Autonomous Mission Scheduling for Satellite Operations

by

Birgit M. Sauer

Submitted to the Department of Aeronautics and Astronautics
on August 14, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

When reducing the operations costs of a satellite program, planning and scheduling are a prime areas for consideration. In particular, scheduling satellite activities is repetitive, time-consuming, and non-trivial. Automating the planning and scheduling tasks can reduce operator staffing requirements, and increase the utility of the satellite. Additionally, since the main cost of an automated scheduler is its development, being able to use the scheduler for different satellite programs would lead to great cost savings. Since there is such variety in satellite programs, no realistic scheduler can be easily reused for them all. Automated schedulers can, however, be developed for “classes” of satellites that share the same fundamental characteristics. This thesis describes a scheduler for three different classes: spin stabilized science satellites; 3-axis stabilized, earth observing science satellites; and constellations of 3-axis stabilized, earth observing science satellites. Each scheduler uses a linear programming model of its mission, optimizing the value gained from the use of the instruments given a set of constraints. As a proof of concept, each scheduler is demonstrated in a case study. Finally, consideration of dynamic rescheduling in response to system failures is provided in an additional set of case studies.

Thesis Supervisor: Dr. Milton Adams
Title: Lecturer, Department of Aeronautics and Astronautics
Acknowledgments

There are many people to whom I owe a great deal of thanks: to Michael Ricard for answering silly questions without laughing; to Stephan Kolitz for his enduring sarcasm in a world that takes itself too seriously; to Milton Adams for his sane views on the treatment of graduate students; to Jim Kuchar for taking the time to cover the thesis in red ink; to Eileen Dorschner for prying obscure articles out of their dusty resting places; to Draper Laboratory for the opportunity to do this research; and to all of you who made me laugh and remember that there is more to life than the thesis due date. Thank you!

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under IR&D 710 and Contract NAS5-32007.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions herein. It is published for the exchange and stimulation of ideas.

Permission is hereby granted by the author to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

[Signature]

Birgit Sauer

August 14, 1997
# Contents

1 Automated Scheduling

1.1 Mission Planning and Scheduling ........................................... 13
1.2 Problem Statement .............................................................. 15
1.3 Literature Review .............................................................. 16
  1.3.1 Scheduling Theory ...................................................... 17
  1.3.2 Satellite Scheduling ................................................... 19
  1.3.3 Schedule Versatility ................................................... 23
  1.3.4 Standardization .......................................................... 25
1.4 Thesis Structure ................................................................. 26

2 The Satellite Scheduling Problem ........................................... 27

2.1 Problem Description ............................................................ 27
  2.1.1 Inputs ................................................................. 27
  2.1.2 Decisions .............................................................. 29
  2.1.3 Output ................................................................. 30
  2.1.4 An Example ............................................................ 30
2.2 Integer Programming Approach ............................................ 31
  2.2.1 Decision Variables .................................................... 32
  2.2.2 Objective Function ................................................... 33
  2.2.3 Constraints .............................................................. 34
5.3.2 Two Instrument Case .............................................. 87
5.3.3 Effect of Separation Angle on Effective Experiment FOV .......... 90
5.4 Case Study: EOS AM-2 ............................................. 92
  5.4.1 Case I: 5 Satellites ............................................. 94
  5.4.2 Case II: 3 Satellites ............................................. 97
  5.4.3 Case III: 5 Satellites, Different Experiment Set ................. 101

6 Uses for an Automated Scheduler .................................. 104
  6.1 Mission Scheduling .............................................. 104
  6.2 Rescheduling .................................................. 106
  6.3 Failure Analysis ............................................... 107
  6.4 Sensitivity Analysis and Design Trade-offs ....................... 109
  6.5 The Ideal Level of Automation .................................. 113
    6.5.1 Levels of Automation ...................................... 114
    6.5.2 Comparing Levels of Automation .......................... 114

7 Conclusions and Future Work ..................................... 119
  7.1 Overview .................................................... 119
  7.2 Capabilities .................................................. 120
  7.3 Limitations .................................................. 121
  7.4 Future Work .................................................. 122
  7.5 Conclusion .................................................... 123

Bibliography .......................................................... 125

A Effective Experiment FOV for Three or More Instruments .......... 129

B Tilted FOV Calculation ........................................... 134
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Growth of a combinatorial problem.</td>
<td>29</td>
</tr>
<tr>
<td>2-2</td>
<td>An optimal solution to the example problem.</td>
<td>32</td>
</tr>
<tr>
<td>3-1</td>
<td>A complete constraint matrix comprised of four decoupled sub-blocks.</td>
<td>44</td>
</tr>
<tr>
<td>3-2</td>
<td>A complete constraint matrix for a time line of four time steps that are mostly decoupled.</td>
<td>45</td>
</tr>
<tr>
<td>3-3</td>
<td>The formulation for one night time step of spectral mode.</td>
<td>57</td>
</tr>
<tr>
<td>3-4</td>
<td>Illustrative segment of the resultant week long schedule.</td>
<td>59</td>
</tr>
<tr>
<td>3-5</td>
<td>Illustrative segment of the default week long schedule assuming a 50% duty cycle on all the instruments.</td>
<td>60</td>
</tr>
<tr>
<td>4-1</td>
<td>Illustration of a satellite's field of view (FOV) and instantaneous field of view (IFOV).</td>
<td>64</td>
</tr>
<tr>
<td>4-2</td>
<td>The FOV of a nadir pointing instrument.</td>
<td>65</td>
</tr>
<tr>
<td>4-3</td>
<td>The experiment FOV is the intersection of the area all of the instrument FOVs.</td>
<td>66</td>
</tr>
<tr>
<td>4-4</td>
<td>Launch timetable for the major EOS satellites.</td>
<td>70</td>
</tr>
<tr>
<td>4-5</td>
<td>The schedule for one orbit of the EOS AM-2 sample problem.</td>
<td>75</td>
</tr>
<tr>
<td>5-1</td>
<td>Illustration of a Walker Delta constellation viewed from the north pole.</td>
<td>77</td>
</tr>
<tr>
<td>5-2</td>
<td>Three example cluster configurations.</td>
<td>78</td>
</tr>
<tr>
<td>5-3</td>
<td>The effective experiment FOV for an experiment requiring three different instruments is the intersection of the three different instrument FOVs.</td>
<td>82</td>
</tr>
<tr>
<td>5-4</td>
<td>The time integrated instrument (or experiment) FOV is a swath across the globe.</td>
<td>86</td>
</tr>
</tbody>
</table>
5-5 The actual FOV for an instrument and its linear approximation. 87
5-6 While each instrument FOV can be approximated as a flat surface, care must be taken to preserve the actual center to center separation distance. 88
5-7 Illustration of the coordinate system used to calculate the effective experiment FOV. 89
5-8 Effective experiment FOV width and area as functions of separation angle. 91
5-9 An illustration of various separation angles. 92
5-10 Effective experiment FOV width as a function of separation angle for an experiment requiring only the three tilted MODISs. 99
5-11 Schedule for one orbit of Case II. 100
5-12 Schedule for one orbit of the nominal case using the new experiments and resource amounts from Case III. 102
5-13 Schedule for one orbit of Case III with a separation angle of 5°. 103
5-14 Schedule for one orbit of Case III with a separation angle of 11°. 103
6-1 Schedule value as a function of available power as generated by the scheduler (both given as a percentage of nominal). 110
6-2 Schedule value as a function of available power as generated by theory (both given as a percentage of nominal). 112
6-3 A presumed curve of life cycle cost versus the level of automation. 113
A-1 The top and bottom bounding points of the effective experiment FOV. 130
A-2 Zero effective experiment FOV. 130
A-3 The next bounding point is the point closest to the old point in a clockwise direction. 131
A-4 The finished algorithm produces a list of bounding points and ellipses. 131
A-5 If no intersection points are found, the effective experiment FOV is the FOV of the center instrument. 132
A-6 The ellipse that connects the two boundary points is centered between the two points. 132
A-7 The ellipse that connects the two boundary points is centered either above or below the two points. 133
B-1 The FOV of a tilted instrument is longer than that of a nadir pointing instrument. .......................... 134

B-2 The tilted footprint calculated above needs to be truncated at the horizon. ............................. 135
List of Tables

2.1 Relative values of data for one time step from each instrument in the example (science points). ........................................... 30

2.2 Resource usages of each instrument in each mode in the example. .......... 31

3.1 Relative values (in science points) of each instrument in the sample TERRIERS problem. ........................................... 55

3.2 The total available amount of each resource and the usages for each instrument for each resource in the TERRIERS sample problem. ................. 55

3.3 Statistics on scheduling the TERRIERS sample problem for varying durations, using a 15 minute time step. ........................................... 57

3.4 Statistics on scheduling the TERRIERS sample problem for varying durations, using a 5 minute time step. ........................................... 58

4.1 Experiment topics, instrument requirements, and values (in units of science points). ........................................... 72

4.2 Resource usage values for each instrument. ........................................... 73

4.3 Statistics on scheduling the EOS AM-2 sample problem. ......................... 74

5.1 Down range (DR) and cross track (CT) dimensions for each of the instrument FOVs. ........................................... 93

5.2 Resource usages for each instrument. ........................................... 94

5.3 Available satellite resource amounts and their renewal rates for the nominal case and Case I. ........................................... 95

5.4 The experiment and modulator values for the nominal case, and Case I with a 5° and 11° separation angle (Me based on FOV width). ................. 96

5.5 Science value of the nominal and each Case I schedule given in science points. 96
5.6 The experiment and modulator values for the nominal case, and Case I with a 5° and 11° separation angle ($M_e$ based on FOV area). ........................................ 97

5.7 Science value of the nominal and each Case I schedule in science points ($M_e$ based on FOV area). ........................................ 97

5.8 The instrument FOV dimensions for Case II. ........................................ 98

5.9 The experiment definitions, values and modulators for the nominal case, and Case II with a 5° and 12° separation angle. ........................................ 100

5.10 Science value of each schedule (in science points) in Case II. ............... 100

5.11 The experiment definitions, values, and modulators for the nominal case, and Case III with a 5° and 11° separation angle. ........................................ 101

5.12 Science value of each schedule (in science points) in Case III. ............... 102

6.1 Science generated by the old optimal schedule and the new optimal schedule due to various failures (given as a percent of nominal). ........................................ 108

6.2 Amount of science generated, given that various failures have occurred (as a percentage of nominal). ........................................ 117
Chapter 1

Automated Scheduling

The need to reduce the cost of developing, building, and operating a satellite program has been steadily intensifying. Not only are the systems themselves becoming more complicated and therefore more expensive, but the available funding is also being cut back. The institutions that fund pure science missions, like NASA, are undergoing budget cutbacks and are driven to less expensive programs (hence the "faster, better, cheaper" paradigm). And commercial satellites have always been driven by the need to make a profit in the face of competition from both space- and ground-based systems.

As a result, satellite programs are becoming more streamlined and efficient. Most of the work to date has been in the portions of the program that incur the large, non-recurring costs: design, manufacture, and launch of the satellite. However, the costs associated with the recurring operations, such as scheduling, data archiving, and orbit maintenance, are a large part of the overall system cost and need to be examined more closely.

One of the important operations tasks is the planning and scheduling the satellite activities once the satellite is in orbit. Without a schedule, the mission cannot proceed. Moreover, since the schedule determines how efficiently the available resources are used, the quality of the schedule, affects the overall mission value. In addition, scheduling is not a trivial task. The sheer size of the scheduling problem makes it difficult to develop a good schedule, much less the best schedule.
Acknowledging this, many of the larger programs have developed and implemented automated schedulers [22, 27, 30, 37]. However, those developments have been expensive and generally not reusable since the finished scheduler does not adapt easily to other programs. Smaller programs on stricter budgets do not have the resources to develop such aids. It is, however, possible to develop a scheduler that generates optimal or near-optimal schedules that is easily adaptable to other satellites with the same type of mission. This thesis develops a proof of concept scheduler for several different types of satellite programs.

1.1 Mission Planning and Scheduling

This thesis defines a “schedule” as a time line of operations to be executed, a “feasible schedule” as a schedule that does not violate any of the constraints on the system, and an “optimal schedule” as a feasible schedule that, if executed, would generate the highest possible mission value. “Planning” refers to the entire process of creating and executing a schedule. “Scheduling” refers specifically to that part of the planning process focused on creating the schedule. The rest of this section describes the planning process in detail so this distinction will be more clear.

Planning a satellite mission involves the following tasks:

- Gathering a list of potential operations
- Deciding which operations will be executed
- Ordering the operations on a time line (scheduling)
- Verifying the schedule as feasible
- Uplinking the schedule to the satellite
- Executing the schedule
- Processing and distributing the results

For example, the “operations” to be executed for a scientific imaging mission are collecting the images or sets of images of the desired areas (referred to here as “experiments”). The
list of all the proposed experiments is compiled from all the participating scientists. Then, assuming that not all the requested experiments can be executed due to resource limitations, some subset of them is chosen. Generally speaking, this is decided by means of a peer review. For each experiment that is chosen, all the requirements and constraints are characterized in such a manner that they are intelligible to the scheduler. The scheduler then orders all these experiments on a time line (schedules them). Before the schedule is uplinked to the satellite, however, it is checked to insure all the constraints are honored. Finally, once the schedule has been executed, the resulting data are downlinked and archived for future use.

Example of a Complex Planning System

One of the most intricate planning systems is the planning system for the Hubble Space Telescope [25, 30, 44, 54, 70]. While the determination of which experiments to include in a year’s observations is left to a human review panel, the rest of the system is automated. The process works as follows. A scientist submits a proposal for an experiment in two phases. Phase I is an overview of the experiment and is submitted to the review board. The review board determines which of the tens of thousands of proposals will be accepted for the year based on value judgments such as chance of success, relevancy, and preparedness. Note that these value judgments are the reasons why this phase is not automated.

