65
6.3	SCHEDULER – QUERY USER TO IMPORT DATA FILE .....	66
6.4	SCHEDULER - QUERY USER TO INPUT CONTACT REQUIREMENTS.....	67
<b>7</b>	<b>BIBLIOGRAPHY .....</b>	<b>68</b>
	<b>APPENDIX A.....</b>	<b>69</b>
	<b>APPENDIX B.....</b>	<b>71</b>

## List of Tables and Figures

Figure 2.2 Nature of the Scheduling Problem.....	3
Figure 2.4 Illustration of Greedy Activity-Selector, [2].....	6
Table 3.1.1 Four Satellites .....	13
Table 3.1.2 Hawaii Tracking Site .....	14
Table 3.1.3 Norway Tracking Site .....	14
Table 3.2.1 Formation Satellites .....	17
Table 3.2.2 Hawaii Tracking Site .....	17
Figure 5.1.1 Satellite Paths .....	21
Figure 5.1.2 Accesses to Tracking Sites .....	22
Table 5.1.3 Base Case Visibilities .....	23
Figure 5.1.4 Visibilities.....	24
Table 5.2.1 Scheduler Results.....	25
Figure 5.2.2 Total Demand, Opportunity and Loading Count per Satellite .....	26
Figure 5.2.3 Total Demand, Opportunity and Loading Time per satellite .....	27
Figure 5.2.4 Site Utilization Count by Satellite .....	28
Figure 5.2.5 Site Utilization Time by Satellite .....	29
Figure 5.2.6 Total Site Availability Count and Total Site Loading Count.....	30
Figure 5.2.7 Total Site Availability Time and Total Site Loading Time .....	31
Table 5.3.1 Scheduler Results.....	32
Figure 5.3.2 Total Demand, Opportunity and Loading Count per Satellite .....	33
Figure 5.3.3 Total Demand, Opportunity and Loading Time per Satellite .....	34
Figure 5.3.4 Site Utilization Count by Satellite .....	35
Figure 5.3.5 Site Utilization Time by Satellite .....	36
Figure 5.3.6 Total Site Availability Count and Total Site Loading Count.....	37
Figure 5.3.7 Total Site Availability Time and Total Site Loading Time .....	38
Table 5.4.1 Scheduler Results.....	39
Figure 5.4.2 Total Demand, Opportunity and Loading Count per Satellite .....	40
Figure 5.4.3 Total Demand, Visibility and Loading Time per Satellite .....	41
Figure 5.4.4 Site Utilization Count by Satellite .....	42
Figure 5.4.5 Site Utilization Time by Satellite .....	43
Figure 5.4.6 Total Site Availability Count and Total Site Loading Count.....	44
Figure 5.4.7 Total Site Availability Time and Total Site Loading Time .....	45
Figure 5.5 Network Support Performance Ratios .....	46
Figure 5.6.1 Visibilities.....	47
Figure 5.6.2 Total Demand, Visibility and Loading Count per Satellite .....	48

Figure 5.7.1 Satellite Paths .....	49
Figure 5.7.2 Accesses to Tracking Site .....	50
Table 5.7.3 Case 2 Visibilities .....	51
Figure 5.7.4 Visibilities.....	52
Table 5.8.1 Scheduler Results.....	53
Figure 5.8.2 Total Demand, Visibility and Loading Count per Satellite .....	54
Figure A.1 Illustration of Satellite Pass Time and Altitude .....	69
Figure A.2 Satellite Pass Time and Altitude.....	70

# Chapter 1

## 1 INTRODUCTION

### 1.1 Background

Since its dramatic public appearance over 40 years ago, space technology has spawned an ever-increasing number of diverse opportunities for improving the human condition. Formations of spacecraft now monitor Earth's global climate to help identify causes of global warming. Commercial communications spacecraft in geo-synchronous orbit routinely provide high-capacity channels, supporting much of the world's television, voice communication and other data. Large constellations of low-earth orbit communications satellites are now being deployed as the space-based equivalent of cellular phone technology. Military and commercial multi-spectral remote sensing satellites help monitor political and environmental trouble spots across the globe. The list goes on...

At some level, all of these types of spacecraft require a supporting infrastructure: their orbits and on-board systems must be monitored and controlled; the often vast amounts of mission data must be downloaded; communications uplinks, downlinks, and cross links must be rapidly established, released, and re-established periodically. All this requires distributed networks of ground-based sites or orbiting relay satellites that provide tracking and data acquisition (TDA) services.

From an economic viewpoint it is essential to make maximum utilization of these network infrastructures. Tracking stations are expensive to build, operate, and maintain. They are usually fixed, tending to involve large amounts of equipment, making mobility a daunting challenge. Most importantly, this almost always demands supporting a wide variety of satellite missions in order to achieve maximum utilization and be most cost-effective.

The network utilization and satellite support problem can be thought of in terms of time frames. At one end of the scale is real or near-real time operational satellite scheduling. This demands high precision modeling and rapid computation to schedule TDA contacts between existing satellites and networks. The other end – long term planning – deals with determining the adequacy of TDA networks in meeting projected TDA demands by future missions. The focus of this investigation is in the latter time frame – the problem of long-term planning and analysis to determine network adequacy for satellite support.

### 1.2 Previous Efforts

The satellite network allocation problem, discussed here, evolved from Veridian's history of working with the US Air Force and NASA on their problem of supporting growing numbers of complex missions needing tracking and data analysis. The 50<sup>th</sup> Space Wing defined a need to be able to project satellite telemetry, tracking and commanding (TT&C) capacity of the Air Force Satellite Control Network (AFSCN) over an extended period of time into the future. NASA had a similar need, compounded by a greater variety of mission and data. Veridian was contracted to develop such a capability using its mission planning tool the Computer-Aided System Engineering and Management Model (CASEMM). This planning issue is faced by all communication providers including, for

example, NASA communicating with deep space probes, the Air Force tracking defense satellites and commercial users managing telecommunications traffic loads.

Veridian's CASEMM tool is very complex and very computation intensive and therefore provides an adequate but time consuming solution. The intent of this project is to provide an alternative to the process by simplifying the problem and applying more rigorous methods.

### **1.3 Scope and Approach**

The simplified problem addressed here is the scheduling of multiple low altitude satellite communication contacts across one or more space tracking networks. Because the resources are always limited, the challenge of the mission planning problem is to maximize the number of satellite contacts that can be provided. In other words, to optimize the use of the tracking resources so as to provide the potential to meet the maximum number of satellite contact requirements. The optimal Greedy Activity-Selector algorithm is applied to determine how tracking sites should be allocated to accomplish contacts with satellites.

Analytical Graphic's Satellite Tool Kit is utilized to design a mission scenario and generate the satellite ephemeris data with respect to a basic tracking network. The data is sorted in the appropriate manner and imported into MATLAB scheduler code. Implementation of the Greedy Activity-Selector algorithm in MATLAB results in an optimal schedule of satellite contacts with respect to the tracking sites.

The practical application of the algorithm is primarily useful for long term mission planning, rather than actual, real-time satellite contact scheduling. The results of the scheduler are used to analyze contact requirements, contact demand and compatibility. This serves as the basis for assessing whether or not a network is adequate to meet support requirements projected for future missions.

Chapter 2 contains a definition of the Greedy Activity-Selector algorithm, defined by Cormen, Leiserson and Rivest of MIT, [2]. The proof that the Greedy Activity-Selector produces solutions of maximum size for the activity-selection problem is also included, [2]. In addition, Chapter 2 details the process of the implementation of the algorithm through STK and MATLAB. Scenario definitions and numerical examples of interest to this study are given in Chapter 3. . The conclusions for this study are given in Chapter 4. We provide plots and results in Chapter 5 and include MATLAB codes developed to implement the scheduler algorithm as Chapter 6.

## Chapter 2

### 2 MODELS AND METHODOLOGY

#### 2.1 Satellite Orbits

For reasons discussed later, satellites treated in this investigation consist of those in low earth orbit (LEO). While various rather arbitrary criteria are used in defining the altitude range for LEO satellites, the altitudes employed here range from approximately 315 to 1020 kilometers above the Earth's surface.<sup>1</sup>

In this investigation, the orbital motion of several satellites is simulated. The simulation does not require extensive accuracy, since the calculated results of scheduling satellite tracking resources are compared against each other. Hence it is relative accuracy, not absolute accuracy that matters most. As a consequence, only the major gravitational effects due to the Earth and its asphericity are incorporated. This is accomplished using the orbit propagation capabilities of AGI's Satellite Tool Kit.<sup>2</sup>

#### 2.2 Nature of the Scheduling Problem

Difficulties in the problem of allocating tracking resources to different satellites arise due to the fact that a single tracking site may be in view of multiple satellites at the same time. The issue is which of the simultaneously viewable satellites will get the track? This is an important question since the choice at one point in time can significantly affect the performance later on. Consider the figure below which shows the potential contacts and visibility periods for three different satellites by a single ground station.

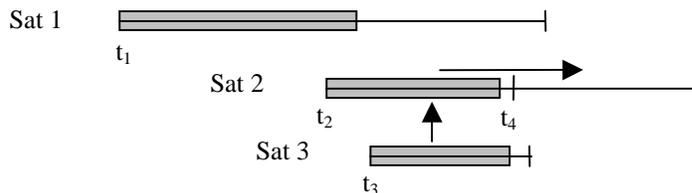


Figure 2.2 Nature of the Scheduling Problem

The shaded regions represent the duration of a satellite contact, and the single line represents the duration that the satellite is visible to the tracking station. Scheduling the satellite contacts based on a "first-come, first-serve" basis, and at the earliest time will result in *only the first satellite* being scheduled, since all the other satellites would be in conflict. However, scheduling Sat 1 at  $t_1$ , Sat 3 at  $t_3$ , and Sat 2 at  $t_4$  will result in *all three satellites* being scheduled. Thus the general challenge in this problem is to find an algorithm that can make this kind of resolution in a timely and dependable manner. The problem becomes vastly complicated when many satellites and opportunities (visibilities) are involved. Changing one scheduled instance can result in a drastically different,

<sup>1</sup> See Appendix A

<sup>2</sup> In particular, the J2 and J4 harmonics are included in addition to the normal spherical term.

subsequent schedule, and there can be an extremely large number of alternatives to explore.

In the case where the satellite contact opportunity (visibility of the satellite by the tracking site) is about the same as the contact demand (the duration of the satellite contact), the problem becomes greatly simplified. Unlike the situation depicted in the illustration (where the opportunity for Satellite 2 was large enough that the contact could be delayed significantly), when the opportunity is about the same as the duration there does not exist that “degree of freedom” to shift contacts around.<sup>3</sup> Thus, rather than tackle the very difficult general problem, this investigation addresses a simplified version, which turns out to be amenable to straightforward solution.

### 2.3 Procedure

For the scheduling approach taken in this project, satellite trajectory data are computed for a modest set of low altitude satellites. Corresponding visibilities are computed with respect to a basic network comprised of up to five tracking sites. Representative contact requirements, the terms for how the user requests satellite time for specific missions and satellites, are used to define the satellite tracking requirements. By simplifying the problem to only low altitude satellites, the contact times are approximately the same as the pass times. Therefore in the low altitude case, the potential for ‘sliding’ a satellite contact within the visibility window for scheduling conflicting events is practically eliminated.

Determining the altitude region over which this condition prevails is straightforward. Since the opportunity time increases with increasing satellite altitude, this altitude regime is defined by the altitude,  $h$ , lying in the interval,

$$h_{\text{minimum contact duration}} < h < h_{2(\text{minimum contact duration})}$$

or

$$315 \text{ km} < h < 1020 \text{ km}$$

(Details of the calculations are given in Appendix A). The Greedy Activity-Selector algorithm is then used to optimize the allocation of these visibilities to maximize the satisfaction of the satellite tracking requirements.

The Greedy Activity-Selector algorithm, defined by Cormen, Leiserson and Rivest, [2], is an optimal algorithm that solves the problem of scheduling competing activities upon a resource. A greedy algorithm is a simple, efficient algorithm that “makes a locally optimal choice in hope that this choice will lead to a globally optimal solution”.

### 2.4 Greedy Activity-Selector

The algorithm is reproduced here as a theorem and proof, from *Introduction to Algorithms* by Cormen, Leiserson, and Rivest of MIT, [2].

---

<sup>3</sup> Actual operational experience at the Air Force Satellite Control Network has shown this to be true. Controllers traditionally scheduled the low altitude satellites before tackling the higher altitude ones.

*Theorem*

Suppose we have a set  $S = \{1, 2, \dots, n\}$  of  $n$  proposed activities that wish to use a resource which can be used by only one activity at a time. Each activity  $i$  has a start time  $s_i$  and a finish time  $f_i$ , where  $s_i \leq f_i$ . If selected, activity  $i$  takes place during the half-open time interval  $[s_i, f_i)$ . Activities  $i$  and  $j$  are compatible if the intervals  $[s_i, f_i)$  and  $[s_j, f_j)$  do not overlap (i.e.  $i$  and  $j$  are compatible if  $s_i \geq f_j$  or  $s_j \geq f_i$ ). The activity-selection problem is to select a maximum-size set of mutually compatible activities.

The input activities are in order by the increasing finishing time:  $f_1 \leq f_2 \leq \dots \leq f_n$ .

$n \leftarrow \text{length}[s]$

$A \leftarrow \{1\}$

$j \leftarrow 1$

for  $i \leftarrow 2$  to  $n$

    do if  $s_i \geq f_j$

        then  $A \leftarrow A \cup \{i\}$

$j \leftarrow i$

return  $A$

The set  $A$  collects the selected activities. The variable  $j$  specifies the most recent addition to  $A$ . Since the activities are considered in order of nondecreasing finishing time,  $f_j$  is always the maximum finishing time of any activity in  $A$ .

That is,

$$f_j = \max\{f_k : k \in A\}.$$

The algorithm selects activity 1, initializes  $A$  to contain just this activity, and initializes  $j$  to this activity. Then each activity  $i$ , in order of decreasing finishing time, is considered and added to  $A$  if it is compatible with every activity currently in  $A$ .

The activity picked next by Greedy Activity-Selector is always the one with the earliest finish time that can be legitimately scheduled. The activity picked is thus a “greedy” choice in the sense that, intuitively, it leaves as much opportunity as possible for the remaining activities to be scheduled. That is, the greedy choice at each step is the one that maximizes the amount of unscheduled time remaining on a resource. This process of making locally optimal choices results in an optimal solution set containing the maximum number of contacts that can be scheduled upon the resource

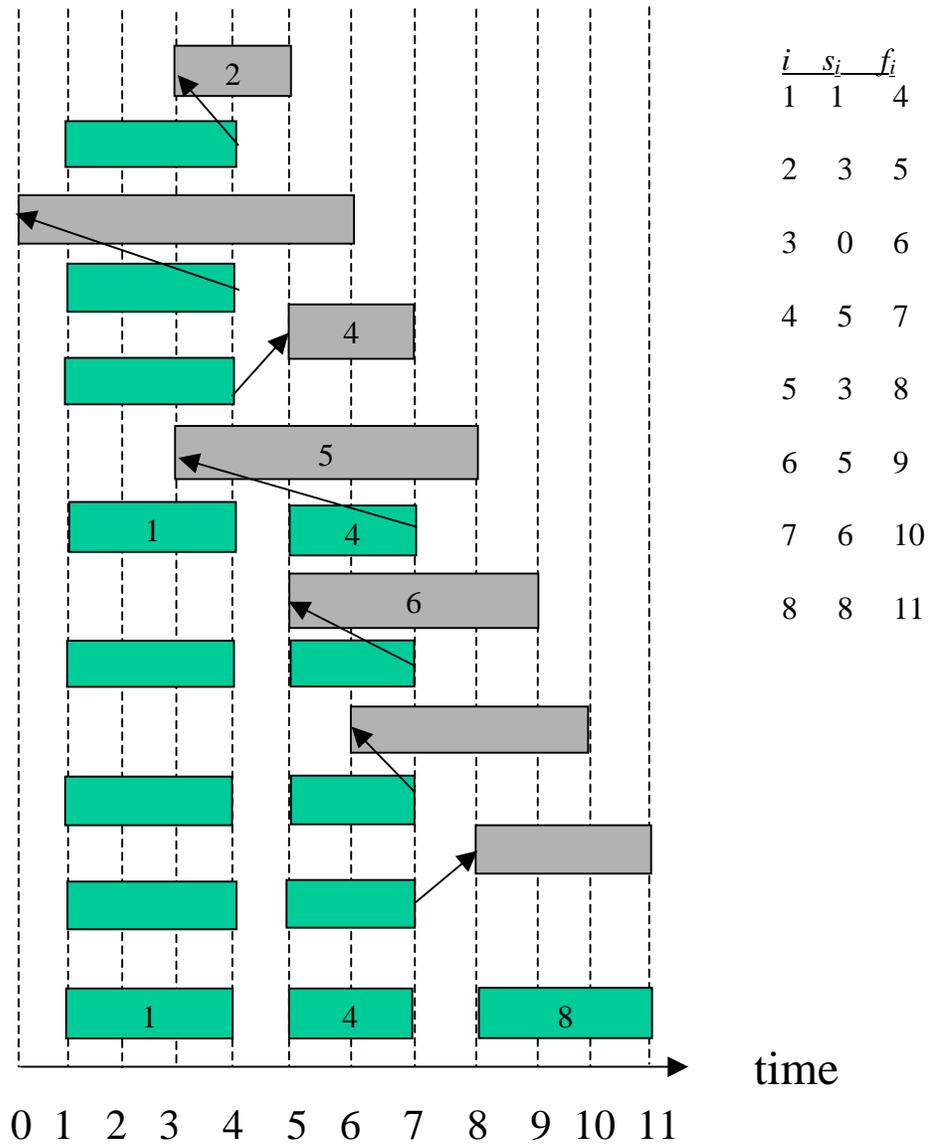


Figure 2.4 Illustration of Greedy Activity-Selector, [2]

## 2.5 Proving the greedy algorithm correct

Greedy algorithms do not always produce optimal solutions. However, the Greedy Activity-Selector always finds an optimal solution to an instance of the activity-selection problem.

### 2.5.1 Theorem

Algorithm Greedy Activity-Selector produces solutions of maximum size for the activity-selection problem, [2].

#### Proof

Let  $S = \{1, 2, \dots, n\}$  be the set of activities to schedule. Since we are assuming that the activities are in order by finish time, activity 1 has the earliest finish time. We wish to show that there is an optimal solution that begins with a greedy choice, that is, with activity 1.

Suppose that  $A \subseteq S$  is an optimal solution to the given instance of the activity-selection problem, and let us order the activities in  $A$  by finish time. Suppose further that the first activity in  $A$  is activity  $k$ . If  $k = 1$ , then schedule  $A$  begins with a greedy choice. If  $k \neq 1$ , we want to show that there is another optimal solution  $B$  to  $S$  that begins with the greedy choice, activity 1. Let  $B = A - \{k\} \cup \{1\}$ . Because  $f_1 \leq f_k$ , the activities in  $B$  are disjoint, and since  $B$  has the same number of activities as  $A$ , it is also optimal. Thus,  $B$  is an optimal solution for  $S$  that contains the greedy choice of activity 1. Therefore, we have shown that there always exists an optimal schedule that begins with a greedy choice. Moreover, once the greedy choice of activity 1 is made, the problem reduces to finding an optimal solution for the activity-selection problem over those activities in  $S$  that are compatible with activity 1. That is, if  $A$  is an optimal solution to the original problem  $S$ , then  $A' = A - \{1\}$  is an optimal solution to the activity-selection problem  $S' = \{i \in S : s_i \geq f_1\}$ . Why? If we could find a solution  $B'$  to  $S'$  with more activities than  $A'$ , adding activity 1 to  $B'$  would yield a solution  $B$  to  $S$  with more activities than  $A$ , thereby contradicting the optimality of  $A$ . Therefore, after each greedy choice is made, we are left with an optimization problem of the same form as the original problem. By induction on the number of choices made, making the greedy choice at every step produces an optimal solution.

## 2.6 Relationship of the Greedy Activity-Selector to the General Satellite Scheduling Problem

Since the Greedy Activity Selector addresses only a portion of the general satellite scheduling problem, it's appropriate to examine its relationship to the general problem. We saw that the optimality of the Activity-Selector tends to maximize the time remaining on the network as each satellite is scheduled. We can extend this concept to the general problem, albeit somewhat heuristically.

Define a performance metric,  $P_k$ , for any stage in the general scheduling process as

$$P_k = \frac{\sum_i^n \sum_j^m v_{ij} - \sum_i^n \sum_k^l c_{ik}}{\sum_i^n \sum_j^m v_{ij}} = 1 - \frac{\sum_i^n \sum_k^l c_{ik}}{\sum_i^n \sum_j^m v_{ij}}$$

where  $c_{ik}$  is the  $k^{\text{th}}$  contact required by the  $i^{\text{th}}$  satellite, and  $v_{ij}(t)$  is the  $j^{\text{th}}$  visibility or access of the  $i^{\text{th}}$  satellite. The sum on  $j$  is over all the  $m$  accesses for the period of interest (e.g., one day). Here, for simplicity, the notation makes no distinction with regard to tracking site. Thus, if the period of interest is one day, and there are a total of 15 accesses for satellite  $i$ , and there are 10 required contacts, then  $m = 15$  and  $l = 10$ .

In general, there may be a number,  $s$ , of (essentially) simultaneous accesses for each potentially "schedulable" contact, and there are  $g$  groups of such simultaneous contacts so that the total number of accesses is  $m = \sum_q^g s(q)$ . For example, these simultaneous accesses could occur on multiple antennas of the same tracking station or even multiple stations. Then the performance metric can be written as

$$P_k = 1 - \frac{\sum_i^n \sum_k^l c_{ik}}{\sum_i^n \sum_j^m v_{ij}} = 1 - \frac{\sum_i^n \sum_k^l c_{ik}}{\sum_i^n \sum_q^g \sum_r^{s(q)} v_{iqr}}$$

The summation limit  $s$  is a function of the summation index  $q$ , since, in general the number of simultaneous accesses,  $s$ , varies by group. For example, one group may have 3 simultaneous accesses, another, 2 accesses, and yet another, 4.

This metric has the following property: at each stage in a general scheduling algorithm, if we always schedule the event for which the ratio

$$\delta_{ijk} = \frac{c_{ik}}{v_{ij}}$$

is the largest in the set, then at each stage  $P_k$  will be maximized. This conforms nicely to the idea that maximizing the remaining time at each stage implies that there will be more time available to schedule succeeding events.

Note that when  $c_{ik}/v_{ij}$  is near or equal to 1,  $P_k$  tends to be maximized at the corresponding stage,  $k$ . To ensure that  $P_k$  is always maximum at each stage, we want to schedule first

those satellites that have  $c_{ik}/v_{ij} \approx 1$ . Thus, applying the Greedy Activity-Selector algorithm first to those low altitude satellites implies that it is the initial *optimal* “phase” of a more general satellite scheduling scheme.

### 2.6.1 Simple Example

To illustrate with a simple example, assume that all the contacts and all the accesses are equal to a constant. Further, assume also that the number of simultaneous accesses,  $s$ , is the same for each group,  $g$ .

$$c_{ik} = v_{ij} = c$$

$$s(q) = s, \text{ for all } q$$

Then the performance metric becomes

$$P_k = 1 - \frac{\sum_i^n \sum_k^l c_{ik}}{\sum_i^n \sum_q^g \sum_r^s v_{iqr}} = 1 - \frac{n(l-k)c}{ns(g-k)c} = 1 - \frac{(l-k)}{s(g-k)}$$

Here there are  $m = gs$  accesses, and for each scheduled contact we remove  $s$  of those accesses, since they correspond to the same satellite (once it’s scheduled those accesses no longer apply for that opportunity).

For the zero<sup>th</sup> stage (before any contacts have been scheduled),

$$P_0 = 1 - \frac{l}{m} = 1 - \frac{l}{sg}$$

If 10 contacts are required per satellite, and there are 2 accesses per group for 10 groups (20 opportunities total per satellite), then  $P_0 = 0.5$ . At the first stage, when the first contact is scheduled ( $k = 1$ ),

$$P_1 = 1 - \frac{(l-k)}{s(g-k)} = 1 - \frac{(10-1)}{2(10-1)} = 0.5$$

At each succeeding stage in this example, the number of contacts scheduled reduces the numerator by 1. The removal of simultaneous accesses reduces the denominator by 2. The result is that  $P_k = 0.5$  for all  $k$ . This fact also holds true for any other case (in the example) where the number of simultaneous groups equals the number of required contacts, i.e.,  $g = l$  (for  $g > l$ ,  $P_k$  is *not* constant). In the case where  $s = 1$  (only one access per contact), note that  $P_k = 0$ . Conversely, for more simultaneous opportunities, the higher the value of  $P_k$ , and, presumably, the higher the likelihood of scheduling contacts. Thus, for constant contact duration,  $c$ , constant access time,  $c$ , and when the number of simultaneous access groups,  $g$ , equals the number of required contacts, the performance metric is constant from stage to stage:

$$P_k = P_{k'}, \text{ where } k' \text{ is any other value of } k.$$

This relationship is a good candidate topic for a future investigation based on extending the concept of maximizing  $P_s$ . One could consider a scheduling scheme based on scheduling

contacts in order of decreasing values of  $\delta$  (ie., schedule first those events that have the highest values of  $\delta$ ). *A priori*, it is not clear that such a scheme would produce an optimal schedule. However, in the simple example above, both approaches *appear equivalent* in terms of  $P_k$  – in both approaches, those contacts where  $\delta$  equals 1 are scheduled first. This suggests that the Greedy Activity-Selector algorithm is an optimal subset of a more general (potentially optimal) scheduling algorithm, and that further, the approach of scheduling based on decreasing order of the value of  $\delta$  is a potentially fruitful approach.

## 2.7 Implementation of the Greedy Algorithm

Implementation of the Greedy Activity-Selector algorithm in MATLAB provides an efficient and straightforward method of solving a simplified, scheduling problem. The mission scenario created in Satellite Tool Kit may define multiple low altitude satellites and multiple tracking stations. The accesses, or visibilities, are calculated for the satellites with respect to the tracking sites. Visibilities are those periods when the satellite is visible to a tracking station defined by the time between satellite rise and satellite set at the site, [4]. The accesses are exported from STK and are sorted, as defined by the algorithm, in order by increasing finishing time. At execution of the MATLAB scheduler code, the sorted accesses are then imported into MATLAB. The scheduler program proceeds to implement the algorithm in an extremely timely manner. The result of reconciling the user's contact requirements and demand with the visibilities produce viable contact opportunities. An opportunity is an time period during which a single contact event may possibly be scheduled, [4]. User constraints of a minimum and maximum required contact time eliminate insufficient passes and strictly define the final contact length. Each opportunity's compatibility is evaluated against the previously scheduled contacts based on the access's start and stop times as well as the site on which the access occurs. If two adjacent contacts wish to occur on the same tracking site, a margin of setup time must be allowed. This setup time is the time required for the tracking antenna to slew from horizon to horizon. Finally, a set of satellite contacts is compiled that represent the schedule that optimizes the use of the given network resources. This loading upon the resources is represented in both count (number of occurrences) and time.

The results of the scheduler are measured in various graphs and ratios. There are a number of useful ratios that summarize the network support performance, [3].

### 2.7.1 Opportunity/Demand Count Ratio (# of opportunities) / (# of required contacts)

This ratio is a measure of the proper definition of contact requirements relative to the simulated network. A low value indicates poor or overspecified requirements relative to network capabilities. A value near 1 indicates good specification of requirements.

### 2.7.2 Scheduler Performance Ratio (# of scheduled contacts) / (# of opportunities)

This is a quick measure of overall scheduler performance. A value near 1 indicates good scheduler performance. A low value indicates significant event conflicts or poor performance.

### 2.7.3 Schedule/Demand Count Ratio (# of scheduled contacts) / (# of required contacts)

This ratio is a measure of the scheduling of contact requirements. A low value indicates poor or overspecified contact requirements relative to network capabilities, weak network capabilities, or poor scheduler performance. A value near 1 indicates good specification of requirements and performance. Note that this ratio equals the product of the other two.

#### 2.7.4 Site Loading / Availability Count Ratio

(# of scheduled contacts) / (# of available opportunities)

This ratio is a measure of the use of a tracking site. Values near 1 mean that all the available opportunities are used to fulfill contact requirements.

## Chapter 3

### 3 EXAMPLES AND SCENARIOS

We consider the following examples to validate the scheduling technique and analyze the results. We start with a basic scenario of four satellites and one tracking station. Starting with the basic scenario, we'll look at the fundamentals of the scheduling problem and continue on to more complex scenarios and examination.

The following scenarios run for one day, from July 1, 1999 to July 2 1999. The epoch date is July 1, 1999.

#### 3.1 Basic Case 4 Satellites, 1 Tracking Site

The satellites in this scenario are the low earth-orbiting satellites: EOS Chem-1, QuickSCAT, Landsat-7, and ICESat. These satellites are all sun-synchronous which means that they pass over the equator at the same local time with every revolution. In a sun-synchronous orbit, the angle between the orbital plane and the Sun remains constant, resulting in consistent light conditions for the satellite. This type of orbit is therefore beneficial for an Earth observing satellite, [5]. The tracking site in this scenario is South Point, Hawaii located in the Hawaiian Big Island.

The following values, called orbital elements, are used to completely describe the size, shape, and orientation of an orbit:

a = Semi-major Axis

e = Eccentricity

i = Inclination

$\Omega$  = Right Ascension of the Ascending Node

$\omega$  = Argument of Periapsis

A sixth element defines the instantaneous location of the satellite within its orbit. Various measures are used, but all provide essentially the position at some reference time (epoch). The one most commonly used is the mean anomaly at epoch,  $M_0$  (see Appendix B for further discussion). In this scenario, the mean anomaly at epoch for each orbit is 0.0 deg.

Table 3.1.1 Four Satellites

Satellite	a (km)	e	i (deg)	$\Omega$	$\omega$	Altitude
EOS Chem-1	7083.137	0	98.2131	124.8673	0	705 Km C
ICESat	6978.137	0	94	94	0	600 Km C
Landsat-7	7083.137	0	98.2131	248.6147	0	705 Km C
QuickSCAT	7181.137	0	98.6206	8.6147	0	803 Km C

Table 3.1.2 Hawaii Tracking Site

<b>Tracking Site</b>	<b>Location</b>	<b>Latitude</b>	<b>Longitude</b>	<b>Elevation Angle</b>
South Point Hawaii	Hawaiian Big Island	19.014	-155.663	5 deg

One orbit path of these four satellites can be seen in Figure 5.1.1 as well as the accesses with respect to the Hawaiian Site (as well as the Norwegian Site, discussed below) in Figure 5.1.2. The visibilities of the satellites with respect to the tracking station are computed using Satellite Took Kit. Figure 5.1.3 and Figure 5.1.4 show the accesses of the satellites with respect to the Hawaiian station.

In the following examples, the demand is adjusted to demonstrate network saturation. Saturation is the point at which the network is so loaded that it can no longer accommodate the requirements for satellite contacts. Once the network is saturated there are modifications that can be made to improve the likelihood of scheduling. Options include increasing the number of available antennas to the existing tracking site or adding robustness to the network by adding new tracking sites to the scenario. Detailed in the examples below, it is apparent that the Hawaiian tracking station is incapable of handling a demand of more than 4 contacts per day for each satellite. Adding an additional antenna to the Hawaii Site would merely double the already low number of accesses to the satellites, but adding a new site would add an entirely new set of accesses. Selecting a site such as the KLM SDS station in Svalbard, Norway would provide a greater number of accesses due to the site's high latitude. Orbits with inclinations,  $i$ , such as those in our example, are near-polar orbits. This means that every revolution passes near both the north and south poles. The coverage of a satellite encompasses the entire globe, but there may be long periods when the satellite will not be in view of a particular ground station, depending upon its latitude. Yet stations at high latitudes will be in view on nearly every orbit revolution.

Table 3.1.3 Norway Tracking Site

<b>Tracking Site</b>	<b>Location</b>	<b>Latitude</b>	<b>Longitude</b>	<b>Elevation Angle</b>
KLM SDS	Svalbard, Norway	78.217	15.383	5 deg

The network performance ratios of the following examples in Case 1 are compared in Figure 5.5. The following mission analysis is valuable to determine availability of network support and resource capacity.

3.1.1 Let the contact requirements be 2 contacts per day for each satellite.

In this case, there is a margin of 1 to 2 contacts for each satellite before the network is saturated. See Figure 5.2.1 for the scheduler results. As can be seen in Figure 5.2.2, the visibilities (contact opportunities) outnumber the demand. Figure 5.2.6 and Figure 5.2.7, show that there is additional site availability beyond the actual loading upon the site. The loading is a mere 66% of the availability.

The opportunity/demand count ratio for this example is 1.5, the scheduler performance ratio is 0.6667 and the schedule/demand count ratio is 1.0. The schedule/demand count ratio being equal to 1.0 reflects that every requested contact was scheduled, but in this case there are more accesses, or visibilities, than there are requests for contacts.

3.1.2 Let the contact requirements be 4 contacts per day for each satellite. With a request for 4 contacts per day for each satellite, the margin is 0 and the network becomes saturated. At this point, the current network resources may no longer accommodate required contacts and even some of the requirements are left unmet. See Figure 5.3.1 for the scheduler results. Note the unmet contact demand in Figure 5.3.2 and the fully utilized availability, or 100% utilization, of the resources in Figure 5.3.6.

The opportunity/demand count ratio for this example is 0.75, the scheduler performance ratio is 1.0 and the schedule/demand count ratio is 0.75. So in this case where saturation occurs, reflected by the ratio of schedule/opportunity being equal to 1.0, all visibilities are the non-conflicting and are used to schedule contacts. At this point, there are also excess demands that can not be handled by the current network (confirmed by the fact that the schedule/demand count ratio is 0.75).

3.1.3 Let the contact requirements be 5 contacts per day for each satellite. By demanding 5 contacts per day for each satellite, there is a margin of saturation of  $-2$ . The current network resources are highly over saturated. See Figure 5.4.1 for the scheduler results. Note from Figure 5.4.2 that the user requests for contacts exceed the visibilities with respect to the tracking site.

The opportunity/demand count ratio for this example is 0.6, the scheduler performance ratio is 1.0 and the schedule/demand count ratio is 0.6. Although contacts are being scheduled during all available accesses, the demand is much too high to be realized on the limited tracking resources.

## 3.2 Satellite Formations

The case of satellites flying in close proximity to one another – in formation – is a situation deserving special attention. When running a scenario with the entire example set of seven satellites, it was found that many of the accesses overlapped and therefore the contacts were in conflict and the requests could not be scheduled. See overlapping visibilities in Figure 5.6.1 Upon examination, the reason for the poor scheduling performance is found to be due to a high number of requests on closely overlapping or simultaneous events for satellites that have very similar orbital paths. See the mission's poorly met demands in Figure 5.6.2.

Consider two satellites,  $i$  and  $j$ , that are within an essentially fixed time,  $\Delta t_{ij}$ , of each other. Call this a static formation of satellites. Further, suppose that  $\Delta t_{ij}$  is less than the time required to slew the tracking antenna from horizon to horizon (call this time the maximum

antenna slew time,  $t_{\max\text{slew}}$  of any antenna in the tracking network). Then, if  $\Delta t_{ij} < t_{\max\text{slew}}$ , the combined contact demand for the two satellites can be represented (for tracking and data acquisition scheduling purposes) as a single satellite having their combined demand.