If the proposal is accepted, the scientist completes Phase II of the proposal in which he details all the requirements of the experiments. From this point on, the planning process is automated, although the human operator can override the computer at any point. The proposal is submitted in a standard form set by the Remote Proposal Submission System (RPSS), which can be accessed on the world wide web. Because the request is now in a standard format, it can be converted into a “scheduling unit” automatically. A program called Transformation (TRANS) accomplishes this, while another program, the Proposal Entry Processor (PEP), places the unit on the schedule as though it were the only unit to be scheduled that year. This serves to highlight the preferential times for each experiment and times that are in demand by many experiments. Another “meta-level” scheduler called the Criterion Autoscheduler for Long Range Planning (CASL) was added to the system to help schedule experiments that cannot be written in the pre-defined manner and to help make schedule changes once all the experiments are input into the system. Then the long
term scheduler, Science Planning Interactive Knowledge Environment (SPIKE), generates a schedule with each experiment assigned to a specific week of that year. Since the review panel approves roughly 30% more experiments than can be accommodated in a year, SPIKE must decide which requests will be honored and insure that the final schedule is feasible. The Science Planning and Scheduling System (SPSS) sorts out the schedule for each week. Once the images have been taken, they are sent to the Space Telescope Data Archive and Distribution Service (ST-DADS) for processing and storage. Data can be retrieved from the archive over the Internet using a software program called StarView.

1.2 Problem Statement

The basic satellite scheduling problem can be stated as follows: Develop a schedule that, if executed, would allow the satellite to produce the maximum amount of value while satisfying all of the physical and operator imposed constraints on the mission, the satellite, and the instruments.

These constraints come in many forms. Some are due to instrument limitations (for example, an instrument might only be able to operate for a given length of time before it overheats). Others are due to resource limitations (for example, the amount of power on-board is generally limited and must be shared by all the instruments). Yet others are due to satellite limitations (for example, the satellite is not radiation hardened enough to pass through the South Atlantic Anomaly without shutting down). There are also schedule constraints such as “the experiment needs to run at dawn any day in the month of May,” or “whenever the first experiment is run, the next three need to be run at exactly 24 hour intervals after it.” Understanding all the constraints is a feat in and of itself, much less generating an optimal schedule that honors all of them. A scheduling method that does not require a human scheduler to schedule every event individually is necessary.

The ultimate scheduler would be an automated scheduler that could be easily modified for any possible satellite. However, satellites come in many shapes and sizes with widely different missions. A completely generic scheduler is in danger of quickly becoming too large to be easily adapted to different missions. It is, however, possible to categorize satellites
and create a scheduler that is generic with respect to a class of satellites. A class is defined by the characteristics inherent in the satellite mission and design. A satellite with an earth-orbiting mission will have very different requirements than one with a sun-orbiting mission, as will science, military, and communications satellites. The goal is to group satellites that share common mission driving features. Developing a generic scheduler for one class is therefore much easier than for all possible satellite configurations.

This thesis describes a proof of concept scheduler for each of three satellite classes. Note that only scheduling is addressed, not any of the other tasks involved in planning. Moreover, only payload activities are discussed. Any satellite has two types of activities that need to be scheduled: payload activities, such as when to turn on the instruments; and bus activities, such as when to fire the station keeping thrusters. While there is no theoretical reason the bus activities cannot be scheduled with the same scheduler, they are not included here in the interest of keeping the schedulers small and easy to understand.

### 1.3 Literature Review

The Hubble Space Telescope is an example of an almost completely automated satellite program that uses entirely custom built software. This software demanded extensive development, but needs relatively little manpower to operate. Naturally, the entire planning process could also be accomplished manually with a large stack of paper. This requires almost no development, but the manpower involved in running the “scheduler” is prohibitive. Most satellite programs opt for a middle ground.

Some tasks lend themselves to automation. Data storage, for instance, is much more compact and accessible on a computer than in hardcopy. Commercial off-the-shelf data base programs can be customized for those satellite programs that do not have the time or money to develop a custom built program. Simple Gantt chart programs, like Microsoft Project [71], are also available commercially and aid the scheduling process by graphically showing the schedule as it is built.

Other pieces of software, like Draper Laboratory’s Timeliner [7], are custom built for one satellite program but were designed to be adaptable (Timeliner was originally developed
for the international space station). It is a computer language that helps streamline the planning process by having built in constructs such as “before” and “after.” This makes it easier to submit requests in the format required by the scheduler. It also makes it easier to create constraint checking programs that test the feasibility of a finished schedule.

Schedulers themselves, however, are generally not adaptable. Since the scheduling problem is complicated, many schedulers make assumptions based on the particular satellite mission that do not transfer well to other missions. The underlying theory, however, is not mission dependent.

1.3.1 Scheduling Theory

Scheduling events is by no means limited to satellite payloads. As scheduling theory is much older than the satellite industry, the formal, theoretical work has been done with regard to other systems, mainly manufacturing systems. While there are many different problems associated with a manufacturing plant, the main category of problems is “job shop scheduling.” The basic job shop model is one wherein there is a set of jobs to be completed, each consisting of a certain number of individual operations. Each operation is performed on one of a set of machines. Furthermore, there are constraints on the system. One such set of constraints might be that the machine on which each operation is to be performed and the order of the operations within a job are specified a priori. The goal is to find the schedule that minimizes the time it takes to complete all the jobs while still satisfying the constraints [41].

The problem as stated is NP-hard [20]. Generally speaking, such problems are solved to near-optimality using heuristics. There are also many variations on the basic problem, some of which are also NP-hard, others of which are solvable in polynomial time. There might be several machines of the same type, so each operation is not constrained to only one machine. The order of the operations might not be set a priori. There might be resource constraints, such as operator manpower or storage space for the finished goods, on the system. Or the particular problem might have a different objective function, such as minimizing the idle time of the machines.
The satellite scheduling problem can be cast as a job shop problem. For a scientific imaging mission there are some number of experiments (jobs) to be executed. Each experiment consists of a set of images (operations) that must be collected by one of a set of instruments (machines).

While not drastically different, the set of constraints on a typical satellite problem varies from that on the basic job shop problem outlined above. The images must be collected by a certain instrument, but the order of the images is not necessarily important. Instead, each image has a range of absolute times when it can be taken (e.g., only at night, only when the supernova is in view, etc.). There are also resource constraints on both the satellite and the instruments. The entire satellite has limited power and data downlink capacity, while each instrument has limited cryogenic coolant. Note that different satellite missions will also have different constraints.

Unlike the job shop problem, the completion time in the satellite problem is fixed. Instead, the objective is to maximize the value of all the experiment data collected in that time. Alternately, the objective may be to minimize the unused resources.

The satellite scheduling problem also incorporates aspects of two other optimization problems: the knapsack problem and the traveling salesman problem. The knapsack problem is the problem of packing a knapsack of finite volume with an assortment of objects, each with a value and a volume. The problem is to get the most valuable pack without exceeding the volume limits. While this problem is NP-complete in theory, in practice it can usually be solved in pseudo-polynomial time [35]. Similarly in the satellite scheduling problem, each observation has a value and uses a certain amount of resources. The satellite as a whole has a limit on the total amount of resources available. However, the satellite problem also has time constraints that cannot be modeled by an equivalent knapsack problem.

Given a set of cities, the traveling salesman problem finds the best route for the salesman so that he sees all of them only once. This problem is NP-complete [35] and generally solved by heuristics. A satellite also “travels” by slewing its instruments so that they can point at each “city.” The problem is, the traveling salesman problem assumes the set of cities is known, which is not necessarily true for the satellite problem, and it has no way of handling resource constraints.
The formulations described in the following chapters are loosely based on the job shop scheduling problem, both because that is the closest to the satellite scheduling problem and because it is the easier one to modify.

1.3.2 Satellite Scheduling

The following is an overview of the scheduling practices in use on various satellites. Each satellite program has its own planning system and no attempt was made to try to document them all. Instead, the goal is to give the reader a sense of the types of schedulers in use and some familiarity with the more common systems.

Manual Scheduling

Some of the first schedulers were little more than aids to the human scheduler. A method that uses rolls of butcher block paper on a conference table and a set of pencils, although not the most efficient in terms of manpower was used for a long time [18]. Computerized versions of the butcher block paper were also developed [65, 71, 72]. Colorizing schedules make them easier to understand at a glance. The amount of each resource used can also be calculated automatically instead of laboriously with a calculator. And the computer can also check all the constraints and flag those that are violated. Note that while this acts as a very convenient aid to the human scheduler, the actual scheduling is still done manually. The human scheduler places every experiment on the time line individually.

Envelope Scheduling

One of the simplest and least time consuming scheduling methods employed for a system with several users is the “envelope method.” Instead of defining the schedule down to the last detail, mission command divides the satellite resources into blocks or “envelopes” that each user is allowed to use as he sees fit for the entire working lifetime of the satellite. Each user is then responsible for his own schedule. As long as each user does not command more resources than allotted to him and does not violate any system constraints, the final schedule is nearly feasible. Any conflicts remaining are worked out by the mission command. In this
manner, mission command does not need to know the details involved in scheduling each experiment, and does not need to put a large effort into scheduling a complicated system [51].

For satellites that can only support one user at a time, the envelope method is not very useful. For instance, any satellite with several instruments that share the same optics falls in this category. A related, scheduling method which is more useful in these cases is called “coarse graining.” Each experiment is allocated a block of time, rather than blocks of resources. For that time, the experiment has full use of the satellite's resources. Coarse graining is not quite as easy a scheduling method as the envelope method since mission command needs to know enough about each experiment to be able to schedule the blocks of time. However, mission command does not need to concern itself with fine details, and the scheduling problem it is solving is still considerably smaller than that for the entire system. This scheduling method has been used quite successfully on the International Ultraviolet Explorer (IUE) for over 15 years [23, 29].

Many satellites have experiments that can be run simultaneously but that do not need a constant amount of each resource all day long. They generally run during only part of the day, week or year. Both the envelope and the coarse graining methods are very inefficient for these systems. In these cases, a variation of these methods can be used. Resource envelopes are allocated to experiments but the envelope sizes vary with time. Such a system was tested in the Earth Observing System (EOS) testbed and was a success. Moreover, it was found that it worked best when the users themselves requested certain amounts of resources at certain times. When conflicts arose, the users were informed and left to sort the schedule out for themselves. EOS will use this system for all of its satellites [39].

Note that the envelope method does not produce an optimal schedule. An optimal schedule produces the highest value possible by using the available resources in the most efficient manner. In the envelope method, any resources not used in any allotment are lost. Mission command can make adjustments and improve the schedule if it knows one user needs 20 more watts of power and a second user has them to spare, but on the whole, depending on such swaps is not practical. What is kept to a minimum with the envelope method is the effort mission command must expend on scheduling.
Heuristic Scheduling

Schedules that use resources in a more optimal manner assign an experiment only the resources required for only the time needed, allowing for a much more efficient schedule. The problem is that there are now many more possible start times, and it is much easier to accidently over subscribe resources. In other words, the problem is much more complex and correspondingly harder to solve.

Given a partial schedule and the next experiment to schedule, determining all the possible times for which that experiment could be scheduled is a painstaking and time consuming job that can be accomplished by a computer. One scheduling method is to let the human scheduler decide which experiment to schedule next, let the computer find all the possible time slots for that experiment and then let the human decide which time slot to actually schedule the experiment in. The Advanced Communication Technology Satellite (ACTS) [37] uses this type of system.

In order to choose which experiment to schedule next and to decide which time slot to schedule it in, the operator is, consciously or unconscious, employing a set of prioritizing rules. The most valuable schedule is chosen, where “value” is determined by these rules. When these rules are written down they are referred to as “heuristics.”

The schedule can be determined by a computer if these heuristics are pre-defined. Experiments are placed on the time line in an order and place determined by the heuristic(s), insuring that the schedule does not violate any of the constraints. Note that very often several heuristics will be used in a hierarchical fashion. Some of the more common heuristics used are: scheduling the most valuable experiments first (“greedy heuristic”); scheduling the most crowded time slot first; scheduling the longest experiments first; and scheduling everything as early as possible. Maestro is a heuristic based scheduler for the Japanese Experiment Module (JEM) on the Space Station [27]. The Clementine mission also used a heuristic based scheduler for their Autonomous Operations Scheduling (AOS) experiment [43]. In the latter case, the scheduler is used to generate both the payload activity and the satellite bus activity schedules.

A twist on this method is to let the heuristics determine the initial schedule while ignoring
the constraints. While the resultant schedule might not be feasible, this initial schedule can be determined quite fast. Then a repair algorithm or heuristic is used to create a feasible schedule from the initial schedule. The long term scheduler for the Hubble Space Telescope, SPIKE, uses this methodology [25]. The initial schedule is created using a min-conflict heuristic that schedules the most constrained experiments first. Repair heuristics based on experiment priority and number of conflicts are used to improve the schedule. This repair phase is ended when some pre-established level of effort has been reached. Then the schedule is “de-conflicted”. The least valuable experiments are removed until the schedule becomes feasible. Any gaps in the schedule after this de-conflicting are filled by a “best-first” heuristic.

The Space Based Visible (SBV) sensor onboard the Midcourse Space Experiment (MSX) satellite uses a scheduler called SBV Processing, Operations and Control Center (SPOCC). A list of targets to be used is supplied by the Ballistic Missile Defense Organization (BMDO). Then each target is assigned a value determined by its priority and availability (the targets are not visible all the time). The schedule is created in real time by scheduling the sensor to look at the current most valuable target [45, 49].

Optimized Scheduling

The schedules generated with heuristics, while feasible and better than most, are not typically optimal. Because of the difficult nature of the problem, there are very few schedulers that produce optimal schedules. Some schedulers, however, come close. The European Resource Satellites (ERS-1 and ERS-2) use a scheduler called PlanErs. It creates an initial feasible schedule using heuristics. Then experiments are removed and added according to another heuristic in an attempt to improve the schedule. Each schedule is saved in memory so that schedules are not duplicated. If the scheduler is allowed to run for long enough, the result will be the optimal schedule. Note that generally, the scheduler is stopped after some pre-defined amount of time and the current best schedule used even though it is not the optimal one [22].

Maestro II uses a combination of optimization techniques and heuristics to solve for the best schedule. The problem is written as an integer program, maximizing the overall resource
usage. However, in solving the problem, branching heuristics based on the value of each experiment are used to make the search space smaller [27].

The Advanced X-Ray Astrophysics Facility - Imaging (AXAF-I) determines the list of targets for the day using heuristics based on some science goal (for instance, maximize time an science targets or minimize thruster fuel consumption). Then the schedule is determined by an optimization algorithm. The final schedule is optimal with respect to that day’s target list although not necessarily with respect to the mission’s target list [32].