For example, suppose the two satellites were in the same orbit, but separated within 1 minute of each other. Typically the slew time of an antenna might be a few minutes so that (in the worst case), after tracking one satellite the antenna might require this few minutes to slew back to setup to track the next satellite. Since this next satellite (in this example) comes over the horizon a minute after the first, there is not enough time to get the antenna back for a full track of the second satellite. Thus if satellite  $i$  had a demand of 6 contacts per day and satellite  $j$  had a demand of 4 contacts per day, then the combined demand is 10 contacts per day. If the number of accesses turned out to be 6 per day, then the second satellite would *never* be tracked, based on scheduling using the optimum utilization algorithm, since it is always the first one in view of a tracking site, and hence always the one scheduled. While one could conceive of some other algorithm that might provide an average or proportional allocation of the tracking time (say, 3-4 per day for satellite  $i$  and 2-3 for satellite  $j$ ), such an approach might not provide the needed contacts, and might not, in general, make maximum use of the tracking resources.

In either case, the number of accesses would have to be increased. To calculate the required number of accesses, the combined demand can be used and the formation of satellites replaced by a single satellite (call it the *formation equivalent*). In the example, the two satellites would be replaced by a formation equivalent having the orbit of one of the two satellites (or some average), and a daily contact demand equal to the combined demand of 10 contacts per day.

Generalize these concepts in the following way, [6]:

Let  $F_n$  be the set of  $n$  satellites, all of which have a separation time not exceeding the maximum antenna slew time,  $t_{\max\text{slew}}$ :

$$F_n = \{n \text{ satellites} \mid \max_{i,j=1,2,\dots,n; i \neq j} (\Delta t_{ij}) \leq t_{\max\text{slew}}\}$$

Let  $D(F_n) = D_n$  represent the total demand for all satellites in the formation,  $F_n$ , and

$$D_n(F_n) = \sum_{k=1}^n d_k$$

where  $d_k$  is the individual satellite daily contact demand for each of the  $n$  satellites in the formation,  $F_n$ .

Let  $V(F_n) = V_n$  represent the total accesses for all  $n$  satellites in the formation,  $F_n$ . Define the *formation equivalent accesses* as the average of the total number of accesses  $V_n$  by all satellites in the formation:

$$\bar{V}_n = \frac{V_n}{n} = \frac{1}{n} \sum_{k=1}^n v_k$$

where  $v_k$  is the individual access for each of the  $n$  satellites in the formation,  $F_n$ .

(Remember that the *formation equivalent* is a single satellite whose elements are characteristic of the formation.) Then the *formation equivalent accesses* (the number of

accesses of the formation equivalent) must be greater than or equal to the total demand of all  $n$  satellites in the formation; i.e.,

$$\frac{1}{n} \sum_{k=1}^n v_k \geq \sum_{k=1}^n d_k$$

or

$$\bar{V}_n \geq D_n$$

Usually,  $\bar{V}_n$  is the same as any of the  $v_k$  in the formation, since the formation moves as a unit. Occasionally, however, it may happen that a few of the satellites have one less access than the rest of the formation. This reduces the total number,  $V_n$ , and the formation equivalent accesses is a fraction (but close to an integer). Over long periods of time (say, weeks to months and longer) this is consistent with the intent of projecting network capacity. Over short periods (days), however, the fractional value would have to be rounded up to the next whole integer in order to ensure adequate accesses for the formation to have its required contacts scheduled. (Note that this requirement exists regardless of the scheduling algorithm used.)

### 3.2.1 Satellites in Formation

The following case is a model of three satellites in formation. One orbit path for each of these three satellites can be seen in Figure 5.7.1 as well as the accesses computed by STK, with respect to the Hawaiian Site, in Figure 5.7.2. See the table and diagram of the accesses in Figure 5.7.3 and Figure 5.7.4.

Table 3.2.1 Formation Satellites

Satellite	a (km)	e	I (deg)	$\Omega$	$\omega$	Altitude
EO-1	7083.137	0	98.2131	248.8647	0	705 Km C
EOS AM-1	7083.137	0	98.2131	253.6147	0	705 Km C
Landsat-7	7083.137	0	98.2131	248.6147	0	705 Km C

Table 3.2.2 Hawaii Tracking Site

Tracking Site	Location	Latitude	Longitude	Elevation Angle
South Point Hawaii	Hawaiian Big Island	19.014	-155.663	5 deg

With a contact demand of 2 contacts per day per satellite, the scheduler merely achieves a scheduler performance ratio of 0.3333. In other words, only 2 contacts out of 6 support requests can be scheduled for the mission due to the overlapping of the three satellites. In this scenario the setup time, or antenna slew time, for the tracking station is 5 minutes. Due to the fact that the time between each satellite's pass is less than one minute on average and is less than the setup time, the satellites are considered in formation. Therefore,

another way of representing this formation, for purposes of our mission model, is to replace it by a formation equivalent. See Figure 5.8.1 for the scheduler results.

By the definition above, we have a formation of 3 satellites; hence,

$$F_3 = \{3 \text{ satellites} \mid \max_{i,j=1,2,3;i \neq j} (\Delta t_{ij}) \leq t_{\max \text{ slew}}\}$$

where

$$\Delta t_{ij} \approx 1 \text{ minute}$$

and

$$t_{\max \text{ slew}} = 5 \text{ minutes};$$

The total demand for all the satellites in formation,  $F_3$ , is represented by

$$D_3(F_3) = \sum_{k=1}^3 d_k$$

where in this case  $d_1 = d_2 = d_3 = 2$  contacts per day. Therefore  $D_3 = 6$  contacts per day.

Calculate the formation equivalent accesses for this case.

$$\bar{V}_3 = \frac{V_3}{3} = \frac{1}{3} \sum_{k=1}^3 v_k$$

The visibilities for each satellite in the  $F_3$  formation are  $v_1 = v_2 = v_3 = 2$ .

Let  $V_n = V_3 = 6$  be the total number of visibilities for all satellites in  $F_3$ .

Therefore,  $\bar{V}_3 = 2$ .

It's defined above that the formation equivalent accesses,  $\bar{V}_3$ , must be greater than the total demand of all n satellites,  $D_3$ . In this case this requirement is not met, hence the mission scenario must be altered in order to obtain the required number of accesses, 6. Adding antennas to the current tracking station, or adding additional stations, will resize the network in order to meet the formation problem.

The formation definition helps assess the network deficiencies and lend focus to modifications that are needed in order to meet mission requirements.

## Chapter 4

### 4 CONCLUSIONS

- An optimal algorithm has been successfully applied to a subset of the general satellite scheduling problem. This investigation has examined a straightforward solution to a problem of satellite network resource allocation. By making some basic simplifications, we are able to tackle a portion of a difficult scheduling problem. The limitation of the mission scenarios to Low Earth Orbiting satellites allows the effective application of the rigorous methods of the Greedy Activity-Selector algorithm. The Activity-Selector's manner of making the optimal choice for every sub-problem leads to a globally optimal solution. The greedy algorithm provides a method for optimizing the use of the tracking resources, which in turn serves as a surrogate for maximizing the number of satellite contacts.
- Selected metrics have been demonstrated to be useful in characterizing and analyzing mission scenarios and their corresponding network performance. These include the various scheduling performance ratios (section 2.7) and the formation equivalent parameter (section 3.2).
- A combination of existing software tools has been effectively and rapidly applied to a difficult resource allocation problem. By utilizing Analytical Graphic's Satellite Tool Kit, we can efficiently propagate the orbits of the satellites and determine accurate visibility periods with respect to the tracking sites. Integration of the STK data, user requirements and constraints, and the implementation of the algorithm in MATLAB result in an effective process for solving the scheduling problem. In Chapter 4, we presented assorted examples of simple mission planning scenarios. The four-satellite scenario demonstrates the results of varying contact demand. The analysis of the scheduler performance illustrates cases of the weak use of resources, the point of resource saturation and highly saturated and therefore insufficient resources. The formation scenario demonstrates how mission scenarios can be effectively modeled for analytical purposes. The numerical and visual results of the scheduler as well as specific network performance ratios help make this type of mission analysis valuable in determining compatibility of satellites, availability and allocation of network support and resource capacity. This serves as the basis for assessing whether or not a network is adequate to meet support requirements projected for future missions.
- Results of the investigation form a foundation for extending the work to the more general satellite scheduling problem. The scheduling technique presented here provides a solution to the first step to further solving the greater resource allocation problem. Shifting contacts within opportunity windows as well as giving priorities to specific requirements are topics for future work. It would also be valuable to follow-on with development of additional user interface for more flexibility.

## **Chapter 5**

### **5 MATLAB AND STK DATA AND GRAPHS**

#### **5.1 Charts and Graphs for Case 1 – 4 Satellites, 2 Tracking Stations**

5.1.1 Satellites (EOS Chem-1, QuickSCAT, Landsat-7, ICESat)  
Tracking Stations (Hawaii, KLM SDS Norway)

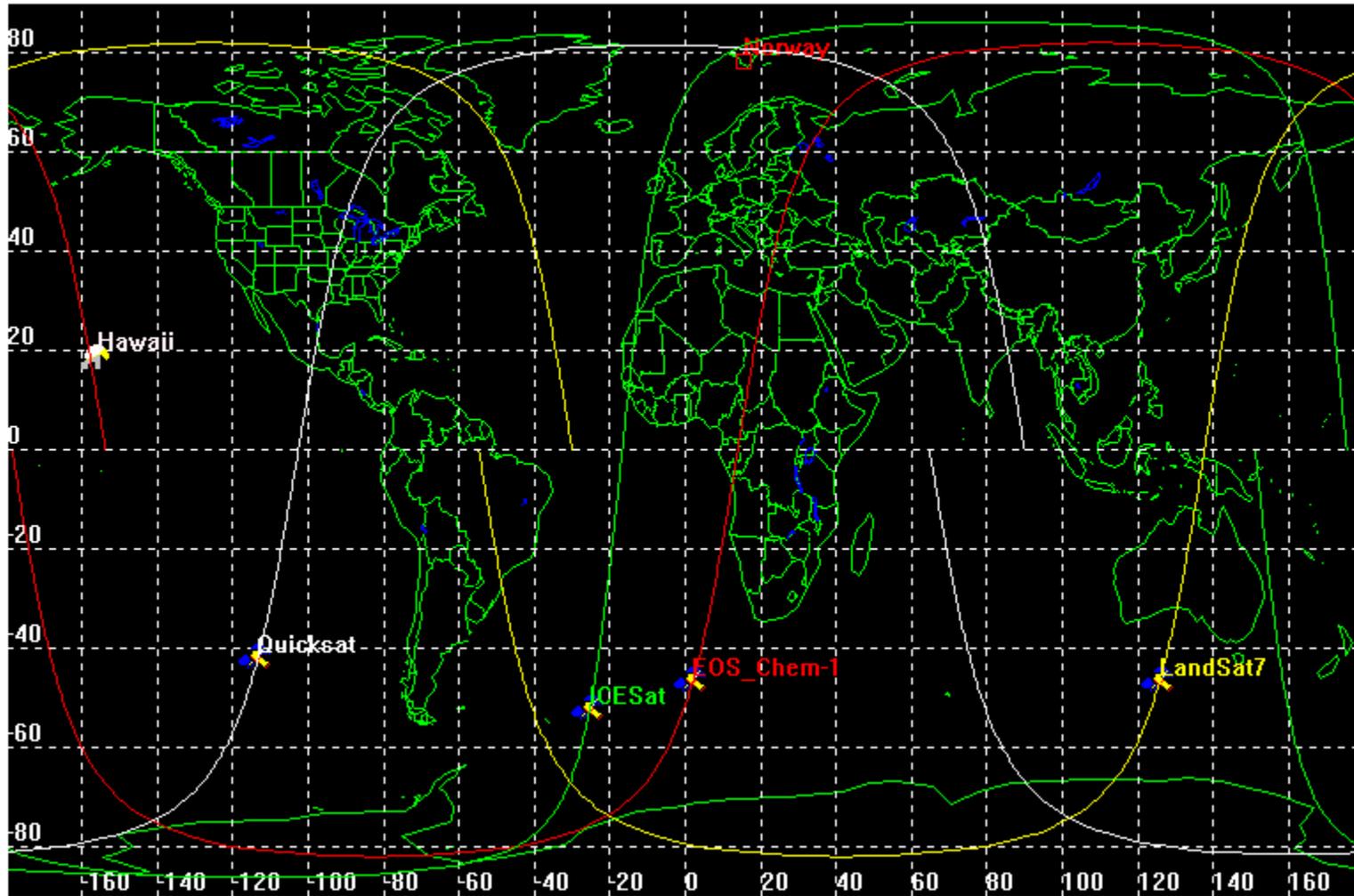


Figure 5.1.1 Satellite Paths

5.1.2 Accesses of 4 Satellites with respect to Hawaii and Norway Tracking Sites

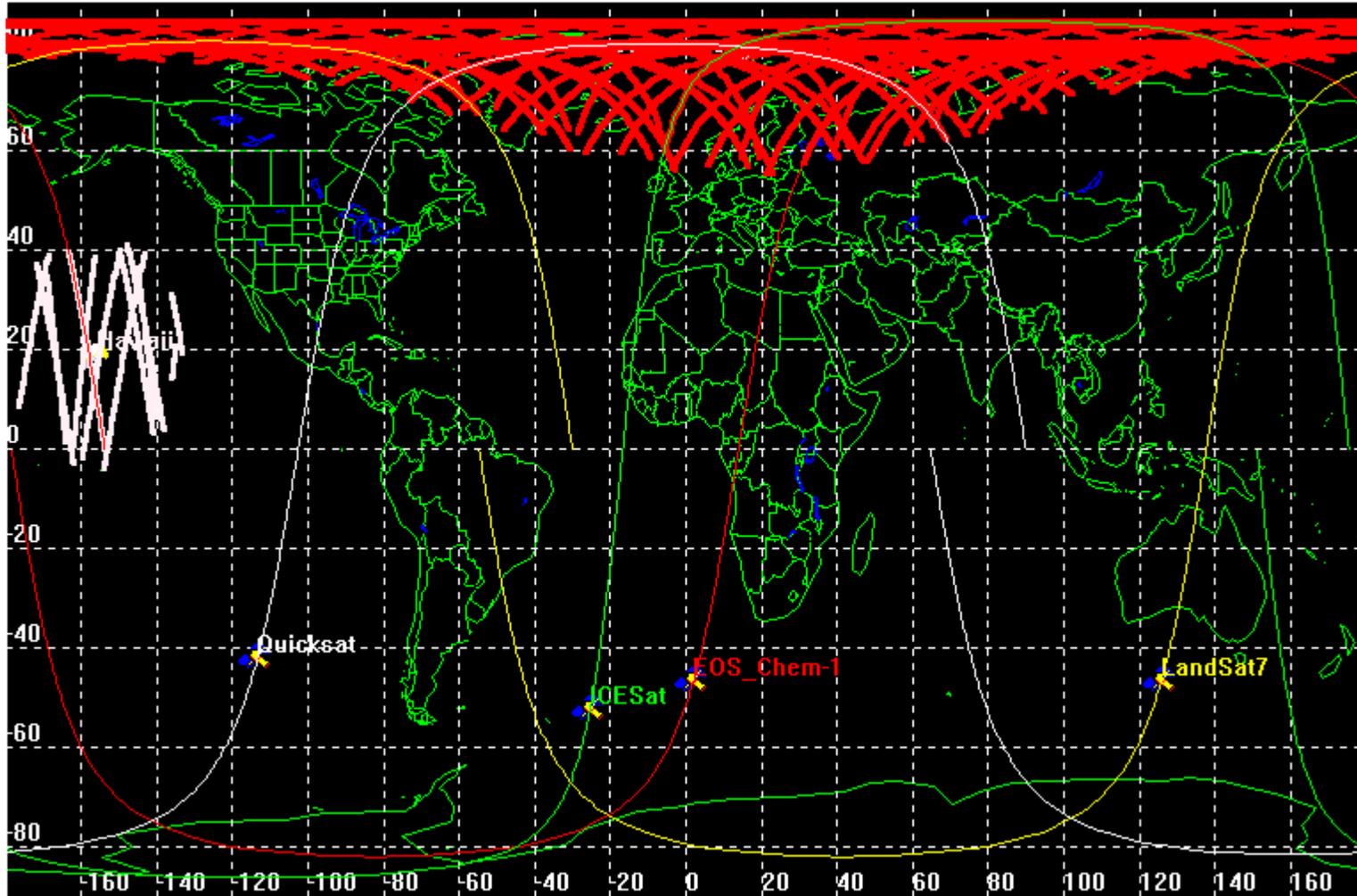


Figure 5.1.2 Accesses to Tracking Sites

### 5.1.3 Satellite visibilities calculated in STK

Sites are : South Point, Hawaii  
 from file : 4sat\_1site.dat  
 Epoch date is July 1, 1999 (2451360.5 JD)

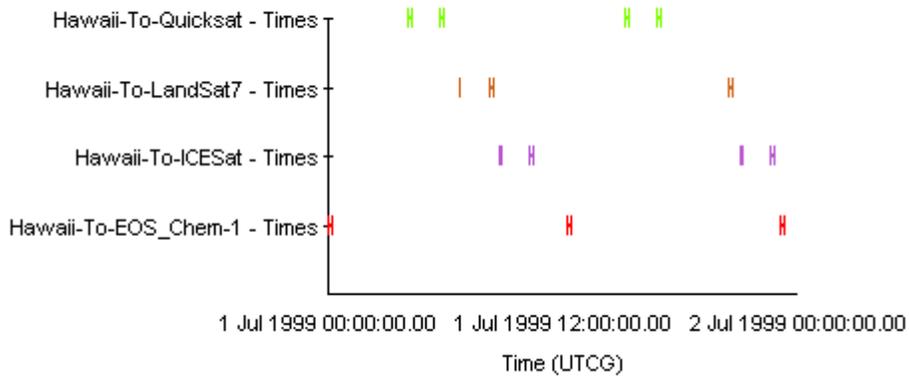
Table 5.1.3 Base Case Visibilities

Satellite	Site	Start Relative Min	Stop Relative Min	Duration (min)
EOS Chem-1	Hawaii	0	10.91908783	10.919
QuickSCAT	Hawaii	241.5013918	253.7033761	12.202
QuickSCAT	Hawaii	342.6435354	351.2881442	8.645
Landsat-7	Hawaii	401.261472	404.5430884	3.282
Landsat-7	Hawaii	494.3898719	505.9051196	11.515
ICESat	Hawaii	526.9667045	528.6193918	1.653
ICESat	Hawaii	618.9351837	629.1485281	10.213
EOS Chem-1	Hawaii	731.443968	743.002992	11.559
QuickSCAT	Hawaii	910.5782401	921.0991681	10.521
QuickSCAT	Hawaii	1009.708848	1021.041647	11.333
Landsat-7	Hawaii	1226.429712	1237.988736	11.559
ICESat	Hawaii	1260.292752	1268.330688	8.038
ICESat	Hawaii	1355.794416	1364.708593	8.914
EOS Chem-1	Hawaii	1386.886752	1397.004912	10.118

### 5.1.4 Satellite Visibilities with Respect to Hawaii Tracking Site

Figure 5.1.4 Visibilities

Facility-Hawaii-To-Satellite-EOS\_Chem-1, Satellite-ICESat, Satellite-LandSat7, Satellite-Quicksat: Access Time



**5.2 Scheduler Results for Case 1 – 4 Satellites, 1 Tracking Station;  
Contact Requirements = 2**

5.2.1 Successfully Scheduled Contacts

Table 5.2.1 Scheduler Results

Satellite	Site	Start Relative Min	Stop Relative Min	Duration (min)
EOS Chem-1	Hawaii	0	10.919002	10.919
QuickSCAT	Hawaii	241.5013918	252.5013938	11
QuickSCAT	Hawaii	342.6435354	351.2885325	8.645
Landsat-7	Hawaii	494.3898719	505.3898739	11
ICESat	Hawaii	618.9351837	629.148182	10.213
EOS Chem-1	Hawaii	731.443968	742.44397	11
Landsat-7	Hawaii	1226.429712	1237.429713	11
ICESat	Hawaii	1260.292752	1268.330746	8.038

Epoch date is July 1, 1999 (2451360.5 JD)

## 5.2.2 Total Demand, Opportunity and Loading Count per Satellite

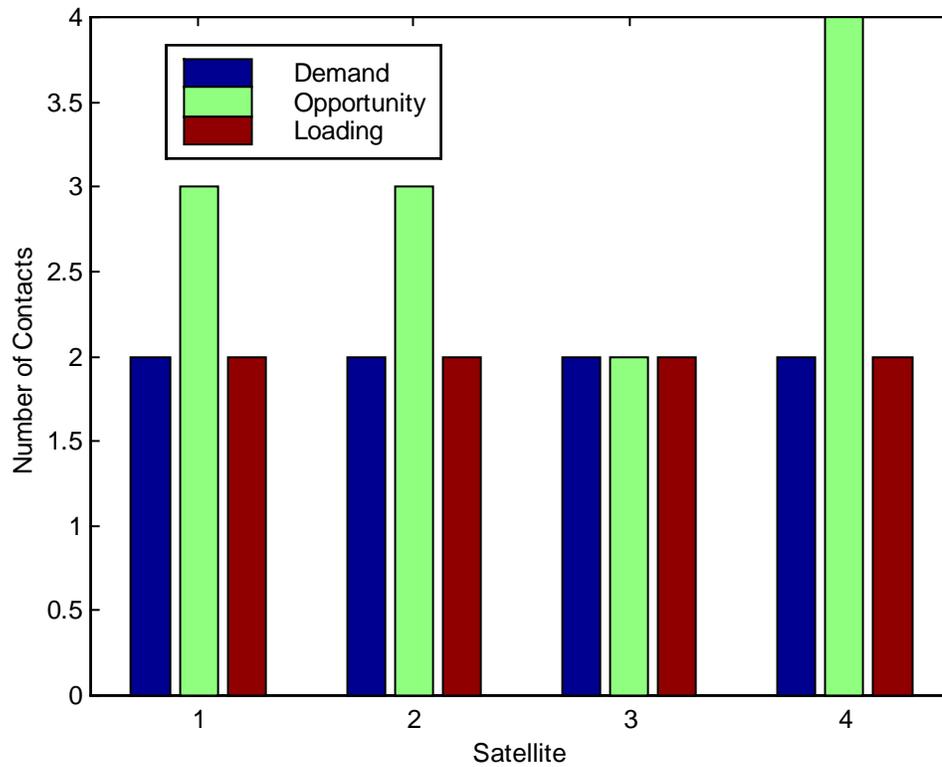


Figure 5.2.2 Total Demand, Opportunity and Loading Count per Satellite

Opportunity/Demand Count Ratio = 1.5

Schedule/Opportunity Count Ratio = 0.6667

Schedule/Demand Count Ratio = 1.0

Margin of 1 to 2 contacts until saturation

### 5.2.3 Total Demand, Opportunity and Loading Time per satellite

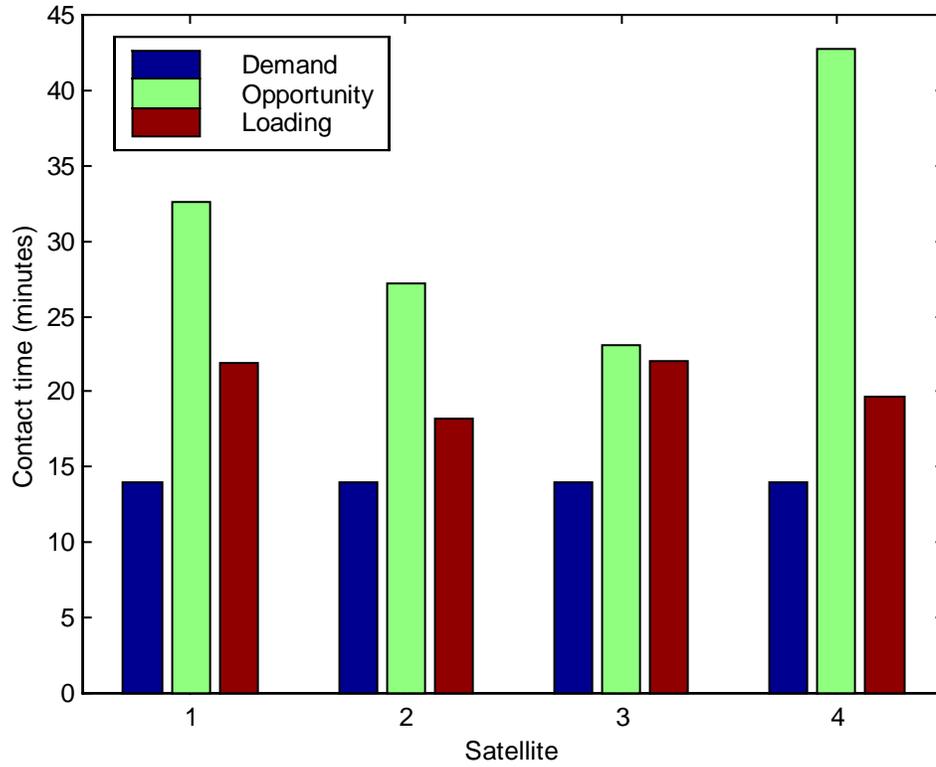


Figure 5.2.3 Total Demand, Opportunity and Loading Time per satellite

### 5.2.4 Site Utilization Count by Satellite

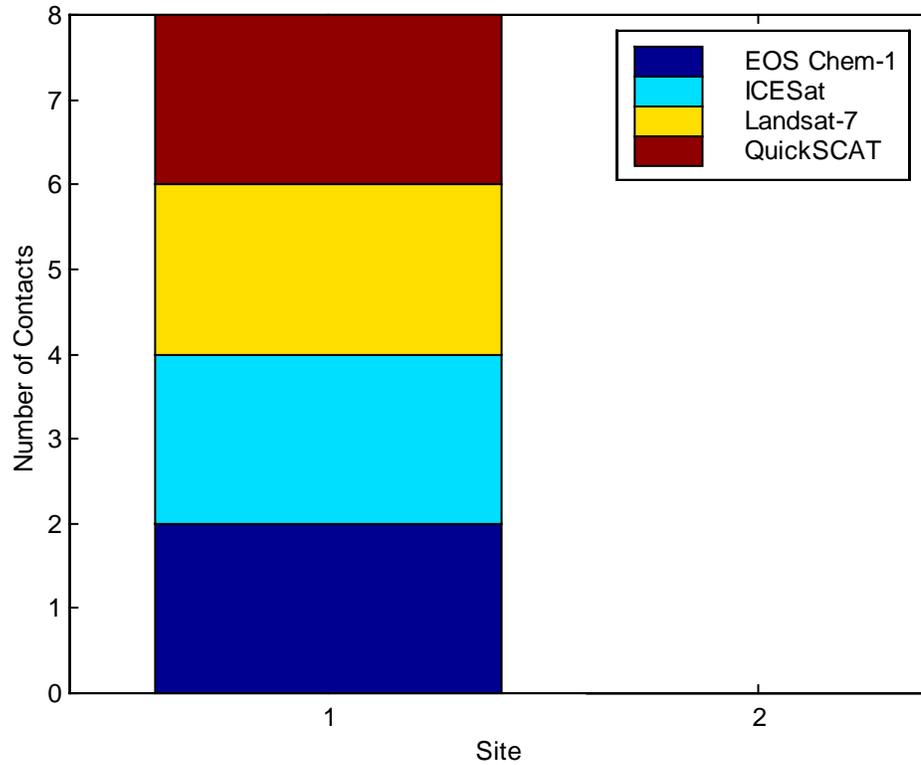


Figure 5.2.4 Site Utilization Count by Satellite

Utilization Count Percentage of Hawaiian Site:

- 25.0 % EOS Chem-1
- 25.0 % ICESat
- 25.0 % Landsat-7
- 25.0 % QuickSCAT

### 5.2.5 Site Utilization Time by Satellite

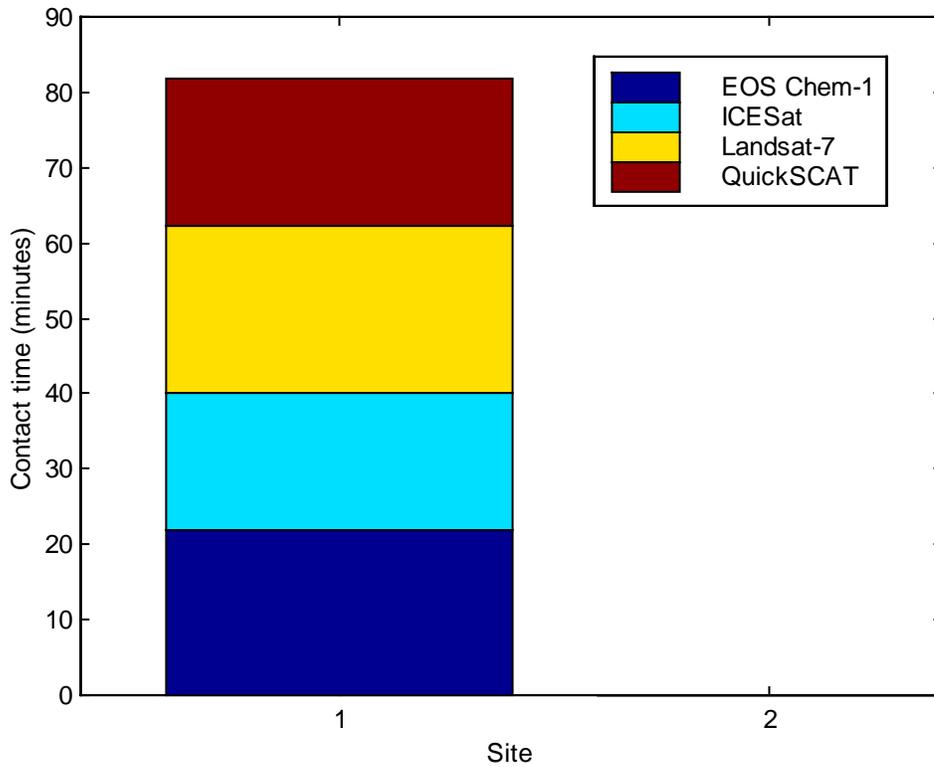


Figure 5.2.5 Site Utilization Time by Satellite

#### Utilization Time Percentage of Hawaiian Site:

- 26.79 % EOS Chem-1
- 24.01 % ICESat
- 26.89 % Landsat-7
- 22.31 % QuickSCAT

### 5.2.6 Total Site Availability Count and Total Site Loading Count

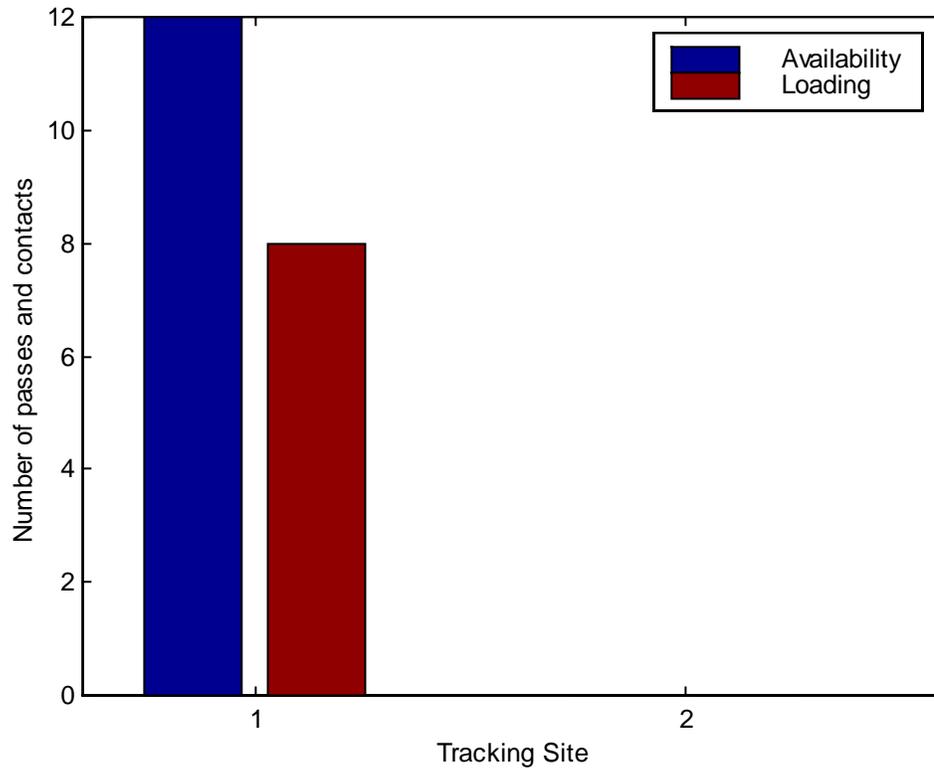


Figure 5.2.6 Total Site Availability Count and Total Site Loading Count

Site Loading / Availability Count Ratio = 0.6666

### 5.2.7 Total Site Availability Time and Total Site Loading Time

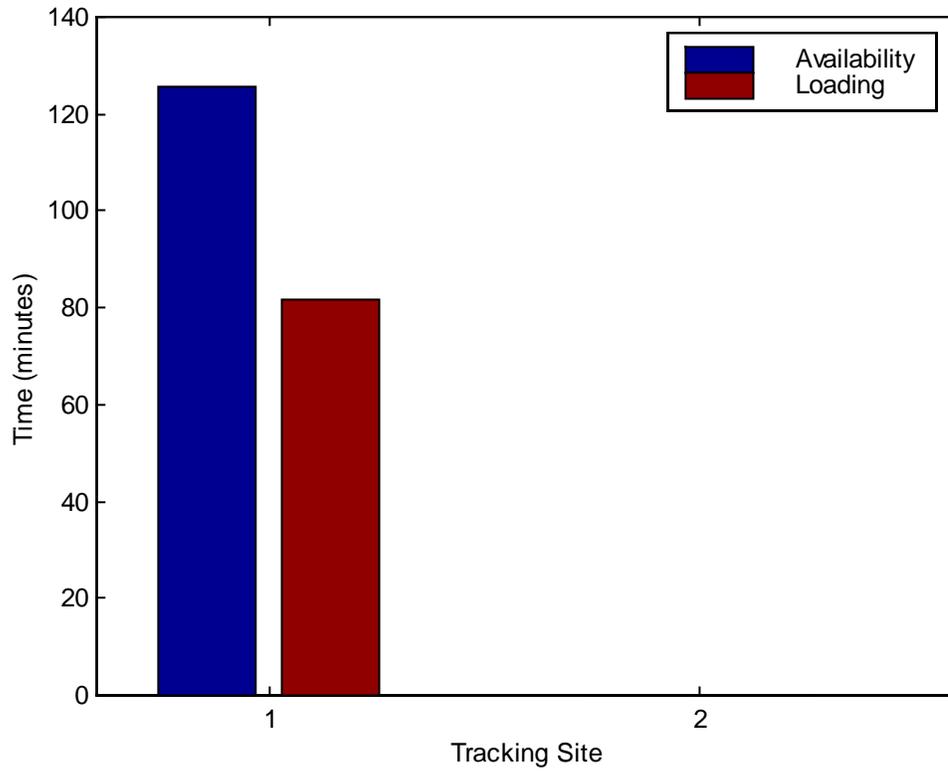


Figure 5.2.7 Total Site Availability Time and Total Site Loading Time

### 5.3 Scheduler Results for Case 1 – 4 Satellites, 1 Tracking Station; Contact Requirements = 4

#### 5.3.1 Successfully Scheduled Contacts

Table 5.3.1 Scheduler Results

Satellite	Site	Start Relative Min	Stop Relative Min	Duration (min)
EOS Chem-1	Hawaii	0	10.919002	10.919
QuickSCAT	Hawaii	241.5013918	252.5013938	11.000
QuickSCAT	Hawaii	342.6435354	351.2885325	8.645
Landsat-7	Hawaii	494.3898719	505.3898739	11.000
ICESat	Hawaii	618.9351837	629.148182	10.213
EOS Chem-1	Hawaii	731.443968	742.44397	11.000
QuickSCAT	Hawaii	910.5782401	921.0992405	10.521
QuickSCAT	Hawaii	1009.708848	1020.70885	11.000
Landsat-7	Hawaii	1226.429712	1237.429713	11.000
ICESat	Hawaii	1260.292752	1268.330746	8.038
ICESat	Hawaii	1355.794416	1364.708419	8.914
EOS Chem-1	Hawaii	1386.886752	1397.004753	10.118