One of the more unique scheduling algorithms models the satellite as a robot arm over a 2-D field. As the arm makes straight passes over the field, it needs to pick up objects of varying value. Both the value and the position of each object is known a priori. In this manner, the motion of a satellite tracking spots of interest on the ground is modeled. The time it takes the satellite to turn its optics to focus on a target is modeled by the time it takes to move the robot arm from one object to another. The schedule is generated using genetic algorithms: a “population” of possible schedules is set up and mutations of those schedules result in other schedules. Good schedules are returned to the gene pool while bad schedules are removed [1].

1.3.3 Schedule Versatility

Robustness

In developing schedules, there is a trade off between optimality and robustness. Scheduling experiments on a time line in an efficient manner is one challenge. However, that schedule must be carried out by a very complex system prone to unexpected events. Experiments might take longer or use more resources than expected, one experiment might suddenly become infeasible due to a malfunction or atmospheric events, or new experiments might be added at the last minute. If this happens when the satellite is following a very efficient schedule whose events are tightly packed in time and resources, the flow of the schedule is interrupted. Perhaps the current event will not be completed. Or perhaps an event will have to be dropped altogether. Or perhaps some future event will have to be curtailed or dropped. When this happens, the schedule is no longer optimal and is said to be “broken.”
Less optimal schedules, on the other hand, tend to be more robust because there is slack in the system. Schedules created with the envelope method tend to have unused time and resources built into them. This makes them less optimal, but less likely to break since any small deviation from the expected events can be absorbed into the schedule. Note that the schedule can be broken by a deviation that requires more time and resources than are available.

Dynamic schedulers, like SPOCC, create schedules in real time. Whenever one experiment finishes, the next is started, so it is almost impossible to break the schedule. They do not, however, produce schedules that are optimal over time.

Systems such as PlanErs and those based on genetic algorithms that create schedules by continuously improving the current schedule by adding and subtracting different experiments also tend to be more robust. If an experiment is added or changed, it can be added to the pool of untried experiments and the scheduler can be allowed to continue without having to be restarted.

Rescheduling

Once a schedule breaks for any reason, a new schedule has to be generated. For some of these schedulers, like SPIKE, the entire schedule has to be redone. Others have ways to avoid regenerating the entire schedule. For the most part, these techniques capitalize on the fact that a schedule usually breaks in one place for one reason. The entire schedule doesn’t need to be regenerated, just one portion of it.

Some schedulers use the broken schedule as an initial schedule and use their scheduling heuristics to create a new, feasible schedule. Others use a local repair approach. They swap experiments in and out of the broken section of the schedule according to some pre-defined set of rules until the schedule is once again feasible. Other schedulers recognize the fact that when a schedule breaks, it is because a small set of constraints have been violated. Instead of repairing the schedule by rearranging the experiments that come right before and after the broken section, they rearrange the experiments that are important to the broken constraints [47].
One scheduler used to create schedules for automated ground-based telescopes uses "predictive error management" to aid in the rescheduling process. This scheduler uses the fact that schedules often break in predictable places. During the times when the telescope is not being used, the scheduler calculates the places and manners in which the schedule is most likely to break. Then an alternate schedule is created for that contingency. If that particular break occurs, the schedule is activated. Naturally, not every contingency can be accounted for, nor is this a good scheduler if the system has no spare CPU time [46].

Adaptation

While most satellite systems have schedulers that are custom designed, there has been some effort made to adapt schedulers from one satellite mission to another. The most notable is the long term Hubble scheduler, SPIKE. It has been used for several other observatory satellites such as the Extreme Ultra-Violet Explorer (EUVE) [66], X-Ray Timing Explorer XTE [59], and the Röntgen Satellite (ROSAT) [33]. While all three efforts have commented that SPIKE’s GUI is extremely well developed, only the ROSAT effort reports any resounding success. The other two efforts report that while SPIKE is well designed, it makes many assumptions about the system that are just not true for their systems. On the whole, the conclusion reached is that it might have been easier to develop new schedulers.

The AXAF-I scheduler is also based on SPIKE, but only in the sense that it reuses the code from the scheduling engine [32]. The new scheduler has a very different methodology and was reported a success.

1.3.4 Standardization

In line with this effort to reuse schedulers, there is also some effort in the satellite industry to standardize the interfaces of the system so that not every satellite is custom designed. SuperMOCA is a group of industry and government officials who are interested in standardizing uplink and downlink formats [2, 56]. There is also some effort being put into standardizing the human-computer interfaces [26, 38]. While the industry in general is very aware that a standard GUI would eliminate confusion and retraining from one satellite
program to the next, no consensus has been reached about that standard.

Satellite buses themselves are also slowly being standardized. Various satellite manufacturers are beginning to develop standard buses that can be used for any mission [4, 48]. Like a launch vehicle, each bus will have limited power, communications, and volume, and have a preset price. Each mission will determine the best bus for its payload.

All these modifications are in the interest of bringing down the development and operations costs for satellite systems. Better schedules make the satellite more efficient for the same cost, and standardizing the system eliminates some of the start up time and cost for the same efficiency.

1.4 Thesis Structure

While this chapter has been an overview of existing satellite planning and scheduling systems, the next chapter describes the basic satellite scheduling problem in detail. It also gives a brief introduction to the mathematical programming theory that is used to solve the scheduling problem.

Chapters 3, 4, and 5 present the formulations for the schedulers for each of the three classes of satellites: spin stabilized science satellites; 3-axis stabilized, earth observing science satellites; and constellations of the latter. The capabilities of each scheduler are demonstrated in a case study. Chapter 6 describes the utility of an automated scheduler and Chapter 7 details some conclusions and future work.
Chapter 2

The Satellite Scheduling Problem

This chapter is designed to familiarize the reader with the satellite scheduling problem. The problem goal and constraints are described here, along some background on the solution theory.

2.1 Problem Description

For a non-pointing satellite, the satellite scheduling problem is to determine which instruments should be on at each instant in time in order to produce the maximum value without violating any of the specified constraints. To be able to do so, information about the satellite, the instruments onboard, and the mission must be detailed.

2.1.1 Inputs

The inputs fall into three distinct categories, each discussed in more detail below:

- Time line information
- Data taking mode information
- Satellite and instrument information
Time Line Information  The scheduling horizon (i.e., the length of time for which the schedule is active) must be specified. Typically this is a day or a week. In contrast, a long term scheduler may be expected to produce a schedule for a year or longer. In the approach taken here, the scheduler discretizes the scheduling horizon into time steps of equal length. The length of a time step is also an input. In addition, the start date of the scheduling horizon and the parameters of the satellite’s orbit must be specified.

Each time step is characterized by the operating condition of the satellite at that time. Typical time step types include day (when the satellite is in sunlight), night (when the satellite is in eclipse), and sleep (when the satellite is shut down for safety or other operational reasons). There are many other characterizations of the operating conditions that may be important as well (for example, atmospheric conditions, such as auroras, or geometric conditions, such as having a clear line of sight to a ground station). Conditions of interest must be predetermined by mission control. Note, however, that all of these conditions are functions only of the date and the satellite’s orbit, and can be determined well in advance of the actual event. Therefore, the time line, which is a list of each time step and its type, can be generated for input either manually or by a computerized orbital propagator.

Data Taking Mode Information  Instruments can operate in different data taking modes. For instance, a science satellite might have one mode to measure over a wide spectral range but only in specific geographic areas and another mode to measure over a wide geographic range but only over select spectra. The desired percentage of time to be spent in each mode is an input and the scheduler will determine the active mode for each time step. If no instruments are on, the satellite is considered to be in sleep mode.

Note that “sleep” is used in two different contexts here: as a data taking mode and as a time step type. In both it means that the satellite is shut down, and all instruments are off. The difference is whether the condition is commanded by mission command or scheduled by the scheduler. For example, a sleep time step may be commanded as input if the physical conditions of the orbit require it (perhaps as the satellite passes through the South Atlantic Anomaly to avoid operating in a high radiation environment). Thus, a sleep time step forces the satellite to be in sleep mode. If, however, the time step type is other than sleep and the scheduler deems it optimal for all the instruments to be off during that time step (perhaps
to conserve resources for a more interesting time in the future), sleep mode is scheduled.

**Satellite and Instrument Information**  All the details about the specific satellite and the operating conditions need to be input for each satellite. Some of this information is used implicitly by mission command in determining which class the satellite fits into. In addition, there is information that must be explicitly stated. This includes the number and types of all the instruments, along with all the constraints on when an instrument can or cannot be on. The relative value of one time step’s worth of data from each instrument allows the scheduler to compute and compare the overall value of different schedules. Also important is the amount of each resource that each instruments requires, and the total amount of each resource available.

2.1.2 Decisions

The scheduler makes the decision as to which instruments are on during each time step. Moreover, the scheduler also decides which mode is active during each time step since it is possible that the same instrument may use different amounts of resources in different modes. This becomes a combinatorial problem which grows as \( T \times M \times 2^N \), where \( T \) is the number of time steps, \( M \) is the number of modes, and \( N \) is the number of instruments (see Figure 2-1).

![Figure 2-1: Growth of a combinatorial problem.](image-url)
2.1.3 Output

The output schedule specifies the mode the payload is in, and which instruments, if any, should be on during each time step in the scheduling horizon. An optimizing scheduler will create the schedule that, if run, will produce as much or more value than any other schedule while honoring all the constraints. As a by-product of the scheduling process, the scheduler also outputs time histories of the resource usages.

2.1.4 An Example

An example is outlined here to demonstrate the scheduling problem. Assume a spin stabilized science satellite in low earth orbit, requiring a schedule for a two-hour scheduling horizon. Each time step is an hour long, and because of the satellite’s orbit both are of type “day.” There are two different data taking modes (M1 and M2). For this example, there are no constraints on the modes.

This satellite has four different instruments onboard (instruments A, B, C, and D). They are all allowed to be on in either mode, however all the instruments active in one time step must use the same mode. Furthermore, instruments A, B, and C can only be on during day conditions while instrument D can only be on during night conditions.

Each instrument has a value assigned to one time step of its data (see Table 2.1). Note that this value does not change with time or mode. For clarity, one unit of value will be called one “science point.”

Table 2.1: Relative values of data for one time step from each instrument in the example (science points).

<table>
<thead>
<tr>
<th>Instrument</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>14</td>
</tr>
</tbody>
</table>

For this example, assume there is only one type of resource on board. Each instrument uses an amount that depends on the instrument and the data taking mode but not time (see Table 2.2). Initially, there are a total of 20 units onboard.
Table 2.2: Resource usages of each instrument in each mode in the example.

<table>
<thead>
<tr>
<th>Instrument</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode M1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Mode M2</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

The problem is to decide which instruments are on during each time step, and which mode they are on in.

One logical approach is to schedule the highest valued instrument first, and the the next highest, etc., until all the resources are used up. Note that care must be taken to insure that all the operating conditions are met. Instrument D is the most valuable, however, it is a night instrument and cannot be on at all during this scheduling horizon. Of the three day instruments, C has the highest value. It also has the lowest resource usage in mode M2, so one might start by assuming that instrument C will be on in mode M2 during both time steps. The next most valuable instrument is B. Remember that all the instruments must be on in the same mode at the same time, so instrument B will also be on in mode M2 for both time steps. After turning on instruments A and B in mode M2 for both time steps, there is not enough resource left over to turn instrument C on and the schedule is finished. This schedule has a value of 22 science points, which is less than the optimal value. So the logical approach does not necessarily produce an optimal schedule.

An optimal solution can be arrived at with some time and care (see Figure 2-2). The combined value of all the data is 26. During the first time step, instruments B and C are on in mode M2. During the second time step, instruments A, B, and C are on in mode M1.

What is rather apparent, however, is that some solution methodology besides intuition, trial and error, or exhaustive search is needed.

### 2.2 Integer Programming Approach

The objective in scheduling is to develop an optimal or near-optimal schedule. One approach to achieve this objective without extensive trial and error is to formulate the problem as
Mathematical programming theory is a body of knowledge containing approaches to solving constrained optimization problems. Among these approaches is linear programming which can be used to model the satellite payload scheduling problem quite naturally. The basic problem can be stated as follows: Develop a schedule that produces the maximum amount of value while satisfying all of the physical and operator imposed constraints on the mission, the satellite, and the instruments.

2.2.1 Decision Variables

The decision variables represent the choices that can be made in the course of optimizing the schedule. In this case, the natural question is whether each instrument is off or on during a given time step. A binary (written as \{0, 1\}) variable can be defined for each instrument to signify its state (off or on). Note that a variable needs to be defined for each instrument in each time step and for each data taking mode. Because all the variables are binary, this problem falls into a subset of linear programming called integer programming.
In order to use any of the linear programming theory, the rest of the problem must be linear with respect to these decision variables.

In the example of Section 2.1.4, there are 16 such variables. They can be denoted by $x_{\text{instrument},\text{mode}}(\text{timestep})$. A similar variable might also be defined for each mode, to signify whether the mode is active in the time step.

### 2.2.2 Objective Function

In order to determine quantitatively the value of a schedule, a metric must be developed. A mission's value is the benefit from the mission to the organization funding it. The problem here, then, is to define a metric which quantifies that benefit. For instance, the purpose of a science mission is to collect data. Therefore, a schedule's expected “value” is the value of the data that would be collected if the schedule were successfully executed.

The parameters employed in defining such a metric are established for each mission individually by mission command. Each instrument is assigned a number that reflects the value generated by it being on for one time interval relative to the other instruments onboard. The schedule's value is computed by summing the value of each instrument that is on over the number of intervals it is on for. Note that this metric is not intended to have a global absolute meaning. Rather, it is intended to be used as a comparison of instruments and schedules for the same satellite.

For the example of Section 2.1.4, the relative values of data from one time step of each instrument (science points) are given in Table 2.1. The objective function itself would be to maximize:

$$
(4x_{A,M1}(1) + 5x_{B,M1}(1) + 6x_{C,M1}(1) + 14x_{D,M1}(1) + 4x_{A,M2}(1) + 5x_{B,M2}(1) + \\
6x_{C,M2}(1) + 14x_{D,M2}(1) + 4x_{A,M1}(2) + \ldots)
$$

Or written in condensed form:

$$\text{maximize } \sum_{\text{timesteps}} \sum_{\text{inst}} \sum_{\text{mode}} \text{Value}_{\text{inst}} \times x_{\text{inst,mode}}(\text{time step})$$
2.2.3 Constraints

Constraints define limits on the decision variables and represent the physical and operational limitations of the satellite and its mission. Constraints can be grouped into three main categories: instrument, resource, and system constraints.