Epoch date is July 1, 1999 (2451360.5 JD)

### 5.3.2 Total Demand, Opportunity and Loading Count per Satellite

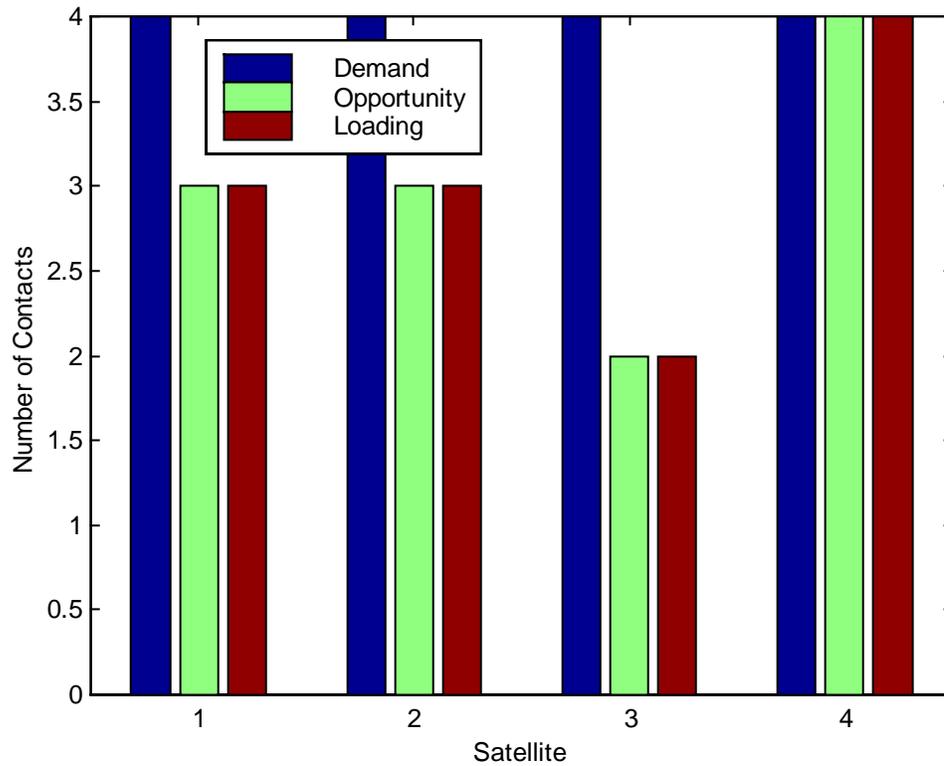


Figure 5.3.2 Total Demand, Opportunity and Loading Count per Satellite

Opportunity/Demand Count Ratio = 0.75

Schedule/Opportunity Count Ratio = 1.0

Schedule/Demand Count Ratio = 0.75

Margin of 0 contacts until saturation. The network is saturated.

### 5.3.3 Total Demand, Opportunity and Loading Time per Satellite

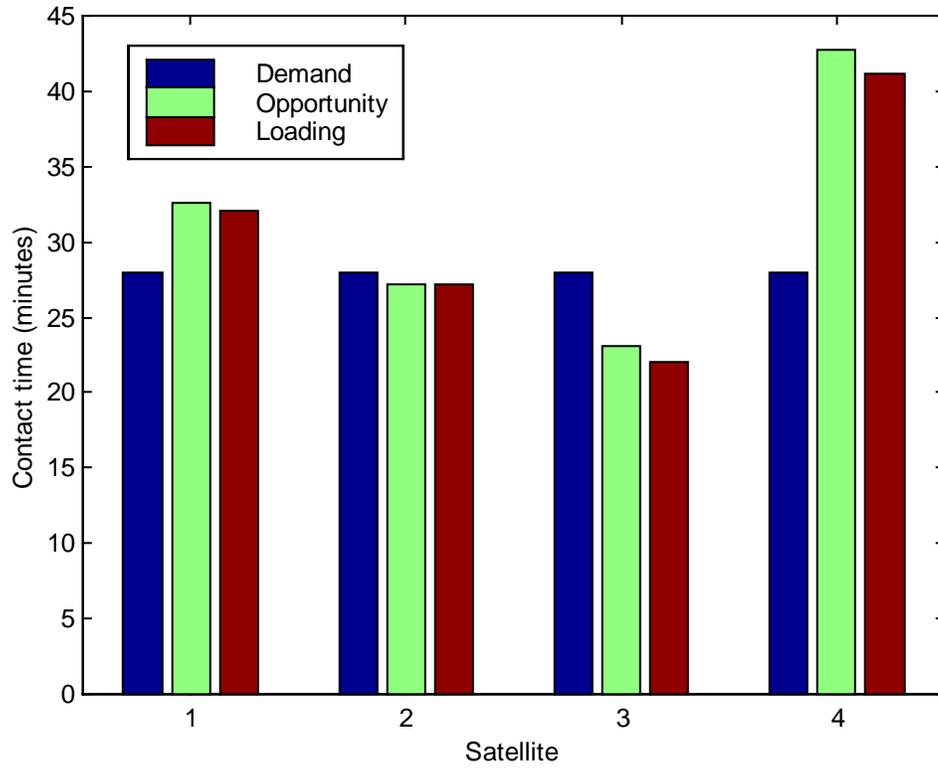


Figure 5.3.3 Total Demand, Opportunity and Loading Time per Satellite

### 5.3.4 Site Utilization Count by Satellite

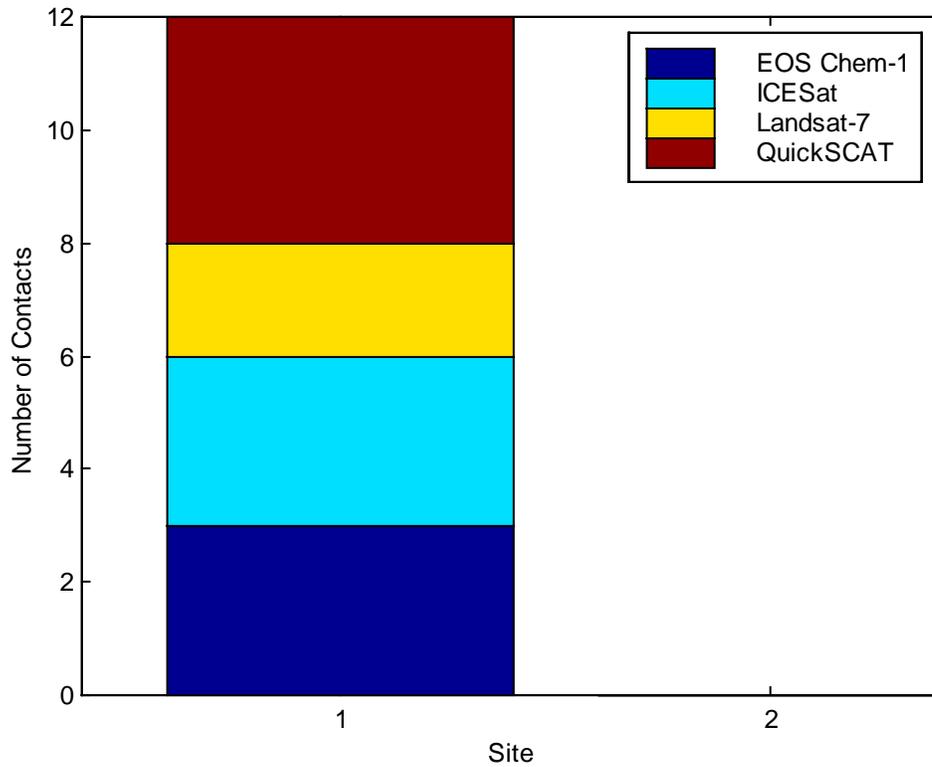


Figure 5.3.4 Site Utilization Count by Satellite

Utilization Count Percentage of Hawaiian Site:

- 25.00 % EOS Chem-1
- 25.00 % ICESat
- 16.66 % Landsat-7
- 33.33 % QuickSCAT

### 5.3.5 Site Utilization Time by Satellite

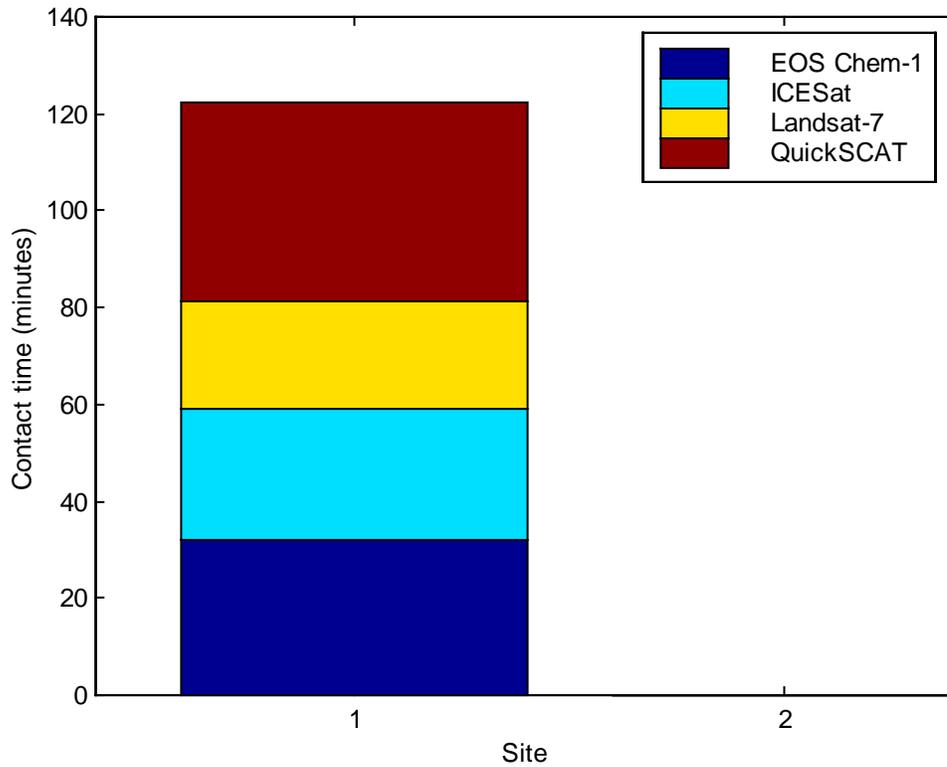


Figure 5.3.5 Site Utilization Time by Satellite

#### Utilization Time Percentage of Hawaiian Site:

- 26.18 % EOS Chem-1
- 22.20 % ICESat
- 17.98 % Landsat-7
- 33.64 % QuickSCAT

### 5.3.6 Total Site Availability Count and Total Site Loading Count

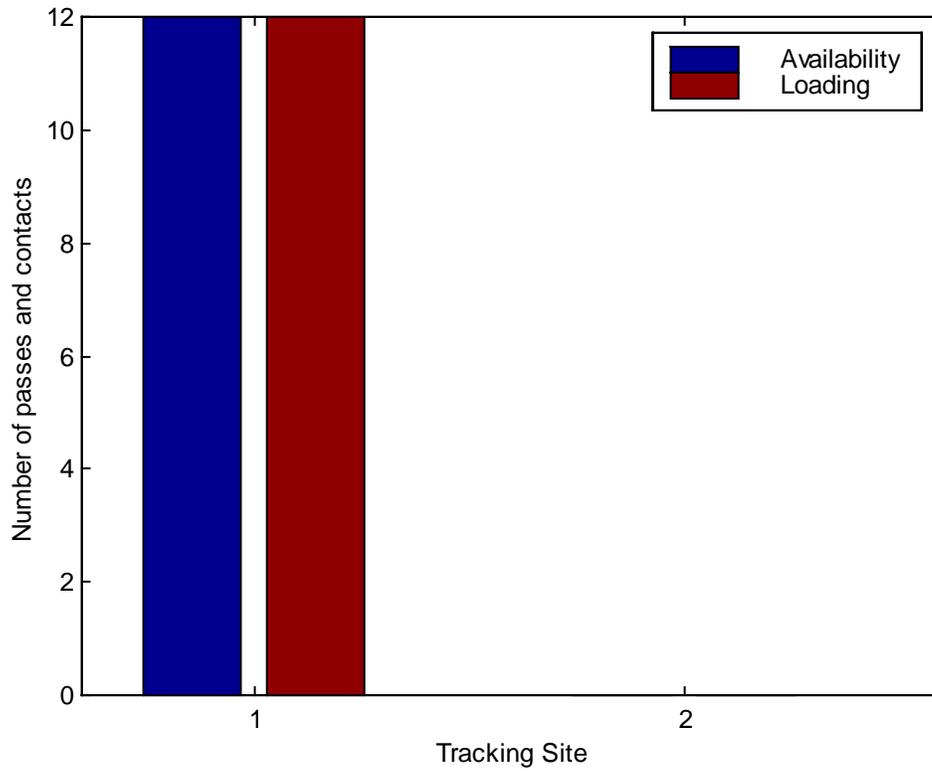


Figure 5.3.6 Total Site Availability Count and Total Site Loading Count

Site Loading / Availability Count Ratio = 1.0

### 5.3.7 Total Site Availability Time and Total Site Loading Time

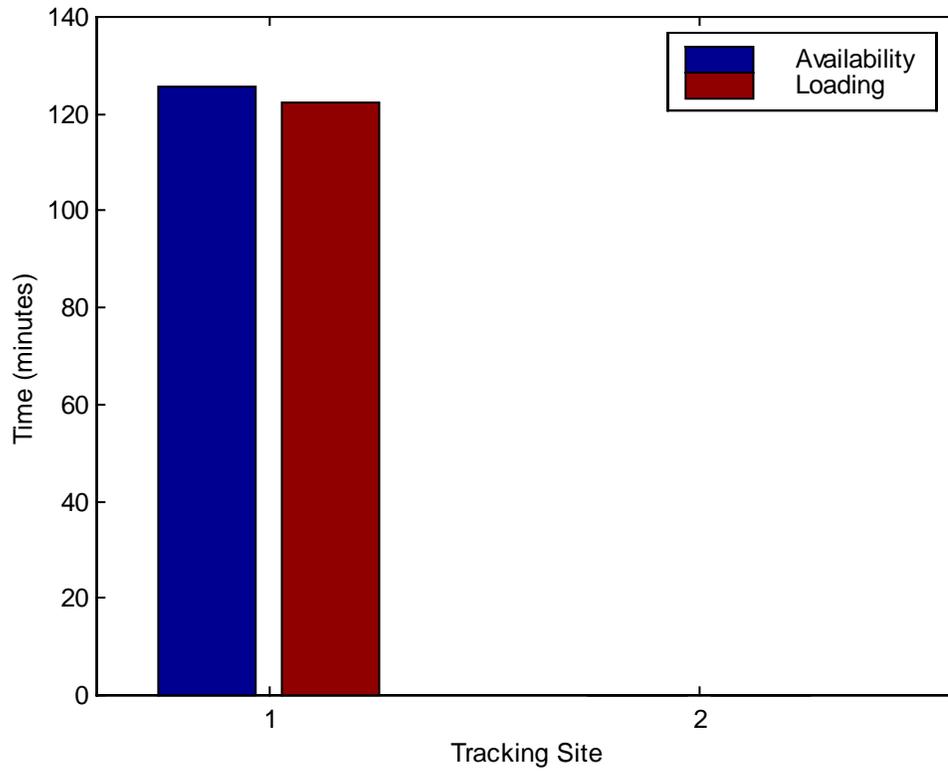


Figure 5.3.7 Total Site Availability Time and Total Site Loading Time

**5.4 Scheduler Results for Case 1 – 4 Satellites, 1 Tracking Station;  
Contact Requirements = 5**

5.4.1 Successfully Scheduled Contacts

Table 5.4.1 Scheduler Results

Satellite	Site	Start Relative Min	Stop Relative Min	Duration (min)
EOS Chem-1	Hawaii	0	10.919002	10.919
QuickSCAT	Hawaii	241.5013918	252.5013938	11.000
QuickSCAT	Hawaii	342.6435354	351.2885325	8.645
Landsat-7	Hawaii	494.3898719	505.3898739	11.000
ICESat	Hawaii	618.9351837	629.148182	10.213
EOS Chem-1	Hawaii	731.443968	742.44397	11.000
QuickSCAT	Hawaii	910.5782401	921.0992405	10.521
QuickSCAT	Hawaii	1009.708848	1020.70885	11.000
Landsat-7	Hawaii	1226.429712	1237.429713	11.000
ICESat	Hawaii	1260.292752	1268.330746	8.038
ICESat	Hawaii	1355.794416	1364.708419	8.914
EOS Chem-1	Hawaii	1386.886752	1397.004753	10.118

Epoch date is July 1, 1999 (2451360.5 JD)

### 5.4.2 Total Demand, Opportunity and Loading Count per Satellite

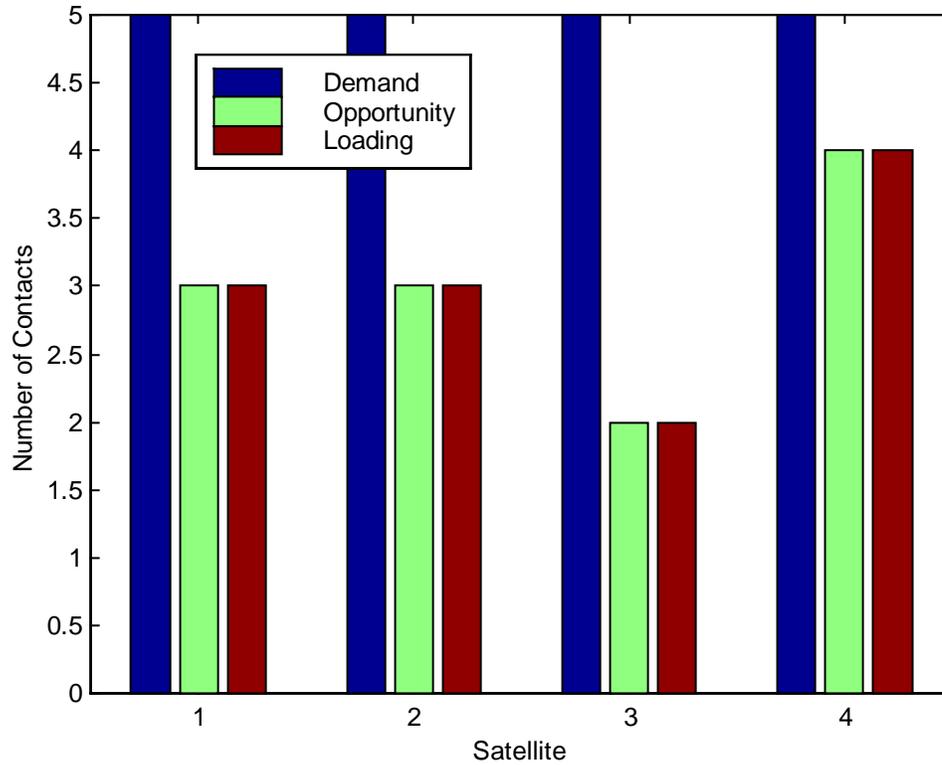


Figure 5.4.2 Total Demand, Opportunity and Loading Count per Satellite

Opportunity/Demand Count Ratio = 0.6

Schedule/Opportunity Count Ratio = 1.0

Schedule/Demand Count Ratio = 0.6

Margin of  $-2$  contacts until saturation. The network is over saturated.

### 5.4.3 Total Demand, Visibility and Loading Time per Satellite

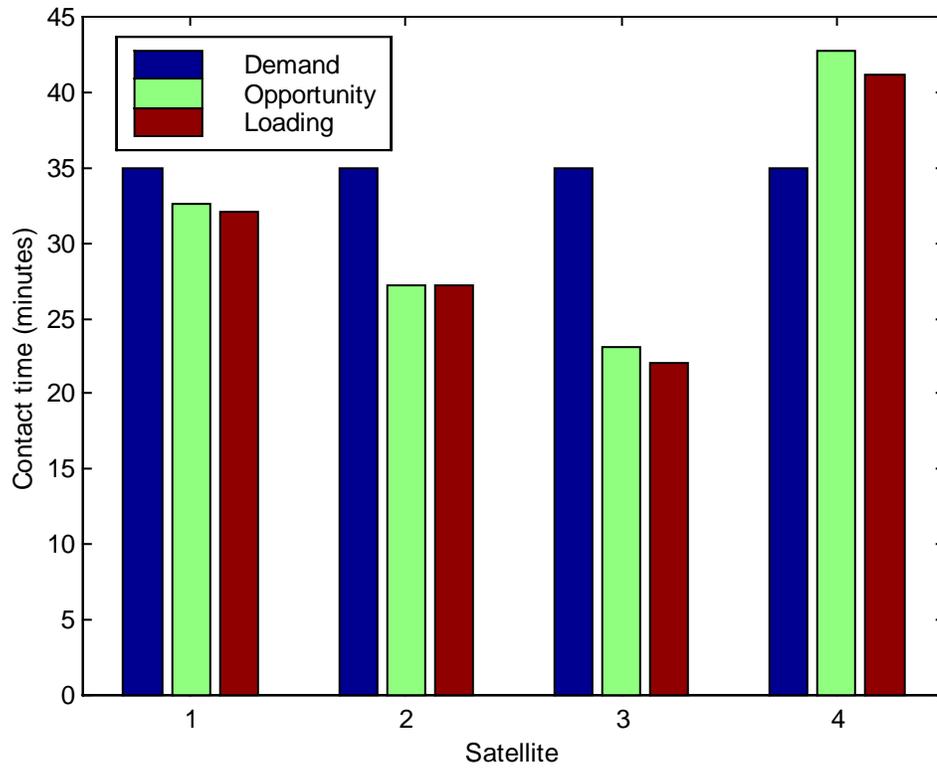


Figure 5.4.3 Total Demand, Visibility and Loading Time per Satellite

#### 5.4.4 Site Utilization Count by Satellite

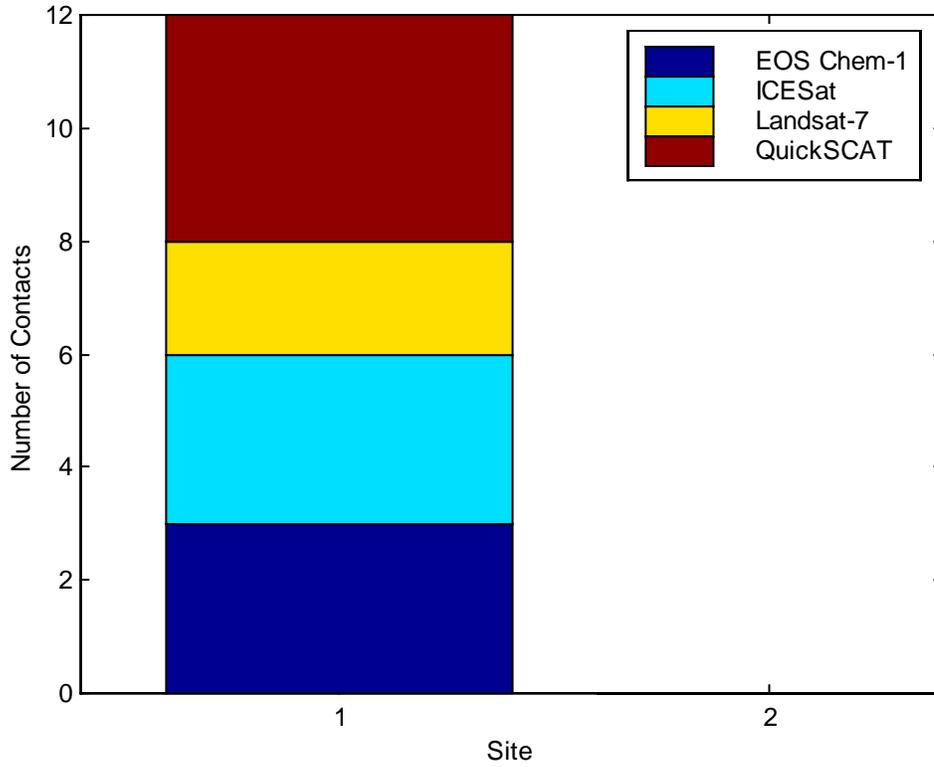


Figure 5.4.4 Site Utilization Count by Satellite

Utilization Count Percentage of Hawaiian Site:

25.00 % EOS Chem-1

25.00 % ICESat

16.66 % Landsat-7

33.33 % QuickSCAT

### 5.4.5 Site Utilization Time by Satellite

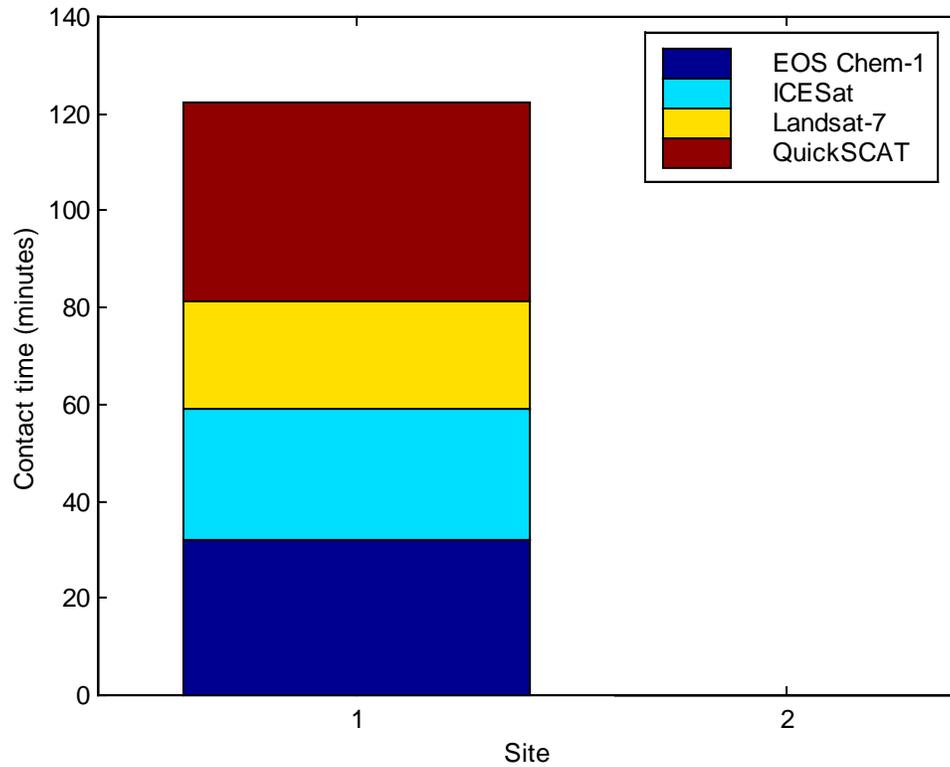


Figure 5.4.5 Site Utilization Time by Satellite

Utilization Time Percentage of Hawaiian Site:

- 26.18 % EOS Chem-1
- 22.20 % ICESat
- 17.98 % Landsat-7
- 33.64 % QuickSCAT

### 5.4.6 Total Site Availability Count and Total Site Loading Count

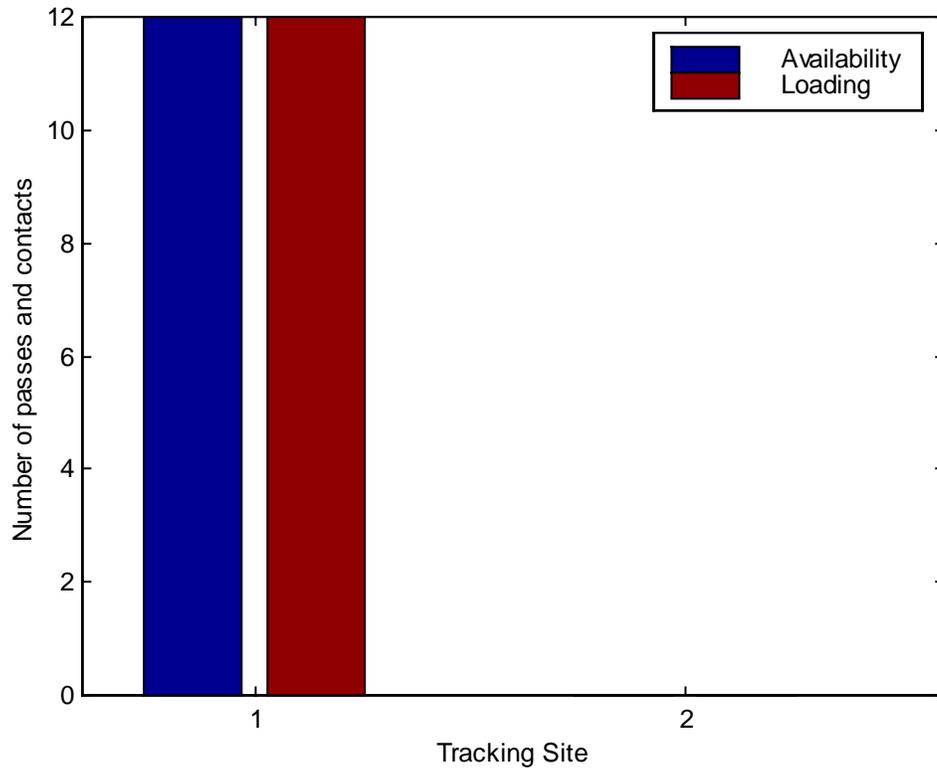


Figure 5.4.6 Total Site Availability Count and Total Site Loading Count

Site Loading / Availability Count Ratio = 1.0

### 5.4.7 Total Site Availability Time and Total Site Loading Time

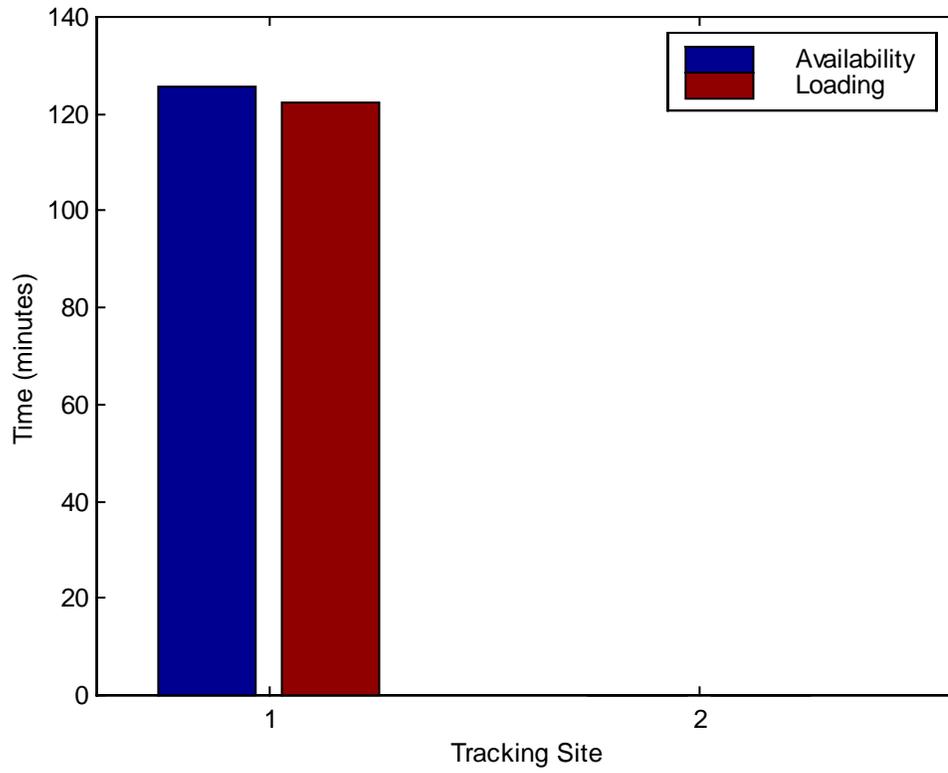


Figure 5.4.7 Total Site Availability Time and Total Site Loading Time

## 5.5 Compare Network Support Performance Ratios for Case 1

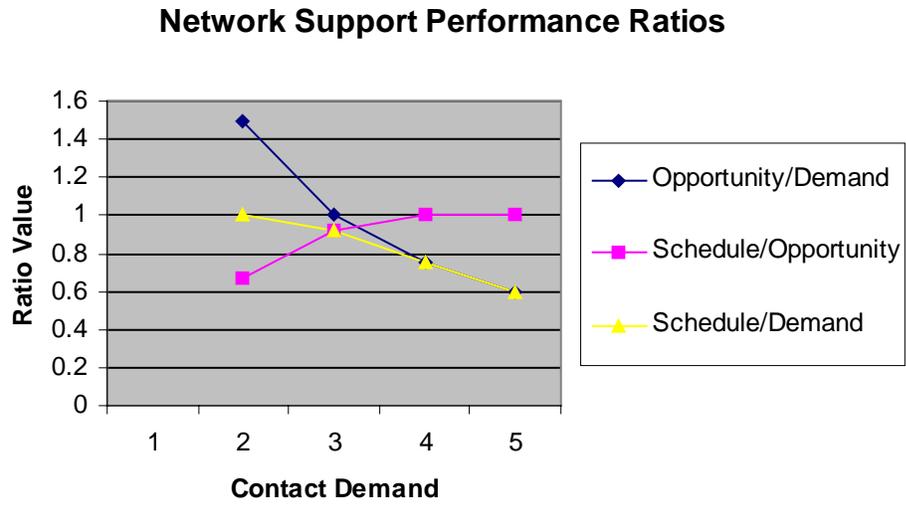


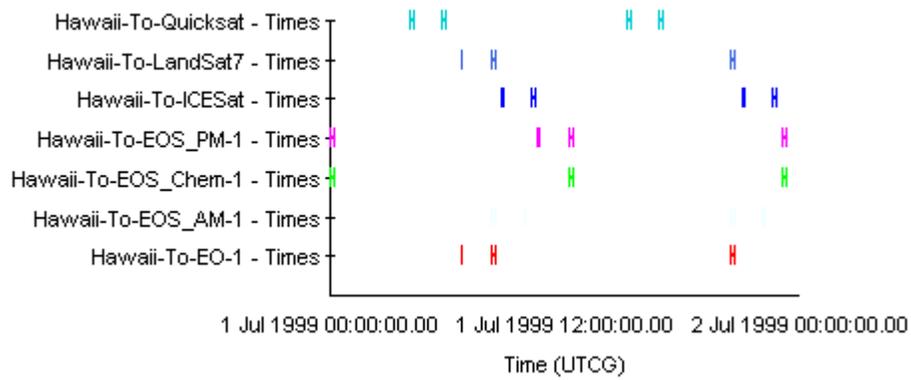
Figure 5.5 Network Support Performance Ratios

## 5.6 7 Satellites, 1 Tracking Station

### 5.6.1 Satellite Visibilities of Seven Satellites with Respect to Hawaii Tracking Site

Figure 5.6.1 Visibilities

Facility-Hawaii-To-Satellite-EO-1, Satellite-EOS\_AM-1, Satellite-EOS\_Chem-1, Satellite-EOS\_PM-1, Satellite-ICE