Instrument Constraints

Instrument constraints are typically specific to the suite of instruments on a given satellite. Some instruments are light sensitive and cannot be turned on during the day while others might overheat if they are left on too long. There can also be constraints on sets of instruments. For example, two instruments might interfere with each other and therefore should not be turned on at the same time. These constraints, whatever they are for the specific satellite, are expressed as linear functions of the decision variables and incorporated into the problem formulation.

In the example of Section 2.1.4, the only instrument constraints are the day/night constraints. Instruments A, B, and C can only be on during the day, while instrument D can only be on at night. In the formulation, these constraints would be expressed by setting the appropriate variables to zero:

\[
\begin{align*}
X_{A,M1}(\text{night}) &= 0 \\
X_{A,M2}(\text{night}) &= 0 \\
X_{B,M1}(\text{night}) &= 0 \\
&\vdots \\
X_{D,M1}(\text{day}) &= 0 \\
X_{D,M2}(\text{day}) &= 0 \\
\end{align*}
\]

Or in condensed form:

\[
x_{\text{instrument}}(\text{timestep}) = 0 \quad \forall \text{ time steps when each instrument cannot be on}
\]
Resource Constraints

Resources are quantities that are used during a mission, and resource constraints are limits on the amount of each resource that is used. These constraints can be further broken down into two types: rate limited and volume limited. A rate limited resource is a resource for which there is no limit to the total amount used over the lifetime of the mission, but for which there is a maximum allowable usage rate. An example is CPU, which is a measure of the available computational effort. At any given instant the payload can use at most the maximum amount of CPU available, but it can use that amount during every instant in the schedule.

A volume limited resource is a resource for which there is a finite supply but for which there is no specified limit on its rate of use. An example is money. It can be spent all at once or gradually. Volume limited resources may be renewable. The quantity is then replenished at specified intervals. An example is money doled out in a monthly allowance.

Resources can be both rate and volume limited, for example, power. Power can be drawn from the battery at no more than the maximum rate, but there is also a limit on the total amount of power available in the batteries.

There is only one resource in the example of Section 2.1.4, and it is volume limited. The constraint can be written as follows:

\[3x_{A,M1}(1) + 5x_{A,M2}(1) + 4x_{B,M1}(1) + 5x_{B,M2}(1) + 5x_{C,M1}(1) + 3x_{C,M2}(1) + 9x_{D,M1}(1) + 10x_{D,M2}(1) + 3x_{A,M1}(2) + \ldots \leq 20\]

Or in condensed form:

\[\sum_{timestep} \sum_{mode} \sum_{inst} ResourceUsage_{inst,mode} \times x_{inst,mode}(timestep) \leq ResourceAvailable\]

System Constraints

System constraints are conditions imposed on the schedule due to the nature of the mission or the design of the satellite. Note that the conditions that define the satellite’s class
are system constraints. Also included in this category are all the other constraints that are neither resource nor instrument constraints but must still be explicitly stated in the formulation. An example from the problem in Section 2.1.4 is that all the instruments that are on at the same time have to be operating in the same mode.

2.2.4 The Solution

Once the specific problem has been formulated as an integer program, it can be solved by the standard integer programming solution algorithms. The reader is referred to any operations research text book for more details on the solution techniques.

The next three chapters describe schedulers for three different classes of satellites. All three formulations are based on the problem description and solution techniques described here, although each class requires specialized constructs that will be defined in each chapter.
Chapter 3

Scheduling Spin Stabilized Science Satellites

For a spin stabilized satellite without a despun instrument platform, there is no practical way to only image a specific target. The entire satellite is spinning much too quickly for the instruments to be turned on just when a target area is in view. Instead, they are left on for the entire time that the satellite can see the target area. Data is taken for the entire $360^\circ$ of each rotation of the satellite, and filtered later to extract the data on the target area of interest. The relative simplicity of the scheduling problem for this class of satellites makes it a good starting point for developing automated satellite schedulers.

This chapter presents the formulation of a scheduler for the class of spin stabilized science satellites. The class definitions, the assumptions made in developing the model, including the class definition, and the integer programming formulation are presented. The scheduler is then demonstrated in a case study.

3.1 Modeling Assumptions

It is usually the case that certain assumptions must be made in order to model an optimization problem associated with any real system. Presented here are the main assumptions
along with the definitions of terms used to model the instrument scheduling problem. Note that the entire model is linear in the decision variables and all variables are deterministic.

The scheduling horizon is subdivided into discrete time steps of equal length. Note that this implies that all instruments must be on for a duration that is an integer multiple of this time step length. For this formulation, the time step length is bounded from below by the minimum time any instrument is allowed to be on and from above by the minimum time the satellite’s orbit remains in any one condition (e.g., day or night). A smaller time step produces a higher resolution schedule.

Any resource that is both rate and volume limited can be modeled as two separate resources without any loss of generality. The total amount of a rate limited resource may vary with time step type (e.g., there might be less power available during the night than during the day) but does not vary with mode. The amount of a rate limited resource used by each instrument per time step varies by both mode and step type.

The total amount of volume limited resource available does not vary with either time or mode. The amount of a volume limited resource used by an instrument can vary by mode. Renewal intervals are assumed to be constant and independent of both time and mode. Resources are assumed to regain their full original value at the end of every renewal interval.

The value of the data generated by an instrument in one time step is assumed to be dependent on only the instrument type, not mode or time step type. Also, no attempt was made to try and schedule instruments to be on during consecutive time steps in this model.

### 3.2 Class Description

Each satellite class has certain characteristics that also need to be modeled. While the above assumptions hold true for all the classes presented in this thesis, the following assumptions are only true of this class.

The main difference between this class and any of the other three presented here is that the satellite is spin stabilized with no despun platform, so all the instruments are rotating too fast to be pointed. During the time steps for which the instruments are scheduled to be
on, they remain on continuously and the data is filtered later to only include that for the target areas. The decision variables in question therefore determine whether an instrument is on in a particular mode during a specific time step.

Although all the instruments can function in one of several different data taking modes, all of the instruments that are on during a given time step are in the same data taking mode. Furthermore, at least one instrument must be on for a mode to be active. The exception to this is sleep mode, which is defined to be when all instruments are off.

3.3 Mathematical Programming Formulation

The formulation consists of the decision variables, the objective function and the constraints written as linear functions of the decision variables. Several different formulations were evaluated in the process of developing a scheduler that both models the satellite well and executes quickly and these are presented here. Note that all the formulations as written herein assume a deterministic model of the satellite and instruments.

3.3.1 Basic Approach (Formulation I)

This initial formulation is the most straightforward and easiest to explain. Since the later formulations are derived from this and use the same basic notation, it is presented to give the reader a better understanding of the problem.

Notation

Indices:

- time step: \( \{ t \in (1, T) \text{ where } T \text{ is the total number of time steps} \} \)
- mode: \( \{ m \in (1, M) \text{ where } M \text{ is the total number of modes} \} \)
- instrument: \( \{ i \in (1, I) \text{ where } I \text{ is the total number of instruments} \} \)
- resource: \( \{ r \in (1, R) \text{ where } R \text{ is the total number of resources} \} \)
Constants:

- $RI_r(t)$ – amount of rate limited resource $r$ available in time step $t$
- $RT_r$ – amount of volume limited resource $r$ available in one renewal period
- $RC_r$ – renewal period of resource $r$
- $R_{imr}$ – amount of resource $r$ used by instrument $i$ in mode $m$
- $V_i$ – value of instrument $i$ being on for one time step
- $MR_{m\bar{m}}$ – ratio of time spent in mode $m$ to time spent in mode $\bar{m}$
- $\delta$ – a number approaching zero used to insure some constraints remain feasible
- $M$ – a large number to facilitate the linearization of the constraints

Decision Variables:

- $x_i(t) = \begin{cases} 1 & \text{if instrument } i \text{ is on during time step } t \\ 0 & \text{otherwise} \end{cases}$
- $y_m(t) = \begin{cases} 1 & \text{if mode } m \text{ is active during time step } t \\ 0 & \text{otherwise} \end{cases}$
- $z_{im}(t) = \begin{cases} 1 & \text{if instrument } i \text{ is on in mode } m \text{ during time step } t \\ 0 & \text{otherwise} \end{cases}$

The variables $x$ and $y$ are, of course, dependent on $z$. In fact, $x_i(t) \times y_m(t) = z_{im}(t)$. They were included because they make the formulation much cleaner and easier to understand.

Sleep mode is somewhat different from the other data taking modes in that all the instruments must be off. When sleep mode is active, all $x$ and $z$ are forced to zero, but $y_{\text{sleep}}(t)$ is one. For any of the other modes to be active, at least one instrument must be on in that mode.
Objective Function

\[
\text{maximize } \sum_t \sum_i V_i \times x_i(t)
\]

The goal of the scheduler is to find the schedule with the highest value. So the objective is to maximize the sum of the values of all the instruments that are on during the entire schedule. Any instrument not on during time step \( t \) will have \( x_i(t) = 0 \) and will therefore not contribute to the schedule value.

Constraints

\[
x_i(t) = 0 \quad \forall \text{ time steps } t \text{ when instrument } i \text{ cannot be on}
\]

\[
z_{im}(t) = 0 \quad \forall \text{ modes } m \text{ when instrument } i \text{ cannot be on during time step } t
\]

\[
\sum_m y_m(t) = 1 \quad \forall \ t
\]

\[
x_i(t) = \sum_m z_{im}(t) \quad \forall \ t, i
\]

\[
M \times y_m(t) \geq \sum_i z_{im}(t) \quad \forall \ t, m \text{ (except sleep)}
\]

\[
y_m(t) \leq \sum_i z_{im}(t) \quad \forall \ t, m \text{ (except sleep)}
\]

\[
|\sum_t x_i(t) - (\sum_t x_i(t))| \leq \delta_1 \quad \forall \ (i, \tilde{i}) \text{ pairs}
\]

\[
|\sum_t y_m(t) - MR_{m\tilde{m}} \times (\sum_t y_{\tilde{m}}(t))| \leq \delta_2 \quad \forall \ (m, \tilde{m}) \text{ pairs}
\]

\[
\sum_i \sum_m R_{imr} \times z_{im}(t) \leq RI_r(t) \quad \forall \ \text{rate limited } r, t
\]

\[
\sum_t \sum_i \sum_m R_{imr} \times z_{im}(t) \leq RT_r \quad \forall \ \text{volume limited } r, RC_r
\]

\( x_i(t), y_m(t), z_{im}(t) \) are all \( \{0, 1\} \) variables
Constraint (3.1) takes into account that certain instruments cannot be on during certain times. For instance, turning a visible frequency instrument designed for night use only on during the day would overload the photo-receptors, so that instrument is constrained to be off during all time steps for which the satellite is in daylight.

In the same manner, constraint (3.2) specifies that there are some modes in which certain instruments cannot be on. For instance, an instrument needing warmth could not be on in a mode that involves releasing cryogenic coolant. Note that because of the dependency between $x$, $y$, and $z$ (see constraints (3.4), (3.5), and (3.6)), constraints (3.1) and (3.2) set not only $z_{im}(t)$ to zero, but also the corresponding $x_i(t)$ and $y_m(t)$. Also remember that when a sleep time step is specified, all the instruments are off by definition. This means that all $x$, $z$, and all $y$ except $y_{sleep}$ are zero for that time step.

Constraint (3.3) forces one and only one data taking mode to be active at once since the entire payload must be in the same mode. If no instruments are on, the payload will be in sleep mode. In this case, all the non-sleep modes are forced to be zero (inactive) by constraints (3.5) and (3.6) and sleep mode is forced by constraint (3.3).

Constraints (3.4) and (3.5) relate $x$ and $y$ to $z$. The constraint $x_i(t) \times y_m(t) = z_{im}(t)$ would be much more succinct, but is not linear. Since only one mode can be active at once (see constraint (3.3)), the sum over all the modes of $z_{im}(t)$ for a given instrument in a time step will either be zero or one. If it is one, instrument $i$ is on during that time step, otherwise it is off.

Because more than one instrument can be on at once, the dependence of $y$ on $z$ is not quite so simple. At time $t$, the number of instruments on in mode $m$ is $\sum_i z_{im}(t)$. When there are no instruments on during time step $t$ for a given mode $m$, this sum is zero, mode $m$ is inactive and $y_m(t)$ should be zero. This is insured by constraint (3.6). If there are instrument(s) on in that mode, $\sum_i z_{im}(t) > 0$ and the mode is active. In this case, constraints (3.5) and (3.6) force $y_m(t) = 1$ as long as $M$ is greater than the upper bound on $\sum_i z_{im}(t)$. Note that this means $M$ must be at least equal to the maximum number of instruments that can be on at once.

Constraint (3.6) insures that a mode $m$ is not active during a time step when there are
no instruments on in that mode. Sleep mode, when all instruments are off by definition, is obviously an exception to this and is not included in this constraint. Note the above comments on the values of the decision variables in sleep mode.

Constraint (3.7) accounts for the fact that if there are several instruments of the same kind onboard, they should all be run for equal amounts of time (so one instrument is not overused while the others stay idle). Therefore, for all pairs of like instruments $i$ and $\tilde{i}$, the total time each instrument is on should be nearly equal (the allowable difference is characterized by the small value $\delta$). Note that the constraint cannot just be written as $\sum_t x_i(t) = \sum_t x_{\tilde{i}}(t)$. If, for instance, there are two like instruments and an uneven number of time steps, satisfying this constraint becomes impossible. So the difference between the number of time steps each is on must be less than some integer $\delta_1$, where $\delta_1$ is at least the number of like instruments modulo the number of time steps. The larger the value, the looser the constraint, the easier it is to solve the problem. The absolute value allows for the fact that there is no preference as to which instrument, $i$ or $\tilde{i}$, is used slightly more often.

Constraint (3.8) is a similar constraint on the modes, except that instead of equal proportions of each mode, a ratio of $MR_{m\delta}$ is desired. Here $\delta_2$ is at least the number of modes modulo the number of time steps. Note that although sleep mode is generally excluded from this constraint, there is nothing in the formulation that precludes it from being included.

Constraint (3.9) is the rate limited resource usage constraint. For every rate limited resource, $r$, the total usage in one time step cannot exceed the total available for the given time step type. The total usage is the sum of the amounts used by all the instruments that are on (in the mode that is active). Note that not all $r$ are necessarily rate limited.