### 5.6.2 Total Demand, Visibility and Loading Count per Satellite

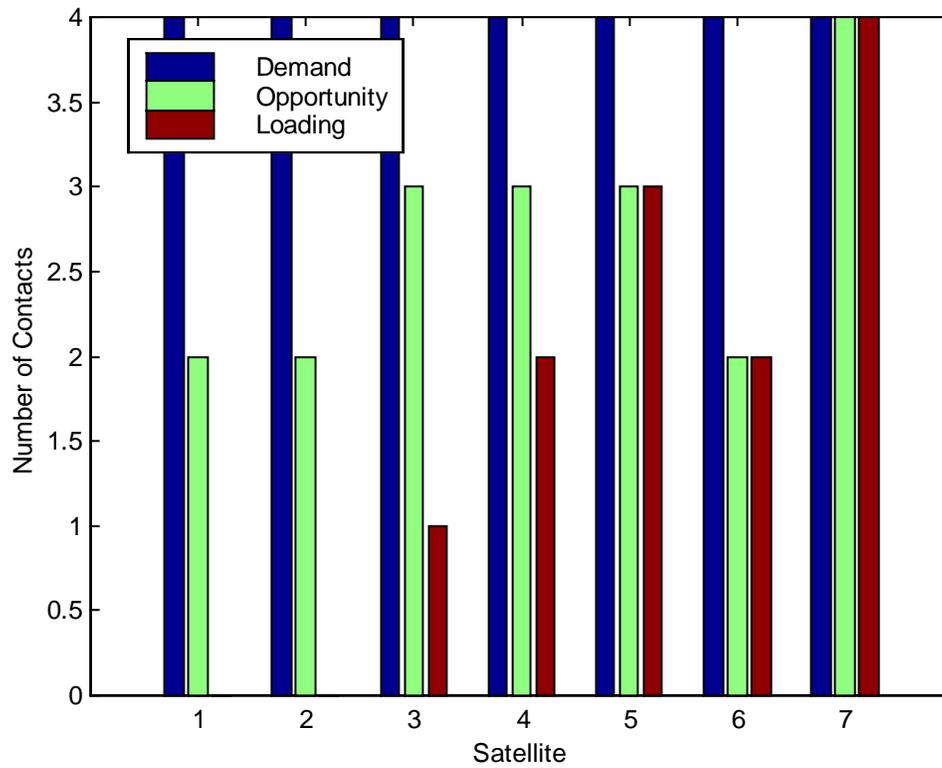


Figure 5.6.2 Total Demand, Visibility and Loading Count per Satellite

Contact Requirements = 4

Opportunity/Demand Count Ratio = 0.6786

Schedule/Opportunity Count Ratio = 0.6316

Schedule/Demand Count Ratio = 0.4286

## 5.7 Charts and Graphs for Case 2 – 3 Satellite Formation

### 5.7.1 Satellites (EOS AM-1, EO-1, Landsat-7)

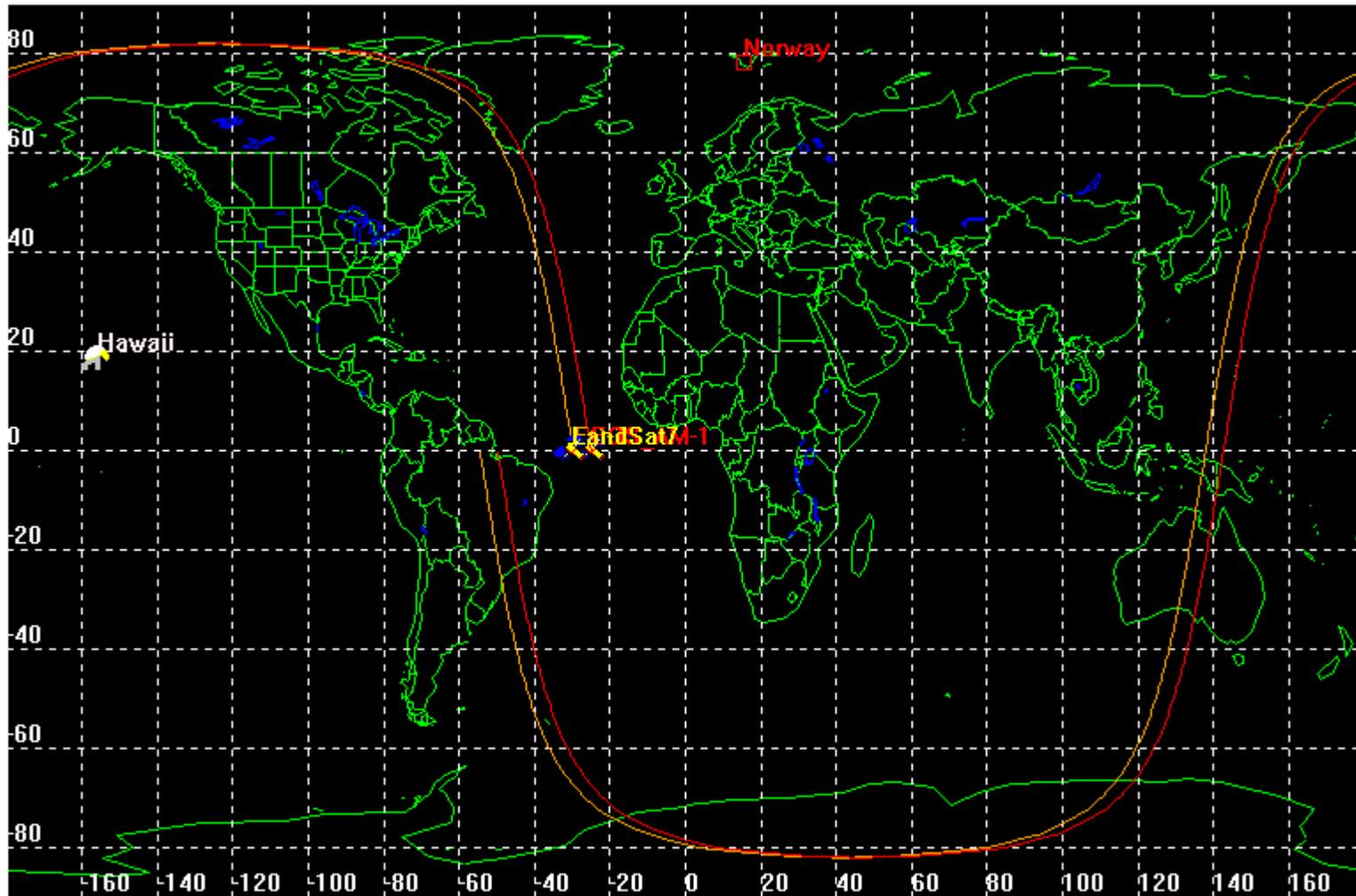


Figure 5.7.1 Satellite Paths

5.7.2 Accesses of 3 Satellites with respect to Hawaii Tracking Site

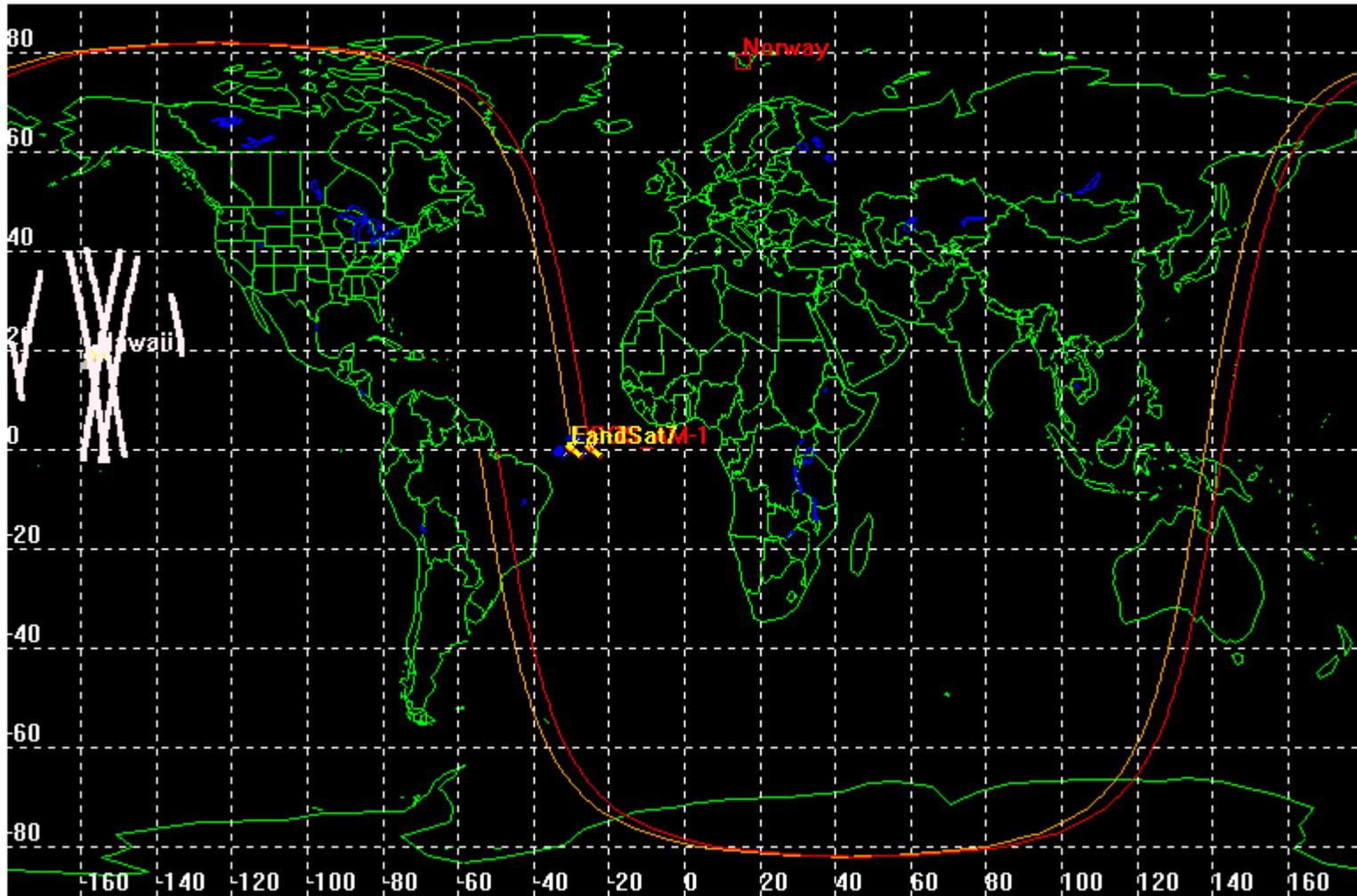


Figure 5.7.2 Accesses to Tracking Site

### 5.7.3 Satellite visibilities calculated in STK

Sites are : South Point, Hawaii  
 from file : 3sat\_1site\_formation.dat  
 Epoch date is July 1, 1999 (2451360.5 JD)

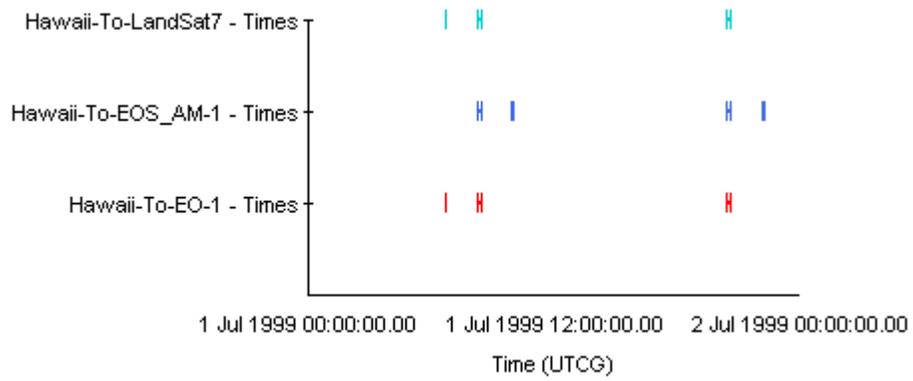
Table 5.7.3 Case 2 Visibilities

Satellite	Site	Start Relative Min	Stop Relative Min	Duration (min)
EO-1	Hawaii	401.5094396	404.3427844	2.833
Landsat-7	Hawaii	401.261472	404.5430884	3.282
Landsat-7	Hawaii	494.3898719	505.9051196	11.515
EO-1	Hawaii	494.3975041	505.9231199	11.526
EOS AM-1	Hawaii	494.6855036	506.1548158	11.469
EOS AM-1	Hawaii	596.8090079	599.9022721	3.093
Landsat-7	Hawaii	1226.429712	1237.988736	11.559
EO-1	Hawaii	1226.444112	1238.00112	11.557
EOS AM-1	Hawaii	1226.829744	1238.092128	11.262
EOS AM-1	Hawaii	1326.554208	1333.129248	6.575

### 5.7.4 Satellite Visibilities with Respect to Hawaii Tracking Site

Figure 5.7.4 Visibilities

Facility-Hawaii-To-Satellite-EO-1, Satellite-EOS\_AM-1, Satellite-LandSat7: Access Times - 19 Nov 1999 11:35



## 5.8 Scheduler Results for Case 2 - 3 Satellites in Formation; Contact Requirements = 2

Sites are : South Point, Hawaii  
from file : 3sat\_1site\_formation.dat  
Epoch date is July 1, 1999 (2451360.5 JD)

### 5.8.1 Successfully Scheduled Contacts

Table 5.8.1 Scheduler Results

Satellite	Site	Start Relative Min	Stop Relative Min	Duration (min)
Landsat-7	Hawaii	494.3898719	505.3898739	11
Landsat-7	Hawaii	1226.429712	1237.429713	11

### 5.8.2 Total Demand, Visibility and Loading Count per Satellite

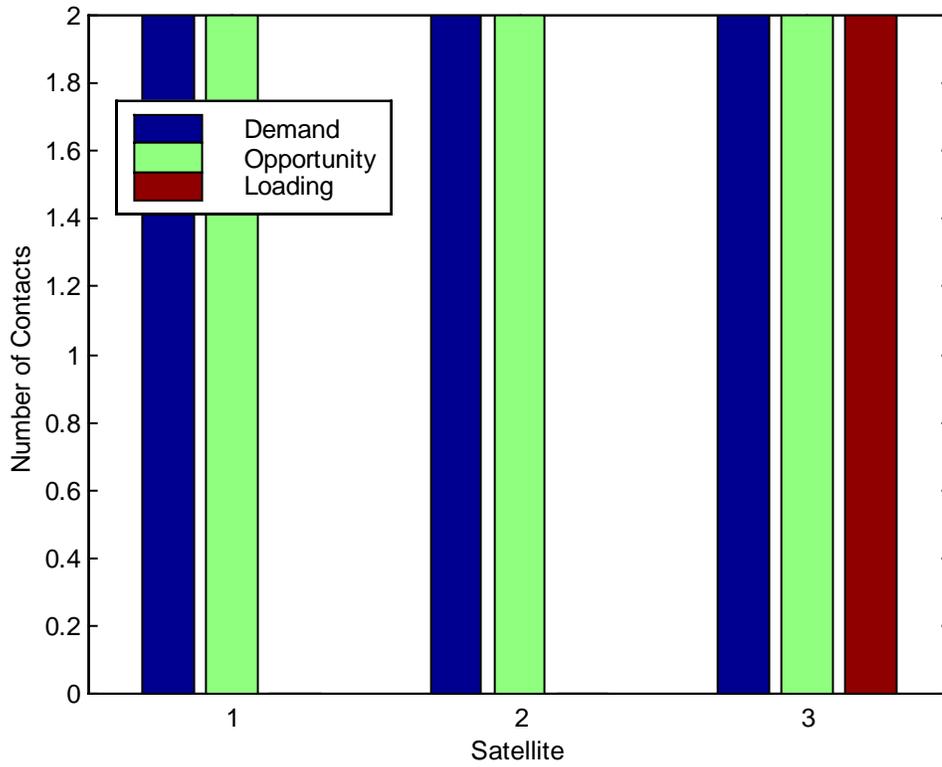


Figure 5.8.2 Total Demand, Visibility and Loading Count per Satellite

Opportunity/Demand Count Ratio = 1.0

Schedule/Opportunity Count Ratio = 0.3333

Schedule/Demand Count Ratio = 0.3333

## Chapter 6

### 6 MATLAB PROGRAMS

#### 6.1 Scheduler – Greedy Algorithm

```
%-----  
  
%filename : sched.m  
%created by seb 990615  
  
%This program "schedules" contacts using the GREEDY-ACTIVITY-SCHEDULER  
found in the book, Introduction  
%to Algorithms.  
%The scheduler is optimal in the sense of making maximum usage of the  
facility site being scheduled.  
%It does not optimize against satellites, but attempts to achieve the  
required number of contact based  
%on satellite needs.  
  
%The inputs are a file of satellite access times that are calculated  
using STK.  
%The algorithm requires that the access time be ordered by stop time  
%(Access times are visibilities with respect to the sites)  
%Inputs are sat, site, start time, stop time, duration;  
  
%Outputs are the scheduled accesses, based on the optimal algorithm, and  
performance ratios;  
%Outputs are a sat, site, contact start time, contact stop time, contact  
duration;  
  
%-----  
  
%Declare global variables;  
  
global h  
global Nsat  
global Nsite  
global access  
global answer  
  
%-----  
  
%Call function ask_user to display dialog box and get scenario info from  
%user;  
  
user_info = ask_user;  
  
Nsat = str2num(user_info{1,1});  
Nsite = str2num(user_info{2,1});  
  
%-----Initialize -----  
  
%Number of days of simulation
```

```

days = 1;

%Total number of minutes
tfinal = days * 1440;

%User Required contacts per day
rqdCntsPerDay = zeros(Nsat,1);

%Actual contacts per day
actCntsPerDay = zeros(Nsat,1);

%Satellite loading time
sattime = zeros(Nsat,1);

%Site loading time
sitetime = zeros(Nsite,1);

%Site loading per sat in minutes;
sat_site = zeros(Nsite, Nsat);

%count contact loading on Site per sat;
count_sched_site = zeros(Nsite, Nsat);

%count contact demand in minutes;
demand = zeros(Nsat,1);

%count contact demand in number or requests;
demand_count = zeros(Nsat,1);

%Satellite total pass time;
passtime = zeros(Nsat,1);

%count number of passes by Satellite;
count_pass = zeros(Nsat,1);

%count number of passes per site
count_pass_site = zeros(Nsite,1);

%number of scheduled contacts by satellite
count_sched = zeros(Nsat,1);

%Satellite pass time per site;
pass_site = zeros(Nsite, Nsat);

%Initialize tsoplast
tstoplast = 0;
schedsat = 0;
schedsite = 0;

%the last scheduled contact stop time for each site;
schedtstoplast = zeros(Nsite,1);

%Total Number of scheduled accesses
numsched = 0;

%Site setup time in minutes
setuptime = 5/1440;