Constraint (3.10) is the volume limited resource usage constraint. For every volume limited resource, $r$, the total usage in one renewal period cannot exceed the total available. Note that a volume limited resource is not necessarily renewable, in which case the renewal period is the entire scheduling horizon, and that not all $r$ are necessarily volume limited.
Implementation

To solve this optimization problem using standard techniques, the above equations need to be expressed in matrix form for each instrument and mode, at each time step. This matrix is called the constraint matrix. The rows represent the constraints while the columns represent the decision variables (in this case, the instruments and modes). For each row, the entries correspond to the coefficients of each decision variable in that constraint.

To create the constraint matrix, it is helpful to note that the complete matrix is made up of relatively uncoupled smaller matrices, one for each time step. Each smaller matrix is created by assuming that there is only one time step in the scheduling horizon. Time steps of the same type have identical small matrices, so once the prototypes matrices for each time step type have been developed, they can be used as building blocks for the complete constraint matrix. The time line specifies their order in the complete matrix. Figure 3-1 shows a block diagram of a complete constraint matrix for a problem with four time steps, each of them a different type. Each sub-block represents one of the time steps. Note that the complete matrix is very sparse.

\[
\begin{bmatrix}
  a \\ b \\ [ c ] \\ d
\end{bmatrix}
\]

Figure 3-1: A complete constraint matrix comprised of four decoupled sub-blocks.

Of course, creating the complete matrix is this simple only when the constraints for all the time steps are completely uncoupled. This is not the case as some of the constraints are summed over more than one time step. The constraints can be divided into three types: (1) those that are active only in one time step; (2) those that are summed over the entire scheduling horizon; and (3) those that are summed over some renewal interval. By keeping careful track of the constraints, it is still possible to create the complete matrix from these simpler building blocks. Figure 3-2 shows a block diagram of a complete constraint matrix for a problem with four time step types (a - d). Each time step has constraints of all three types (1 - 3). Note that the complete matrix is still very sparse.
Figure 3-2: A complete constraint matrix for a time line of four time steps that are mostly decoupled.

\[
\begin{bmatrix}
1_a \\
1_b \\
1_c \\
1_d \\
2_a & 2_b \\
3_a & 3_b \\
2_c & 2_d \\
3_c & 3_d
\end{bmatrix}
\]

One refinement that can drastically reduce the number of decision variables in the formulation is made by noting that many of the decision variables associated with the sub matrices for each step type are constrained to be zero (see constraints (3.1) and (3.2)). If those matrices are preprocessed so that these variables are identified and left out of the formulation entirely, the total number of variables is reduced by more than half. For instance, in a satellite with 4 instruments (2 night and 2 day) and 3 modes (including sleep), there are 19 variables for each time step (12 \( z \), 3 \( y \), and 4 \( x \)). During a day or night time step, 11 of those 19 are constrained to be zero (8 \( z \), 1 \( y \), and 2 \( x \)). During a sleep time step 18 of those 19 are zero (all but \( y_{\text{sleep}} \)). Since reducing the number of variables reduces the solution space and therefore the solution time, this is a recommended step.

A sample problem (See TERRIERS case study, Section 3.4) was formulated in this manner with 7 instruments, 3 modes, and 3 step types. The problem was run on a Sparc20 using the default branch and bound integer program routines from a commercial solver, CPLEX. With a time step of 15 minutes, the computer ran out of memory after several hours of run time when it tried to solve for a scheduling horizon of more than a couple of hours. As scheduling horizons can be longer than this, a better approach was needed.

The solution method used for this formulation was not optimized. First, the instrument decision variables could be eliminated entirely. Although including them in the formulation makes it easier to discuss, they are just linear combinations of the \( z \) variables (e.g. \( x_i(t) = \sum_m z_{im}(t) \)) and are redundant. Eliminating them would reduce the variables by the number of possibly active instruments in each time step times the number of time steps. Second, the branching rules used in the optimization could be modified to better exploit the nature
of the problem. What was used in this example were CPLEX’s default branching rules, not specifically tailored rules. However, since this formulation took so much longer to run than would be acceptable in a working situation, attention was turned from modifying this formulation to alternate approaches, as described in the next sections.

3.3.2 Split Approach (Formulation II)

While running the CPLEX solver, it was noted that the objective function value was within 4% of the linear program relaxation (this is the upper bound on the objective function value) in relatively few iterations (less than 5 seconds of elapsed time). Calculating the exact solution took several hours of elapsed time. Upon closer examination, it was noted that while the optimal (or near optimal) instrument solution is found quickly, finding a feasible mode assignment takes much longer. The mode ratio constraint (specifying the ratio of how much of each data taking mode is active during the scheduling horizon) coupled with the integer constraint and the instrument problem becomes a very difficult problem to solve.

It was therefore decided to decompose the problem into two subproblems. 1) Find the optimal schedule for the instruments, ignoring the mode ratio constraint. 2) Find the optimal distribution of modes in that schedule that satisfy the mode ratio constraint. This substantially reduced the search space for the modes and, as shown below, decreased the solution time enough to allow the formulation to be used in an operational setting without any loss of optimality.

The formulation looks much the same as the basic formulation. The main differences are that it is split into two separate optimization problems and the \( z \) variables are not required.

**Instruments (Formulation IIa)**

The only variables in the instrument problem are the instrument decision variables, \( x_i(t) \). The objective function and the equal usage constraint (3.7) stay the same. The resource constraints (3.9) and (3.10), along with constraint (3.1), are written in terms of the instrument decision variable, \( x_i(t) \), since the resource usages are assumed to be mode independent.
(i.e., $R_{ir} = R_{imr} \forall m$). See below for more detail.

Objective Function:

$$\text{maximize } \sum_t \sum_i V_i \times x_i(t)$$

Subject to:

$$x_i(t) = 0 \quad \forall \text{ times } t \text{ when instrument } i \text{ cannot be on}$$

$$|\sum_t x_i(t) - (\sum_t x_i(t))| \leq \delta_1 \quad \forall \text{ (i, i) pairs}$$

$$\sum_i R_{ir} \times x_i(t) \leq RI_r(t) \quad \forall \text{ rate limited } r, t$$

$$\sum_t \sum_i R_{ir} \times x_i(t) \leq RT_r \quad \forall \text{ volume limited } r, RC_r$$

$x_i(t)$ are $\{0, 1\}$ variables

Modes (Formulation IIb)

The only variables in the mode formulation are the mode decision variables, $y_m(t)$. The objective function is rewritten to minimize the amount of time spent in sleep mode. Each mode is given a value of 1, with the exception of sleep mode which is given a value of 0. This insures that data taking modes besides sleep will be active whenever possible.

Constraints (3.3) and (3.8) are the same. After having solved the instrument problem, the $x_i(t)$ variables are fixed. Using the relationship $z_{im}(t) = x_i(t) \times y_m(t)$, constraints (3.9), (3.10), and (3.2) can be written in terms of $y$ only.

With the instrument and mode decision variables decoupled, the formulation could admit solutions in which a time step both had no instruments on and was in a data taking mode besides sleep. This is avoided by modifying the time line input using the instrument solution. Any non-sleep time step that has no instruments on is converted to a forced sleep time step.
Objective Function:

\[
\text{maximize } \sum_t \sum_m V_m \times y_m(t)
\]

Subject to:

\[
\begin{align*}
y_m(t) &= 0 & \forall \text{ times } t \text{ when mode } m \text{ cannot be on} \\
\sum_m y_m(t) &= 1 & \forall t \\
|\left(\sum_t y_m(t)\right) - MR_{m, \tilde{m}} \times (\sum_t y_{\tilde{m}}(t))| &\leq \delta_2 & \forall (m, \tilde{m}) \text{ pairs} \\
\sum_i \sum_m R_{imr} \times x_i(t) \times y_m(t) &\leq RI_r(t) & \forall \text{ rate limited } r, t \\
\sum_t \sum_i \sum_m R_{imr} \times x_i(t) \times y_m(t) &\leq RT_r & \forall \text{ volume limited } r, RC_r
\end{align*}
\]

\(y_m(t)\) are \(\{0, 1\}\) variables

**Implementation Issues**

One assumption underlying this formulation is that the instruments and modes are completely independent. Unfortunately, this is not always true. Some of the constraints depend on mode as well as instrument (namely the resource usages). The mode problem accounts for the instruments that are on and uses the appropriate values. The instrument problem, on the other hand, has no information about modes but still needs to assume resource usage values.

For volume limited resources there is a simple work around. If it is assumed that the modes split according to the specified ratio within each renewal period, a modified resource usage can be created. This modified resource usage is just the average of the actual resource usages for each mode weighted by the expected amount of time spent in each mode. In a two mode example with mode \(a\) being used \(A\%\) of the time and mode \(b\) being used \(B\%\) of
the time (note \( A + B = 1 \)), then:

\[
R_{ir} = A \times R_{iar} + B \times R_{ibr}
\]

A similar modification of the rate limited resource usage is not effective. The averaging creates a modified amount of resource used that is either greater than or less than the actual. Over time, as in the volume limited case where the renewal interval is several time steps, the actual usage converges to this modified usage. In the rate limited case, however, the renewal interval is one time step. There is no time for the actual usage to converge. So this formulation is not appropriate for situations in which the rate limited resource usages depend on the data taking mode.

For situations in which only volume limited resource usages depend on the data taking mode, however, this is an appropriate formulation. Running this with the same sample problem and equipment as before generates a week's schedule in about 4 seconds. One item of note is that the instrument and mode problems do not take equal amount of time. The mode problem takes about 75% of the total solution time.

3.3.3 Instrument Only Approach (Formulation III)

For satellites that have mode dependent rate limited resource usages, another approach must be developed. One concept is to take the modes out of the optimization entirely. Note that the mode problem takes the bulk of the solution time and is practically solved by the mission scientists when they specify the desired mode ratio. Generally speaking, they are interested in one of two scenarios: all of one mode in a solid block followed by all of the next mode mode in a block; or smaller blocks of each mode in the proper ratio interleaved at given intervals. Either way, the solution to the mode problem has been pre-specified and it is arguably a waste of resources to have the scheduler resolve it. Instead, this formulation will only optimize over the instrument space, taking the modes as an input.
Formulation

If $y_m(t)$ are known constants, then the resulting formulation has only the variables $x_i(t)$ and has much the same form as the instrument part of the split formulation.

Objective Function:

$$\text{maximize } \sum_t \sum_i V_i \times x_i(t)$$

Subject to:

$$x_i(t) = 0 \quad \forall \text{ time steps } t \text{ when instrument } i \text{ cannot be on}$$
$$|\left(\sum x_i(t)\right) - \left(\sum x_i(t)\right)| \leq \delta_i \quad \forall (i, \bar{i}) \text{ pairs}$$
$$\sum_m \sum_i R_{imr} \times x_i(t) \times y_m(t) \leq RI_r(t) \quad \forall \text{ rate limited } r, t$$
$$\sum_t \sum_m \sum_i R_{imr} \times x_i(t) \times y_m(t) \leq RT_r \quad \forall \text{ volume limited } r, RC_r$$

$x_i(t)$ are $\{0, 1\}$ variables

Implementation Issues

This approach depends on the mission scientist intelligently deciding when the satellite should be in each mode, and to do so with a feel for how the resource usages will be affected. If this is not done intelligently, the results can be unfortunate, as in the following example. Assume a two mode situation where mode $a$ requires more resource than mode $b$. Furthermore, assume one of the resources is limited such that the instruments can only be on during half the time steps and consider only one renewal interval. Then if the scientist specifies mode $a$ half the time and mode $b$ the other half, the scheduler will automatically turn on all the instruments during the mode $b$ time steps and only a few mode $a$ time steps.
So the potential is that no instrument will ever be run in mode \( a \) despite what the scientist specified.

One way to guard against this might be to include a mode ratio-like constraint on the instruments.

\[
\left| \left( \sum_i x_i(t) \times y_m(t) \right) - M R_{m \tilde{m}} \times \left( \sum_i x_i(t) \times y_{\tilde{m}}(t) \right) \right| \leq \delta_2
\]

However more than one instrument can be on during one time step, so this does not guarantee the exact mode ratio desired, merely that the mode using more resources will be on at least part of the time.

Another way is to specify that the mode changes only occur at the renewal time for the limiting resource. This is actually a reasonable assumption if changing the data taking mode of an instrument requires that it be turned off and then back on again. In this case, the power off/on cycle takes some amount of time and resources by itself, and the number of such cycles should be minimized. (Note that a limit to the number of such cycles could be added to the basic formulation as well if it were deemed necessary.) This assumption also makes specifying the modes even easier.

With Formulation III and using the second method of insuring the correct mode ratio mentioned above, the sample problem described for the previous formulations produced a schedule for a week in about 2.5 seconds. Although there are ways to improve both the speed and memory usage of the scheduler, the goal was to develop a generic scheduler that finds an optimal schedule fast enough to be useful and generic enough to be easily adaptable to any satellite in this class. The next section presents the case study that generated the sample problem mentioned above.

3.4 Case Study: TERRIERS

TERRIERS is the acronym for Boston University's Tomographic Experiment using Radiative Recombinative Ionospheric EUV and Radio Sources satellite. The mission is to measure
photo-emissions from the earth’s atmosphere, creating a 3-D map of the ionosphere and thermosphere. TERRIERS is scheduled to launch in the fall of 1997 on a Pegasus launcher. The entire satellite weighs about 270 lbs and uses an average of 16 W of power generated by a solar array and four batteries. The satellite spins at 10 rpm in a cartwheel formation (its spin axis is perpendicular to its orbital plane) in a 550 km sun synchronous 97° inclined orbit.

There are five spectrometers, two photometers and a radio beacon onboard. Four of the spectrometers and the photometers are light-sensitive instruments designed for use solely at night. The fifth spectrometer is meant to be used only during the day. There is also an instrument to measure background photo-emissions from the sun (Gas Ionization Solar Spectral MOniter, GISSMO) onboard, and a possible experiment designed and built by a high school team.

The instruments can operate in five data taking modes: debug, synoptic, spectral, tomographic, and sleep. Debug and synoptic modes are diagnostic modes designed to be used for calibration and trouble shooting. When in spectral mode, the instruments measure data in 256 color channels and 4 position bins. In tomographic mode, they measure data in 20 color channels and 120 position bins. Sleep mode is the safety shut down mode.

The design of the satellite bus specifies that, at any given time, all of the active instruments must be in the same data taking mode. The satellite design also requires that whenever an instrument is turned on or off, or the mode is changed, all instruments are powered down and then back up again. For a week long scheduling horizon, the scientists have indicated a desire for the equivalent of six days in tomographic mode and one day in spectral mode.

All communication with the ground is accomplished by contact with a single ground station for an average of 10 minutes every 12 hours. Since this contact is infrequent, there are certain automatic safety features onboard. The satellite automatically checks that the photon count rate, voltage, current, temperature, and pressure are all within allowable limits and verifies whether it is in night or day sunlight conditions. If a limit on an instrument is reached, the instrument is considered to have “red lined” and is put into sleep mode (the instrument is turned off). If two or more instruments red line, the entire satellite is put into sleep mode (the entire payload is turned off). The information that sleep mode was forced is
downlinked on the next contact, along with the usual health and status data. The satellite (or the instrument) cannot be turned on again, despite what the schedule might specify, until an override command is issued from the ground.

The satellite has a nominal one year life span of activity to be scheduled in week long increments. Under the current operations concept, the instrument scheduling will be done manually. Since not all the instruments can be active all the time due to power limitations, they are put on a 50% duty cycle. The resulting default schedule is for all the instruments to be on for one orbit and off for the next.

There are two ways to utilize the capabilities of an automated scheduler in this situation. The first is to automate the scheduling process so that a better schedule is developed in fewer man hours. The second is to better respond to forced sleep mode. Twelve hours of additional scientific data can be collected by being able to receive the information that sleep mode was forced, diagnose why, generate a new schedule if necessary, and uplink the restart commands all in one contact period.

### 3.4.1 The Model

The TERRIERS satellite was still undergoing design revision during the time this research was done. Therefore, the satellite modeled here is based on, but is not identical to, the TERRIERS satellite. Any TERRIERS information that was available at the time has been used and reasonable estimates have been made for the unknown quantities. Note that the instrument only approach to scheduling (Formulation III from Section 3.3.3) is used.

The satellite has a scheduling horizon of a week, broken down into discrete 15 minute time steps. Several runs were made with 5 minute time steps to examine the effect of a finer time step on the scheduler. This length of this step represents the minimum amount of time an instrument can be on plus the time it takes to turn the instruments on or off. Since any change in the active instruments or modes requires the entire payload to cycle off and on, and since this is a deterministic model, the time of the power cycle and the resultant small loss of data taking time are assumed to be constant for every time step and are ignored.

Each orbit is 95.65 minutes long, with approximately 66% of that in day (sunlight) and 33%
in night (eclipse) conditions. The amount of eclipse time and when it occurs in the orbit each vary with the time of year due to the earth’s axial tilt. In addition, since some of the instruments can be on only during the day and some only during the night, there should be a few minutes of sleep mode between each day/night transition as a safety margin. However, for the purposes of demonstrating the scheduler, the scheduling horizon is approximated by one 90-minute orbit (made up of 60 minutes of day followed by 30 minutes of night) repeated for a week.

Only spectral, tomographic, and sleep modes are scheduled on a regular basis since debug and synoptic modes are used for debugging in the event of an anomaly. Remember (Section 3.3.3) that in order to keep the proper mode ratio in the final schedule, Formulation III specifies that the modes must be scheduled in blocks of time corresponding to the renewal interval of the limiting resource. In this case, power is the limiting resource and it renews every two orbits. This example uses a cycle of six two-orbit periods of tomographic mode followed by one two-orbit period of spectral mode and no pre-set sleep mode.

Only the five spectrometers and two photometers are modeled. GISSMO and the radio beacon both have additional individual constraints that were not clearly defined at the time this model was developed. In addition, the high school experiment is still in the design phase and no parameters are known yet.

Two instrument constraints are modeled. First, the day/night operating constraints must be honored. Second, the four night spectrometers are considered as four instances of the same instrument and as such are subject to the equal usage constraint (as are the two photometers).

Table 3.1 shows the relative value associated with each instrument, given in science points. Multiple instances of the same instrument are assumed to have the same value. Note that the instrument value does not depend on the data taking mode or time step.

The relevant resources were deemed to be power, memory, and CPU. As there is a limit on both the rate that power can be used and the total amount available, power is considered to be two separate resources: rate limited power and volume limited power. There is very little information available about the instrument usages of rate limited power, so estimated
Table 3.1: Relative values (in science points) of each instrument in the sample TERRIERS problem.

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Value (Science Points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Night Spectrometer</td>
<td>7</td>
</tr>
<tr>
<td>Photometer</td>
<td>5</td>
</tr>
<tr>
<td>Day Spectrometer</td>
<td>4</td>
</tr>
</tbody>
</table>

values are used. It is assumed that there is nominally enough rate limited power that all the allowable instruments could be on during one time step.

The total amount available and the instruments' usages of volume limited power are derived from the power budget of September 1996 [5]. It is assumed that volume limited power is fully renewed at the beginning of every other orbit and that the usage does not depend on the battery history. While this is only an estimate, the full battery model used by the TERRIERS designers indicates that it is a valid first order assumption [5]. As the batteries get older and change characteristics, the total power available can be changed in the formulation.

The nominal power requirements are shown in Table 3.2 and are independent of the data taking mode.

Table 3.2: The total available amount of each resource and the usages for each instrument for each resource in the TERRIERS sample problem. All instances of multiple instruments are assumed to have the same nominal power usages. If the usages differ with mode, the spectral mode usages are in parentheses.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Total Available</th>
<th>Night Spectrometer</th>
<th>Photometer</th>
<th>Day Spectrometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Limited Power (W)</td>
<td>48</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>CPU (sec)</td>
<td>48</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Volume Limited Power (W)</td>
<td>8.10</td>
<td>0.396</td>
<td>0.0747</td>
<td>0.600</td>
</tr>
<tr>
<td>Memory (Mb)</td>
<td>960</td>
<td>8 (6)</td>
<td>8 (6)</td>
<td>8 (6)</td>
</tr>
</tbody>
</table>

All data taken is stored in memory (RAM), so memory is the data storage resource. Assuming that data can always be recorded as fast as it can be taken, volume limited memory is the only potentially active memory resource constraint. The renewal interval is the time...
from one contact to the next. It is approximately 12 hours, but will actually change with the relative geometries of the ground station and the satellite (as will the length of the contact period). The worst case scenario of 12 hours is used as the renewal interval. Again, exact memory usages are not available and estimated values are used, but it is assumed that there is enough memory for all the instruments to be run continuously for one contact interval, and that all the data can be downlinked during one contact period (i.e., memory is renewed entirely at the start of each renewal interval). It is also assumed that memory usage changes with mode, so different estimated values are picked for each mode (see Table 3.2).

Although the satellite was designed with enough CPU capacity so that computing power should never be a limiting resource, this cannot be verified until the satellite is on orbit (or at least in hardware-in-the-loop simulation tests). Therefore, the scheduler was designed with a rate limited CPU resource placeholder. It is not, however, a limiting constraint (Table 3.2 shows the values used). Note that a volume limited CPU resource makes no physical sense and was not considered.

As mentioned in the formulation section, all of this information is used to create a small constraint matrix for each time step type / data taking mode combination. These small matrices are then combined in the correct order to create the full constraint matrix for the scheduling horizon. Figure 3-3 shows a sample small matrix.

3.4.2 Results

For the sample problem above, schedules for various scheduling horizons were generated using the commercial solver CPLEX running on a Sparc20 (using Solaris 2.5). A 15 minute time step is used. The durations are: 2 orbits (3 hours); 1 contact interval (12 hours); 1 day (24 hours); and 1 week (168 hours). Table 3.3 shows the results. Active time steps denotes the number of time steps during which the payload was not in sleep mode. The total number of time steps in the scheduling horizon is also shown. Schedule value is the number of science points the instruments would generate if the schedule were run. Also shown is the maximum number of science points that could be generated if there were no resource constraints (all instrument constraints are still honored). Scheduling time is the actual CPU time required to solve the integer program.
Figure 3-3: The formulation for one night time step of spectral mode. The columns denote the instruments (order: spectrometers A-D, photometers A-B) while the rows denote constraints (order: 1-2 are rate limited resource constraints, 3-4 are volume limited resource constraints, and 5-12 are equal usage constraints). The day spectrometer is constrained to be off during a night time step and is removed from the matrix entirely to avoid unnecessary variables.

Table 3.3: Statistics on scheduling the TERRIERS sample problem for varying durations, using a 15 minute time step.

<table>
<thead>
<tr>
<th>Duration</th>
<th>Time (hrs)</th>
<th>Time Steps active/total</th>
<th>Schedule Value (science points) actual/max</th>
<th>Scheduling Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 orbits</td>
<td>3</td>
<td>5/12</td>
<td>156/184</td>
<td>0.04</td>
</tr>
<tr>
<td>1 contact</td>
<td>12</td>
<td>20/48</td>
<td>624/736</td>
<td>0.08</td>
</tr>
<tr>
<td>1 day</td>
<td>24</td>
<td>40/96</td>
<td>1248/1476</td>
<td>0.16</td>
</tr>
<tr>
<td>1 week</td>
<td>168</td>
<td>280/672</td>
<td>8736/10304</td>
<td>2.21</td>
</tr>
</tbody>
</table>
The problem is solved quite quickly: in less than three seconds. Note that these times are for relative comparison only. No optimization was done on the code. Given the way the modes are specified in two orbit periods, the schedule repeats every two orbits. The only difference is which mode the instruments are in. The schedule (see Figure 3-4) specifies that all night instruments be on during all night time steps and the day instruments be on for one day time step every orbit (one time step of data from the day spectrometer is worth less and costs more than the same amount of data from one of the night instruments, so it is only scheduled if there are extra resources after scheduling all the night instruments).

Figure 3-5 shows part of the default schedule for comparison. The default schedule assumes a 50% duty cycle on all instruments, making it very easy to schedule. When all the instruments are only allowed to be on half the time, there are enough resources so that all the instruments can be on all the times allowed. However, it is a much less efficient schedule. It has a value of 92 science points, only 59% of the value of the schedule generated by the scheduler.

The same problem is run with 5 minute time steps to evaluate the impact of the finer resolution time line on the schedule value and scheduling time. Both the value for each instrument per time step and the resource usages per time step are scaled appropriately (divided by 3) so that the objective values can be compared directly. See Table 3.4 for results.

Table 3.4: Statistics on scheduling the TERRIERS sample problem for varying durations, using a 5 minute time step.

<table>
<thead>
<tr>
<th>Duration</th>
<th>Time (hrs)</th>
<th>Time Steps active/total</th>
<th>Schedule Value (science points) actual/max</th>
<th>Scheduling Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 orbits</td>
<td>3</td>
<td>17/36</td>
<td>159/184</td>
<td>0.06</td>
</tr>
<tr>
<td>1 contact</td>
<td>12</td>
<td>68/144</td>
<td>635/736</td>
<td>0.22</td>
</tr>
<tr>
<td>1 day</td>
<td>24</td>
<td>136/288</td>
<td>1269/1476</td>
<td>0.47</td>
</tr>
<tr>
<td>1 week</td>
<td>168</td>
<td>952/2016</td>
<td>8885/10304</td>
<td>8.33</td>
</tr>
</tbody>
</table>

The schedule now takes a bit longer to compute, but is still quick enough to reschedule during a contact period. The objective value is 0.7% higher since in each power renewal period there is enough unused power remaining in the schedule with 15 minute time steps.
Figure 3.4: Illustrative segment of the resultant week-long schedule.
Figure 3.5: Illustrative segment of the default week long schedule assuming a 50% duty cycle on all the instruments.
to run the instruments for about ten more minutes. While the schedule with 15 minute
time steps cannot take advantage of this, the schedule with 5 minute time steps can.
Chapter 4

Scheduling 3-Axis Stabilized Science Satellites

The next class of satellites considered is the class of 3-axis stabilized satellites with missions involving global science studies. In other words, the instruments are designed to image the entire earth on a constant, regular basis to form a baseline measurement of the earth and/or the atmosphere. Since a global baseline is desired, there is no need to look at a specific target area for as long as possible, or to follow a particular phenomenon.

This chapter presents a scheduler for this class of satellites. Since the basic modeling assumptions made here are the same as in the spinning satellite class, they are not restated. Instead, only the characteristics unique to this class are detailed and then the formulation and the case study are presented.

4.1 Class Description

As the name of the class implies, these satellites are not spinning but remain fixed at a constant attitude with respect to the earth. Like the spin stabilized class, the instruments themselves remain at a constant angle with respect to the satellite. The instruments are only on if they are pointed at an area of interest, and none of their data is discarded as it is in the spin stabilized class.
4.1.1 Instruments

Each instrument is assumed to operate independently of all the others. This means that they can all operate simultaneously: they have their own optics and the measurements of one instrument does not interfere with the measurements of another.

When a mission includes more than one instrument, it is possible that each of the instruments will be designed to share the same optical system. This generally saves a great deal of on-orbit mass and power. However, the cost is that the optics must be shared in some fashion, thus creating time sharing problems and decreasing the resolution of the instruments (since resolution is a function of wavelength and the size of the optics, two instruments imaging in different wavelengths but using the same optics will have different resolutions). While there are satellites of this type, the class of satellites discussed here will only include those satellites for which all instruments have their own optics.

The intersection of each instrument's field of view (a cone with its apex at the instrument, centered around the instrument's line of sight) and the earth is the ground area that an instrument can image and is called its field of view (FOV). Due to size and mass considerations of the detector array and focal plane, an instrument generally cannot image its entire FOV at once. Instead, it images a smaller area called an instantaneous field of view (IFOV) (see Figure 4-1). The detector array is moved or rotated in some coherent manner so that the entire FOV is imaged over time. It is assumed that instruments do not interfere with each other during this process. In other words, there are no vibrations or power losses affecting the performance of one instrument due to another's detector array scan.

As the satellite moves forward in its orbit, new areas are brought into view. The particular scan pattern an instrument employs depends on the parameters of the instrument: the size and type of array, the FOV size, etc. It is possible to model the instrument and optimize this scan pattern [50]. However, since each instrument has its own optics and can scan independently, the scan pattern is not dependent on the state of the satellite and can be generated off line before the mission. This way it does not have to be part of the overall instrument schedule.

It is also assumed that for this class of satellites, each instrument can image its entire FOV.
Each IFOV is imaged very quickly (on the order of nanoseconds), but depending on the relative sizes of the IFOV and FOV, and the scanning speed of the detector array, it is possible that the entire FOV cannot be imaged before the area has been passed. If the entire area cannot be imaged, it is still possible to generate a scan pattern that allows the instrument to image the most area although there will be gaps in the coverage. For this class of satellites, however, it is assumed that the entire FOV can be imaged.