```

```

%Minimum duration of contact in minutes
durmin = 7;

%Maximim duration of contact in minutes
durmax = 11;

%initial total number of eligable accesses;
total_passes = 0;

%-----

%Accesses; Need to get these from STK!
%Now import the .dat file into our matix;

getdata

%Number of Accesses
A = size(access,1);

%-----

%define required contacts per day for each satellite 1 to N;

for i = 1:Nsat

    %Call function ask_user_contacts to display dialog box and get info
    from user;

    user_req = ask_user_contacts(i);

    rqdCntsPerDay(i) = str2num(user_req{1,1});

end

%Contact demand is the number of windows that occur * the minimum req
duration
%In our case we don't have windows, so the demand and just the required
contacts * req duration;

for i = 1:Nsat

    demand(i) = rqdCntsPerDay(i) * durmin;

end

%Number of contacts demanded;

D = 0; %initialize

for i = 1:Nsat

    D = rqdCntsPerDay(i) + D;

end

```

```

d = 1;

%-----

%Loop through all accesses

for i = 1:A

    sat(i)    = access(i,1);
    site(i)   = access(i,2);
    tstart(i) = access(i,3);
    tstop(i)  = access(i,4);
    dur(i)    = access(i,5);

    %find sat and site name corresponding to current contact;
    get_sat_names;
    get_site_names;

    access_sat_names(i,:) = sat_names(sat(i),:);
    access_site_names(i,:) = site_names(site(i),:);

    %current sat and site;

    N = sat(i);
    M = site(i);

    %Only count pass/time if its an eligiable pass
    if (dur(i) >= durmin)

        %count total number of eligable accesses
        total_passes = total_passes + 1;

        %find total passtime (visibilities) of each satellite;
        passtime(N) = passtime(N) + dur(i);

        %find number of passes (visibilities) for each satellite;
        count_pass(N) = count_pass(N) + 1;

        %count number of passes per site
        count_pass_site(M) = count_pass_site(M) + 1;

        %Satellite pass time per site;
        pass_site(M,N) = pass_site(M,N) + dur(i);

    end

    %If we're trying to schedule on the same site then we need the
    % margin of setup time;
    %Otherwise there is no margin

    if schedsite == site(i)

        setupmargin = setuptime;

    else

        setupmargin = 0;

```

```

end

%Get the last contact's stop time that's been scheduled on the
%current site.

tstoplast = schedtstoplast(M);

%if we haven't reached the required contacts per day for the current %sat;
if actCntsPerDay(N) < rqdCntsPerDay(N)

    %Test for the opportunities start time to be later than the
    %previous stop time plus the setup time;
    %Also the duration must be greater than the minimum duration
    %allowed;
    %duration is in minutes;

    if (dur(i) >=durmin)

        if (tstart(i) >= tstoplast + setupmargin) %& (dur(i) >=durmin)

            %find sat and site name corresponding to current contact;
            get_sat_names;
            get_site_names;

            sched_sat_names(numsched+1,:) = sat_names(sat(i),:);
            sched_site_names(numsched+1,:) = site_names(site(i),:);

            a = 'Access ';
            b = num2str(i);
            c = ' for ';
            d = sat_names(N,:);
            e = ' was scheduled';
            abcde = [a b c d e];

            disp(abcde);

            %----- Set scheduled accesses -----

            sched(numsched+1,1) = sat(i);
            sched(numsched+1,2) = site(i);
            sched(numsched+1,3) = tstart(i);
            %sched(numsched+1,4) = tstop(i);

            %-----

            %limit the contact duration to the user set maximim duration;

            if dur(i) >= durmax

                sched(numsched+1,5) = durmax;

            else

                sched(numsched+1,5) = dur(i);

```

```

end

%new stop time is now start time + actual contact duration;
sched(numsched+1,4) = tstart(i) + sched(numsched+1,5)/1440;

%Keep track of the sats and sites that have now been
%scheduled

schedsat = sat(i);

schedsite = site(i);

%Keep track of the end of the last scheduled contact;
schedtstoplast(M) = tstop(i);

%----- Increment data counters -----

%Actual contacts per day
actCntsPerDay(N) = actCntsPerDay(N) + 1;

%count contact loading by satellite
count_sched(N) = count_sched(N) + 1;

%count contact loading on Site per sat;
count_sched_site(M,N) = count_sched_site(M,N) + 1;

%Satellite loading time (minutes);
sattime(N) = sattime(N) + sched(numsched+1,5);

%Site loading time (minutes);
sitetime(M) = sitetime(M) + sched(numsched+1,5);

%Site loading time per sat (minutes);
sat_site(M,N) = sat_site(M,N) + sched(numsched+1,5);

%increment for matix index; Total Number of scheduled
%accesses
numsched = numsched + 1;

else

%Cannot be scheduled - write to screen

a = 'Access ';
b = num2str(i);
c = ' for ';
d = sat_names(N,:);
e = ' cannot be scheduled';
abcde = [a b c d e];

disp(abcde);

end %end if (tstart(i) >= tstoplast + setupmargin)

```

```

else

    %The duration of the pass is insufficient to meet the minimum
    %required contact duration;

    %find sat name corresponding to current satellite number;
    get_sat_names;

    x = 'Pass duration for ';
    y = sat_names(N,:);
    z = ' insufficient.';

    xyz = [x y z];

    disp(xyz);

end %end if (dur(i) >=durmin)

else

    %number of required contacts for this sat has been met;

    %find sat name corresponding to current satellite number;
    get_sat_names;

    p = 'Required contacts for ';
    q = sat_names(N,:);
    r = ' already met.';

    pqr = [p q r];

    disp(pqr);

end %end if actCntsPerDay(N) < reqcntsperday(N)

end %end for i = 2:A

%Output final schedule
disp('The optimal schedule is...');
disp('(sat, site, start time, stop time, duration)');

format long;
sched
format short;

%----- Calculate ratios -----

%Number of scheduled contacts
S = size(sched,1);

%Opportunity/Demand Count Ratio
opp_demand_ratio = total_passes / D

%Schedule Performance Count Ratio
sched_ratio = S / total_passes

```

```

%Schedule/Demand Count Ratio
sched_demand_ratio = S / D

%----- Plot Results -----

%Show (min) demand, visibility time and total loading per satellite%
demand_pass_load = [demand passtime sattime];

bar(demand_pass_load);

h = legend('Demand', 'Opportunity', 'Loading');
%title('Total demand, visibility and loading time (min) per satellite');
xlabel('Satellite');
ylabel('Contact time (minutes)');

figure;

%----Show (count) demand, visibility time and total loading per satellite

count_demand_pass_load = [rqdCntsPerDay count_pass count_sched];

bar(count_demand_pass_load);

h = legend('Demand', 'Opportunity', 'Loading');
%title('Total demand, visibility and loading count per satellite');
xlabel('Satellite');
ylabel('Number of Contacts');

figure;

%----- Show loading (in minutes) on sites by satellite -----

if Nsite == 1
    new_sat_site(1,:) = sat_site;
    new_sat_site(2,:) = zeros;
else
    new_sat_site = sat_site;
end

bar(new_sat_site, 'stack');

get_legend;

%h = legend(sat_names(1,:), sat_names(2,:), sat_names(3,:),
sat_names(4,:), sat_names(5,:),...
%         sat_names(6,:), sat_names(7,));
%title('Site utilization by satellite');
xlabel('Site');
ylabel('Contact time (minutes)');

figure;

%----- Show loading (contact count) on sites by satellite -----

if Nsite == 1

```

```

    new_count_sched_site(1,:) = count_sched_site;
    new_count_sched_site(2,:) = zeros;
else
    new_count_sched_site = count_sched_site;
end

bar(new_count_sched_site, 'stack');

get_legend;

%h = legend(sat_names(1,:), sat_names(2,:), sat_names(3,:),
sat_names(4,:), sat_names(5,:),...
%         sat_names(6,:), sat_names(7,:));
%title('Site utilization by satellite');
xlabel('Site');
ylabel('Number of Contacts');

figure;

%----- Show availability on sites by satellite; -----

if Nsite == 1
    new_pass_site(1,:) = pass_site;
    new_pass_site(2,:) = zeros;
else
    new_pass_site = pass_site;
end

bar(new_pass_site, 'stack');

get_legend;

%h = legend(sat_names(1,:), sat_names(2,:), sat_names(3,:),
sat_names(4,:), sat_names(5,:),...
%         sat_names(6,:), sat_names(7,:));
%title('Site availability by satellite');
xlabel('Site');
ylabel('Availability time (minutes)');

figure;

%----- Show (min) availability, loading time per site; -----

sum_avail_min = sum(new_pass_site,2);
sum_load_min = sum(new_sat_site,2);

avail_load_min = [sum_avail_min sum_load_min];

bar(avail_load_min);

h = legend('Availability', 'Loading');
%title('Total availability, loading time (min) per site');
xlabel('Tracking Site');
ylabel('Time (minutes)');

figure;

```

```

%----- Show number of visibilities, contacts per site; -----

if Nsite == 1
    new_count_pass_site(1,:) = count_pass_site;
    new_count_pass_site(2,:) = zeros;
else
    new_count_pass_site = count_pass_site;
end

if Nsite == 1
    new_sum_load(1,:) = sum(count_sched_site,2);
    new_sum_load(2,:) = zeros;
else
    new_sum_load = sum(count_sched_site,2);
end

avail_load = [new_count_pass_site new_sum_load];

bar(avail_load);

h = legend('Availability', 'Loading');
%title('Total number of passes and contacts per site');
xlabel('Tracking Site');
ylabel('Number of passes and contacts');

```

## 6.2 Scheduler – Query User for Scenario

```
%-----  
%filename : ask_user.m  
%created by seb 092099  
  
%Dialog box  
%Ask user for number of satellites and tracking sites in the scenerio  
%-----  
  
function out = ask_user;  
  
prompt = {'Enter number of satellites in the scenario ',  
          'Enter number of sites in the scenario '};  
  
title = 'Scenerio data';  
  
lines= 1;  
  
answer = inputdlg(prompt,title,lines);  
  
out = answer;
```

### 6.3 Scheduler – Query User to Import Data File

```
%-----  
  
%filename : getdata.m  
%created by seb 090199  
  
%file to import access data;  
%Accesses are calculated in STK;  
%Ordered by stop time and arranged in the required format in Excel;  
  
%-----  
  
%Now import the .dat file into our matix;  
%The matix is read by the following command:  
% access = dlmread(answer{1,1},',,');  
  
global access;  
global answer;  
  
prompt = {'Enter the name of the .dat file to import '};  
title = 'Import STK data';  
  
lines= 1;  
  
answer = inputdlg(prompt,title,lines);  
  
access = dlmread(answer{1,1},',,');
```

## 6.4 Scheduler - Query User to Input Contact Requirements

```
%-----  
%filename : ask_user_contacts.m  
%created by seb 092099  
  
%dialog box  
%ask user for number of required contacts per day for each satellite in  
the scenerio  
  
%-----  
function out = ask_user_contacts(sat);  
prompt = {'Enter number of required contacts per day'};  
s = 'Satellite ';  
%find sat name corresponding to current satellite number;  
get_sat_names;  
n = sat_names(sat,:);  
sn = [s n];  
title = [s n];  
lines= 1;  
answer = inputdlg(prompt,title,lines);  
out = answer;
```

## Chapter 7

### 7 BIBLIOGRAPHY

- [1] Roger R. Bate, Donald D. Mueller and Jerry E. White, Fundamentals of Astrodynamics, Dover Publications, Inc, New York, NY, 58-70, 151-162, 1971.
- [2] Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, McGraw-Hill Book Company, New York, NY, 229-336, 1990.
- [3] CASEMM Analysts' Guide, Veda, Inc, 1997.
- [4] CASEMM User's Guide, Veda, Inc, 1997.
- [5] Satellite Tool Kit Training Manual, Analytical Graphics, Incorporated, 1999.
- [6] Hamilton Hagar, private communication, October 1999.

## Appendix A

### Satellite Pass Time and Altitude

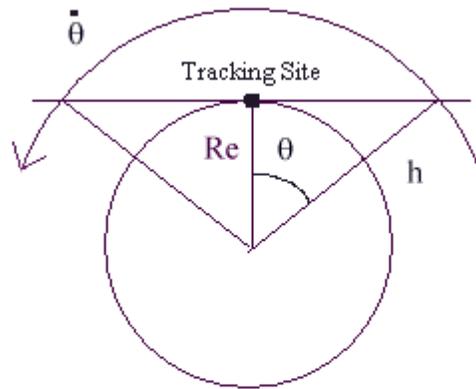


Figure A.1 Illustration of Satellite Pass Time and Altitude

$R_e$  = Radius of the Earth = 6378 km

$\mu$  = Gravitational Parameter =  $3.986 \times 10^5$

$h$  = Altitude

$T_p$  = Pass Time

$r = R_e + h$  = Radius of Circular Orbit = Semi-major Axis (for circular orbits)

Total pass angle =  $2\Theta$

$$\Theta = \dot{\Theta} \frac{T_p}{2}$$

$$\cos \Theta = \frac{R_e}{r}$$

$$\Theta = \cos^{-1}\left(\frac{R_e}{r}\right)$$

$$\dot{\Theta} = \sqrt{\frac{\mu}{r}}$$

$$T_p = \frac{2\Theta}{\dot{\Theta}}$$

$$\text{Avg } T_p = \frac{\Pi}{4} T_p$$

## Satellite Altitudes and Average Pass Time

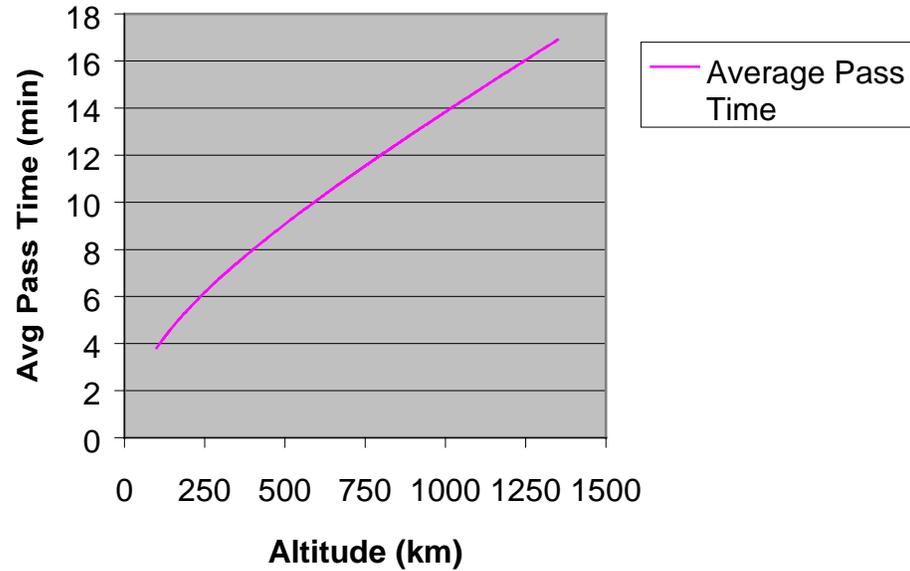


Figure A.2 Satellite Pass Time and Altitude

Minimum Contact Duration = 7 minutes

$$h_{\text{minimum contact duration}} < h < h_{2(\text{minimum contact duration})}$$

For  $h = 315$  km

$$\dot{\Theta} = 0.0011529840 \text{ rad/sec}$$

$$\Theta = 0.3080160288 \text{ rad}$$

$$\text{Avg Pass Time} = 6.993887581 \text{ min}$$

For  $h = 1020$  km

$$\dot{\Theta} = 0.0009921661 \text{ rad/sec}$$

$$\Theta \square = 0.5313425272 \text{ rad}$$

$$\text{Avg Pass Time} = 14.0203487 \text{ min}$$

Therefore for this scheduling approach, the altitudes,  $h$ , of the satellites are bound by

$$315 \text{ km} < h < 1020 \text{ km}.$$

## Appendix B

### Orbital Element Definitions

Definitions from Fundamentals of Astrodynamics by Bate, Mueller and White, [1].

1.  $a$ , semi-major axis – a constant defining the size of the conic orbit.
2.  $e$ , eccentricity – a constant defining the shape of the conic orbit.
3.  $i$ , inclination – the angle between the unit vector in the direction of the Z-axis and the angular momentum vector.
4.  $\Omega$ , right ascension of the ascending node – the angle in the Earth's equatorial plane measured eastward from the Vernal Equinox to the ascending node of the orbit.
5.  $\omega$ , argument of periapsis – the angle, in the plane of the satellite's orbit, between the ascending node and the periapsis point, measured in the direction of the satellite's motion.
6.  $M_0$ , mean anomaly at epoch – the angle defined by the product of [an "average" angular rate of the satellite's orbital motion] and [the elapsed time since the satellite passed its periapsis (position nearest the central, attracting planet)].

## **Vita**

Sarah Burrowbridge was born in Alexandria, Virginia in 1974. Her family soon moved to Charlottesville, Virginia where she graduated from Albemarle High School in 1992. She attended Virginia Polytechnic Institute and State University where she studied both Engineering and Mathematics as well as participated on the women's club lacrosse team and was a member of Kappa Delta Sorority. Throughout her academic career, Ms. Burrowbridge participated in the summer intern program at Veda Incorporated in Alexandria, Virginia. Upon receiving her BS degree in Mathematics in the fall of 1996, she joined Veda Inc full time. As a result of several acquisitions and mergers, Veda Incorporated became Veridian Corporation where Ms. Burrowbridge currently serves as a software engineer focusing on space initiatives and oracle application development. In January of 1997, she enrolled in the Interdisciplinary Applied Mathematics program at Virginia Tech's Northern Virginia Center. She received her MS degree in the fall of 1999.