Furthermore, the instruments are assumed to be nadir-pointing. If the entire FOV can be imaged and the goal is global coverage, the instrument can be pointed at a constant angle to the line between the center of the earth and the satellite. The motion of the orbit will eventually allow the entire globe to be imaged. Since image resolution is a function of distance from the optics to the object being imaged (among other things), the best resolution is achieved when that angle is zero. Then the instrument is pointing “straight down”, or towards nadir. The FOV is centered at the point on the earth’s surface directly beneath the satellite, or sub satellite point (see Figure 4-2).

### 4.1.2 Experiments

In typical missions, some of the desired information can only be derived by combining simultaneously gathered data from several different instruments. When such co-registration
is necessary, all of the involved instruments image the same area at the same time. This helps eliminate corruption from effects such as atmospheric and sun incidence angle interference. In reality, the images do not need to be of the exact same place at the exact same time. However, the time scale of the acceptable shift is much smaller than the scheduler's time step. For the purposes of scheduling, it is required that the instruments image the same place at the same time.

A group of instruments that must image the same area at the same time in order to produce the desired information will be called an "experiment." Note that in this case all of the instruments reside on the same satellite and all are nadir pointing, so all of the instruments are pointing in the same direction and the spatial co-registration is achieved automatically. In order for an experiment to be running then, all of the required instruments must be on.

In this class of satellites, the decision variables determine whether the experiments are on or off. Values are associated with experiments, not with the individual instruments themselves. This is a generalization of the previous formulation since an experiment can, of course, comprise only one instrument. Note that the experiment FOV is the intersection of the area that all of the required instruments can image. In other words, it is the intersection of all the required instruments' FOVs (see Figure 4-3).
4.2 Mathematical Programming Formulation

The formulation for this class of satellites has much the same form as that for the spin stabilized science satellite class. In fact, this formulation is based on the instrument only approach (formulation III from Section 3.3.3).

Again, the inputs capture all of the information about the system: the time step type time line; the data taking mode time line; the system constraints; and the instrument information. The decision to be made is which experiments are active during which time steps and which instruments should be turned on to execute these experiments. In addition, the experiment definitions and the relative value for one time step of data from each experiment must also be specified and modeled.

The output schedule specifies which instruments, if any, should be on during each time step in the scheduling horizon. It also specifies which experiments are active during each time step and the time history of the resource usages.

Notation

Indices:

- time step: \( \{ t \in (1, T) \text{ where } T \text{ is the total number of time steps} \} \)
- mode: \( \{ m \in (1, M) \text{ where } M \text{ is the total number of modes} \} \)
• experiment: \( \{ e \in (1, E) \text{ where } E \text{ is the total number of experiments} \} \)

• instrument: \( \{ i \in (1, I) \text{ where } I \text{ is the total number of instruments} \} \)

• resource: \( \{ r \in (1, R) \text{ where } R \text{ is the total number of resources} \} \)

Constants:

• \( RIT_r(t) \) – amount of rate limited resource \( r \) available in time step \( t \)

• \( RT_r \) – amount of volume limited resource \( r \) available in one renewal period

• \( RC_r \) – renewal period of resource \( r \)

• \( R_{i m r}(t) \) – amount of resource \( r \) used by instrument \( i \) in mode \( m \) during time step \( t \)

• \( V_e \) – value of experiment \( e \) being on for one time step

• \( \rho_e \) – number of instruments in experiment \( e \)

• \( I_e \) – set of instruments in experiment \( e \)

• \( y_m(t) = \begin{cases} 1 & \text{if mode } m \text{ is active during time step } t \\ 0 & \text{otherwise} \end{cases} \)

Decision Variables:

\[ x_i(t) = \begin{cases} 1 & \text{if instrument } i \text{ is on during time step } t \\ 0 & \text{otherwise} \end{cases} \]

\[ w_e(t) = \begin{cases} 1 & \text{if experiment } e \text{ is active during time step } t \\ 0 & \text{otherwise} \end{cases} \]

Objective Function

\[ \text{maximize } \sum_i \sum_e V_e \times w_e(t) \]
The mission scientists set the relative value of each experiment, \( V_e \). This value is not currently dependent on the data taking mode or the time step. However, if that were deemed necessary, it would be a straightforward modification. The value of the schedule is the value of all the experiments that are on during the scheduling horizon.

Constraints

\[
x_i(t) = 0 \quad \forall \text{times } t \text{ when instrument } i \text{ cannot be on} \quad (4.1)
\]

\[
w_e(t) = 0 \quad \forall \text{times } t \text{ when experiment } e \text{ cannot be on} \quad (4.2)
\]

\[
x_i(t) + x_{i'}(t) = 1 \quad \forall \text{instruments } i \text{ and } i' \text{ that cannot be on simultaneously} \quad (4.3)
\]

\[
w_e(t) + w_{e'}(t) = 1 \quad \forall \text{experiments } e \text{ and } e' \text{ that cannot be on simultaneously} \quad (4.4)
\]

\[
\sum_m \sum_i R_{imr} \times x_i(t) \times y_m(t) \leq RI_r(t) \quad \forall \text{rate limited } r, t \quad (4.5)
\]

\[
\sum_t \sum_m \sum_i R_{imr} \times x_i(t) \times y_m(t) \leq RT_r \quad \forall \text{volume limited } r, RC_r \quad (4.6)
\]

\[
\rho_e w_e(t) \leq \sum_{t_e} x_i(t) \quad \forall \ e, \ t \quad (4.7)
\]

\( x_i(t), w_e(t) \) are \( \{0, 1\} \) variables

Constraints (4.1) and (4.2) are analogous to constraints (3.1) and (3.2). Certain instruments and certain experiments cannot or should not be on during certain time steps. For instance, it makes no sense to run an experiment measuring the sun incidence angle on clouds during the night.

Constraints (4.3) and (4.4) take into account that there might be sets of instruments, or sets of experiments, that cannot be run simultaneously. The constraints specify that only one of the two can be active in any given time step.

The rate and volume limited resource constraints are expressed in (4.5) and (4.6) respec-
tively. These are functionally the same as constraints (3.9) and (3.10), although they have
the form of the resource constraints in the instrument only formulation of Section 3.3.3.
Note that once again, care should be taken when setting up the mode time line. Mode
changes should only come at the renewal intervals of the renewable resources.

The experiment definitions are embodied in constraint (4.7). Note that the only instruments
included in the sum are the instruments required for the experiment. No experiment can
be on unless all of its required instruments are on.

Implementation

This scheduler is implemented in the same manner as the one for the spin stabilized science
satellite case. The complete constraint matrix is built up from the constraint matrices of
the individual time steps. Then, the resulting integer programming problem is solved using
CPLEX.

A sample problem involving 8 experiments, 5 instruments, 1 mode, and 4 time step types
was formulated in this manner. Running on a Sparc20, it produced a schedule for one orbit
in 1.21 seconds. The next section describes this sample problem in more detail.

4.3 Case Study: EOS AM-2

Part of NASA’s Mission to Planet Earth initiative is the Earth Observing System (EOS)
series of satellites. EOS is designed to be a 15-year continuous study of the global environ-
ment, focusing on seven different areas [3]:

- Water and Energy Cycles
- Oceans
- Chemistry of Troposphere and Lower Stratosphere
- Land Surface Hydrology and Ecosystem Processes
- Glaciers and Polar Ice Sheets
• Chemistry of Middle and Upper Stratosphere

• Solid Earth

There are 24 different measurements that will be made in these areas throughout the entire mission lifespan [10].

EOS is composed of three main satellite systems: AM, PM, and CHEM. These three systems will be maintained continuously throughout the EOS lifetime. In addition there are several other, smaller satellites that are also part of the EOS program.

Since it is very expensive and possibly not feasible to build satellites that will remain fully functional on orbit for 15 years, each main system is to be made up of three satellites. Each satellite will have a lifetime of 5-6 years (see Figure 4-4 for a timetable). This will also allow instruments to be replaced and updated periodically, taking advantage of any new information gained to date and developing technology. For example, EOS AM-2 is the second satellite of the AM series.

<table>
<thead>
<tr>
<th>Year</th>
<th>Satellite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>AM-1</td>
</tr>
<tr>
<td>1998</td>
<td>PM-1</td>
</tr>
<tr>
<td>1999</td>
<td>CHEM-1</td>
</tr>
<tr>
<td>2000</td>
<td>AM-2</td>
</tr>
<tr>
<td>2001</td>
<td>PM-2</td>
</tr>
<tr>
<td>2002</td>
<td>CHEM-2</td>
</tr>
<tr>
<td>2003</td>
<td>AM-3</td>
</tr>
<tr>
<td>2004</td>
<td>PM-3</td>
</tr>
<tr>
<td>2005</td>
<td>CHEM-3</td>
</tr>
<tr>
<td>2006</td>
<td>AM-3</td>
</tr>
<tr>
<td>2007</td>
<td>PM-3</td>
</tr>
<tr>
<td>2008</td>
<td>CHEM-3</td>
</tr>
</tbody>
</table>

Figure 4-4: Launch timetable for the major EOS satellites.

All nine of the main EOS satellites are in 705 km altitude, 98.2° inclined sun synchronous orbits. This allows for a 98.8 minute orbit and a 16 day repeat ground track. The AM
series has a 10:30 am descending node crossing time while the PM and CHEM serieses have an ascending node crossing time of 1:30 pm and 1:45 pm respectively.

EOS AM-2 will have five instruments onboard: the Clouds and Earth's Radiant Energy System (CERES); the Earth Observing Scanning Polarimeter (EOSP); the Landsat Advances Technology Instrument (LATI); the Multi-angle Imaging Spectroradiometer (MISR); and the Moderate Resolution Spectroradiometer (MODIS). Note that some of these instruments are to be flown on earlier missions and AM-2 is expected to fly advanced versions (namely AMISR and AMODIS).

CERES is to take continuous cloud and radiation flux measurements as inputs for oceanic and atmospheric models. EOSP is the only instrument onboard with polarization channels. Its mission is to examine cloud and aerosol properties. LATI is the Landsat mission follow-on. As such it must continue Landsat's mission to provide timely, high quality visible and infrared images of all of the earth's land masses and near coastal areas [60]. MISR will look at clouds and the earth's surface with cameras at nine different angles. The goal is to eliminate some of the sun angle effects and accordingly, the instrument will only be run during the day. It also has two data taking modes: a high and a low resolution mode. MODIS is the centerpiece of many of the EOS missions. It is designed to provide continuous data from the earth's surface and the lower atmosphere. It is also the only instrument with a 100% duty cycle.

4.3.1 The Model

EOS AM-1 is due to launch in June of 1998 and as such has a fairly fixed design. AM-2 is not due to launch for another six years, and is still in the design phase. All perspective AM-2 characteristics that were available at the time of this research, have been used. Otherwise, AM-1 characteristics and estimated values were used.

Because the satellite orbit has been designed to have a 16 day repeat ground track, 16 days may seem to be a natural scheduling horizon. On closer inspection, an horizon of one orbit is more appropriate as all of the resources are renewed every orbit, and there are no constraints that continue over more than one orbit. Thus the 16-day schedule is just the
one orbit schedule repeated over and over.

Each 98.8 minute orbit is approximated by 20 five minute time steps. The satellite is in eclipse 36% of the time, so there are 13 day and 7 night time steps. The time required to turn the instruments on or off and any scheduled maintenance are ignored.

There are no satellite-wide data taking modes for EOS AM-2 like there are for TERIERS. The problem is modeled with one default mode. MISR, however, has two modes. To handle this, two different instruments are defined: MISR-global (low resolution) and MISR-local (high resolution) and it is specified that these two instruments cannot be on at the same time. MISR is planned to be on in local mode six times per day, where those six times are at the mission scientist’s discretion. This is specified through a MISR-local day time step. The other time step types are MISR-global day, night, and sleep.

The AM satellites make 15 of the 24 EOS measurements. Omitting most of the experiments that use the same sets of instruments (to keep the problem smaller but just as interesting), there are eight experiments which will be use in this problem. Their associated values are an approximation of the respective values the mission scientists put on one time step of data from each experiment. Table 4.1 lists the experiments and their values.

Table 4.1: Experiment topics, instrument requirements, and values (in units of science points).

<table>
<thead>
<tr>
<th>Experiment Name</th>
<th>Required Instruments</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Cloud properties</td>
<td>EOSP, MISR, MODIS</td>
<td>5</td>
</tr>
<tr>
<td>2 Radiative energy fluxes</td>
<td>CERES, MISR, MODIS</td>
<td>4</td>
</tr>
<tr>
<td>3 Atmospheric temperature</td>
<td>MODIS</td>
<td>3</td>
</tr>
<tr>
<td>4 Land cover and use change</td>
<td>LATI, MISR, MODIS</td>
<td>2</td>
</tr>
<tr>
<td>5 Land surface temperature</td>
<td>LATI, MODIS</td>
<td>1</td>
</tr>
<tr>
<td>6 Ocean surface temperature</td>
<td>MODIS</td>
<td>5</td>
</tr>
<tr>
<td>7 Land ice</td>
<td>LATI</td>
<td>4</td>
</tr>
<tr>
<td>8 Sea ice</td>
<td>LATI, MODIS</td>
<td>4</td>
</tr>
</tbody>
</table>

The modeled constraints are the experiment definitions and the resource constraints. The instruments involved in each experiment are listed in Table 4.1. Table 4.2 lists the individual instrument resource usages and the total amount available for each resource. Note that MISR has two usage values listed, signifying global and local mode respectively. Local
mode is assumed to take 10% more resources than global mode.

Table 4.2: Resource usage values for each instrument. The two values listed for MISR denote its resource usages in global and local mode respectively.

<table>
<thead>
<tr>
<th></th>
<th>CERES</th>
<th>EOSP</th>
<th>LATI</th>
<th>MISR</th>
<th>MODIS</th>
<th>Total Available</th>
<th>Renewal Period</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rate Limited</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7/8</td>
<td>7</td>
<td>36</td>
<td>–</td>
</tr>
<tr>
<td>CPU</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7/8</td>
<td>7</td>
<td>36</td>
<td>–</td>
</tr>
<tr>
<td><strong>Volume Limited</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power (W)</td>
<td>58</td>
<td>19</td>
<td>81</td>
<td>88/97</td>
<td>170</td>
<td>7704</td>
<td>1 orbit</td>
</tr>
<tr>
<td>Data Rate (Mbps)</td>
<td>0.10</td>
<td>0.44</td>
<td>60</td>
<td>8.2/9.0</td>
<td>6.2</td>
<td>1040</td>
<td>1 orbit</td>
</tr>
</tbody>
</table>

Power and computer processing capability are the important rate limited resources. Since the actual values have not yet been set, estimates are used. No limits have been put on the number of instruments that can be on simultaneously, so it is assumed that power and CPU are not limiting resources.

Power and data rate are the important volume limited resources. The listed usages are the average expected usages from a preliminary study done by S. P. Neeck [31]. The satellite is not expected to be power limited under normal circumstances, so the total power available was set so all the instruments can be on the entire orbit. It is also assumed that the solar arrays recharge the batteries completely every orbit.

In reality, there are two different processes happening with the data during an orbit: the instruments are taking data and storing it in the onboard memory, and the satellite is downlinking stored data from the onboard memory to the ground station. However, the process is modeled as though the data from the instruments is downlinked directly. Not only does this make the formulation simpler, but it also insures that the satellite is capable of downlinking all the data in steady state, and that the renewal rate is one orbit. The current estimate is that EOS AM-2 will have a 400 Mbps downlink rate [10]. The downlink capability shown in Table 4.2 assumes this rate and one 13-minute ground station contact per orbit. Note that the satellite is data rate limited.

The only other explicit constraint is that MISR must be on during a MISR-local day time
step. The 100% duty cycle on MODIS is not explicitly enforced but will occur in the schedule because of the high values of any experiment involving MODIS.

### 4.3.2 Results

A schedule for one orbit of the sample problem described above was generated using the commercial solver CPLEX running on a Sparc20 (running Solaris 2.5). Table 4.3 shows the results. Active time steps denotes the number of time steps during which the payload was not in sleep mode. The total number of time steps in the scheduling horizon is also shown. Schedule value is the number of science points the instruments would generate if the schedule were run. Also shown is the maximum number of science points that could be generated if there were no resource constraints (all instrument constraints are still honored). Scheduling time is the actual CPU time required to solve the integer program.

<table>
<thead>
<tr>
<th>Duration</th>
<th>Time (min)</th>
<th>Time Steps active/total</th>
<th>Schedule Value (science points) actual/max</th>
<th>Scheduling Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 orbit</td>
<td>100</td>
<td>20/20</td>
<td>420/483</td>
<td>1.21</td>
</tr>
</tbody>
</table>

Note that longer schedules are multiples of the schedule for one orbit.

The schedule itself is shown in Figure 4-5. All the instruments are on during the day, while only MODIS is on at night. Since MISR cannot be on at night, CERES and EOSP also generate no value at night (there is no experiment that uses either one of them without MISR). So there are five instruments that are possibly on during a day time step (resulting in all eight experiments) and only two instruments possibly on during a night time step (resulting in only 5 experiments), making a day time step more valuable than a night time step. And the scheduler deems that, for their respective data rate usages, a night time step of LATI is worth less than a night time step of MODIS.

A manually created schedule would probably look very much the same. There is a possibility that the human scheduler would decide that the schedule looked uneven and turn LATI on for a few night time steps instead of day time steps. However, given the experiment values,
that would only reduce the value of the schedule by at most 3%. From this perspective, this particular problem is not all that interesting for the automated scheduler, but will serve as a baseline for the case study in the following chapter.
Chapter 5

Scheduling Clustellations of Science Satellites

Consistent with the “smaller, cheaper, faster” paradigm, satellite design is moving towards smaller satellites with more focused missions. The “great observatories,” like the Hubble Space Telescope [69], that take decades to design and build but that record data on every conceivable phenomenon are being replaced by small satellites, like Clementine [68]. These satellites are designed and built in a matter of years, cost less, and return equivalent data on a few specific phenomena.

There are some missions, however, that cannot be executed by a single small satellite. For instance, one small satellite cannot support the large aperture needed for high resolution radar. This leads to the concept of multiple small satellites working together to accomplish a common goal. No single small satellite can carry a large enough filled aperture, but ten satellites, each with a small aperture, can work together to create a large, synthetic aperture radar (SAR).

Satellites working together in concert are traditionally referred to as a “constellation.” The most commonly thought of constellation is a group of satellites, all with the same altitude, eccentricity, and inclination but different nodal crossing times and spacings in the orbits. Figure 5-1 shows an example of the most common constellation: a Walker Delta pattern. The example has six satellites, each inclined at 60° with respect to the equator. The
satellites are equally spaced about the globe, and image every point on the globe within some time interval. This interval is a function of many things including the number of satellites. Note that given enough satellites, the entire globe can be imaged simultaneously. This is true of the Global Positioning Satellite (GPS) constellation.

![Figure 5-1: Illustration of a Walker Delta constellation viewed from the north pole.](image)

While this concept has the added advantage of making the mission more robust (one failure might degrade the system but will not end the mission), it also increases the operational costs. Automated operations tools, such as a scheduler, which reduce costs by increasing efficiency are not just useful but necessary.

This chapter presents a scheduler for a particular type of constellation called a clustellation. Once again, the class description is presented, along with the formulation and a case study. Since the clustellation can be arranged in many different ways (number of satellites in the cluster, number of instruments on each satellite, etc.) three different scenarios are presented as case studies.

### 5.1 Class Description

The individual satellites discussed here fall into the same class as described in Chapter 4. They are 3-axis stabilized, science satellites with independent, nadir pointing instruments. However, they differ from those in the last chapter in that more than one satellite is necessary to achieve a common experiment goal. The instruments involved in the experiments are not necessarily on the same satellite.
The assumption made for these satellites is that instruments onboard different satellites need to image the same place at the same time. For instance, if two instruments that image in two different spectral frequencies are onboard different satellites but both images are needed to generate the desired data. Thus the satellites' orbits should be close enough together that there is considerable overlap of the instruments' fields of view at all times. This class of satellites does not, however, necessarily require simultaneous full global coverage. This means that the traditional constellation is not ideal (too costly) for these types of missions.

Instead, a type of constellation called a “clustellation” [42] will be assumed. Instead of the satellites evenly spaced around the globe, they are placed in a tight formation, or cluster. There are several different ways to accomplish this: (a) the satellites are in the same orbit following one right after the other; (b) they are in orbits with slightly different nodal crossing times; or (c) they are flying side by side in non-keplarian orbits (see Figure 5-2). Each of these concepts allows the satellites to co-register their images and gives them a ground-track repeat time that is the same as that of the single large satellite.

![Figure 5-2: Three example cluster configurations.](image)

5.2 Mathematical Programming Formulation

The formulations presented in this chapter are based on the formulations developed previously. There are two major differences: the instruments are now on multiple satellites; and the data from several instruments must be co-registered in space as well as time. For simplicity and clarity, it will also be assumed that there is only one data taking mode for the mission. The formulation can easily be extended to use multiple data modes following
the format used in the previous chapters.

Two different approaches are detailed here. The first is again the more direct approach although it is not feasible to implement. The second takes advantage of the nadir-pointing aspect of this class of satellites to drastically reduce the number of variables and constraints.

5.2.1 Grid Approach (Formulation I)

Ignoring the co-registration problem for a moment, the question of how the instrument resource usages and the satellite resource availabilities are affected by the multiple satellites is addressed. It is assumed that an instrument’s resource usages remain the same regardless of which satellite it is on. However, the total resource availability and the renewal rates can vary for different satellites. Thus, care must be taken in writing the constraints so that the resource usages are summed over only the instruments that are actually on the satellite.

The orbit of each satellite and which instruments are onboard each satellite are inputs to the problem. This, in turn, means that the position and orientation of each instrument is known at every point in time. The time step type time line, which must be generated for each satellite, can be used to convey this information.

For an experiment to be active, all the required instruments must be on at the same time, imagine the same place. Previous formulations divide the time line into discrete time steps to facilitate the time co-registration. The instruments are specified to be either on or off for the duration of a time step. Continuing with that approach, this formulation will divide the globe into a grid of locations. For each time step, the sub satellite position of each satellite can be located on this grid. Since the sub satellite point is also the center of the instrument FOV, if the size of the FOV is known, all the grid locations in the instrument’s FOV can be identified. For each of the grid locations, a \( \{0,1\} \) variable is defined: it is set to a value of 1 if the instrument is turned on and is over that location during the current time step. These variables then become part of the experiment definitions as in the previous formulation (see Section 4.2). A discussion of appropriate grid sizes is presented later in this chapter.
Notation

Indices:

- time step: \( \{ t \in (1, T) \text{ where } T \text{ is the total number of time steps} \} \)
- grid location: \( \{ g \in (1, G) \text{ where } G \text{ is the total number of global grid locations} \} \)
- satellite: \( \{ s \in (1, S) \text{ where } S \text{ is the total number of satellites} \} \)
- experiment: \( \{ e \in (1, E) \text{ where } E \text{ is the total number of experiments} \} \)
- instrument: \( \{ i \in (1, I) \text{ where } I \text{ is the total number of instruments} \} \)
- resource: \( \{ r \in (1, R) \text{ where } R \text{ is the total number of resources} \} \)

Constants:

- \( R Ir_s(t) \) – amount of rate limited resource \( r \) available on satellite \( s \) in time step \( t \)
- \( RT_{rs} \) – amount of volume limited resource \( r \) available on satellite \( s \) in one renewal interval
- \( RC_{rs} \) – renewal period of resource \( r \) on satellite \( s \)
- \( R_{ir} \) – amount of resource \( r \) used by instrument \( i \)
- \( V_e \) – value of experiment \( e \) being on for one time step
- \( \rho_e \) – number of instruments in experiment \( e \)
- \( I_e \) – set of instruments in experiment \( e \)
- \( F_i \) – radius of the FOV for instrument \( i \)
- \( y_{it} \) – global grid location of the center of instrument \( i \)’s FOV at time \( t \)
Decision Variables:

- \( x_{ig}(t) = \begin{cases} 
  1 & \text{if instrument } i \text{ is on over location } g \text{ during time step } t \\
  0 & \text{otherwise} 
\end{cases} \)

- \( w_{eg}(t) = \begin{cases} 
  1 & \text{if experiment } e \text{ is active over location } g \text{ during time step } t \\
  0 & \text{otherwise} 
\end{cases} \)

Objective Function

\[
\text{maximize } \sum_t \sum_e \sum_g V_e \times w_{eg}(t)
\]

Note that if necessary, the experiment value could also be dependent on the location and the time \((V_{eg}(t))\).

Constraints

- \( x_{ig}(t) = 0 \quad \forall \text{ times } t \text{ when instrument } i \text{ cannot be on} \)
- \( w_{eg}(t) = 0 \quad \forall \text{ times } t \text{ when experiment } e \text{ cannot be on} \)
- \( x_{ig}(t) + x_{ig}(t) = 1 \quad \forall \text{ instruments } i \text{ and } i' \text{ that cannot be on simultaneously} \)
- \( w_{eg}(t) + w_{eg}(t) = 1 \quad \forall \text{ experiments } e \text{ and } e' \text{ that cannot be on simultaneously} \)
- \( \sum_{i \in S_r} R_{ir} \times x_{iy_{ir}}(t) \leq RI_{rs}(t) \quad \forall \text{ rate limited } r, s, t \)
- \( \sum_t \sum_{i \in S_r} R_{ir} \times x_{iy_{ir}}(t) \leq RT_{rs} \quad \forall \text{ volume limited } r, s, RC_{rs} \)
- \( \rho_e w_{eg}(t) \leq \sum_{i \in S_e} x_{ig}(t) \quad \forall e, t, g \)

\( x_{ig}(t), w_{eg}(t) \) are \( \{0, 1\} \) variables
None of these constraints is particularly difficult to write. The problem is in this implementation. The logical choice for the size for each global grid location is the IFOV of the instruments. Note that while the instruments’ IFOVs are not all the same size, they are all of the same rough order of magnitude. For argument’s sake, assume a 10 x 10 km location. This means that there are about 16 million locations on the globe for each instrument/experiment and each time step. Many of these locations are not feasible because of geometry. An instrument with a horizon to horizon FOV in a 700km orbit covers about 250 thousand grid locations. Assume the 5 instruments, 8 experiments, and 20 time steps of the example in Section 4.3 and that means there are on the order of 65 million variables in the problem. This is not a problem that can be reasonably solved, so it was not implemented. An alternative approach is developed in the next section.

5.2.2 Overlap Approach (Formulation II)

This approach takes advantage of the fact that the location of each instrument’s FOV on the globe is known at any given time. If all the FOVs of an experiment are located on the globe for a given time step, the experiment’s effective area corresponds to the intersection of all the FOVs (see Figure 5-3). By using this effective experiment FOV as a measure of the area of an experiment, it is not necessary to sum the experiment value over all the global grid locations.

Figure 5-3: The effective experiment FOV for an experiment requiring three different instruments is the intersection of the three different instrument FOVs.

For nadir pointing instruments, the effective experiment FOV is the largest when all the instruments are on the same satellite. Since the effective experiment FOV for a time step is a measure of the amount of experiment data collected during a time step, this configuration
(which is discussed in Chapter 4) gives the maximum amount of data. In comparison, the configuration discussed here with the instruments separated on different satellites results in some loss of experimental data. This can be modeled as a loss of experiment value. $M_e$ is defined as the ratio of the actual to the maximum experiment FOV size. In Figure 5-3, the experiment FOV would be the entire smallest FOV if all the instruments were co-located. The effective experiment FOV for the separation shown is approximately $4/5$ of this, so $M_e = 0.8$.

The following formulation is based on the satellite formulation of Section 4.2, updated for the multiple satellites. The value of each experiment is modulated by the effective experiment FOV factor.

Notation

Indices:

- time step: \( \{t \in (1, T) \} \) where \( T \) is the total number of time steps
- satellite: \( \{s \in (1, S) \} \) where \( S \) is the total number of satellites
- experiment: \( \{e \in (1, E) \} \) where \( E \) is the total number of experiments
- instrument: \( \{i \in (1, I) \} \) where \( I \) is the total number of instruments
- resource: \( \{r \in (1, R) \} \) where \( R \) is the total number of resources

Constants:

- \( RI_{rs}(t) \) – amount of rate limited resource \( r \) available on satellite \( s \) in time step \( t \)
- \( RT_{rs} \) – amount of volume limited resource \( r \) available on satellite \( s \) in one renewal period
- \( RC_{rs} \) – renewal period of resource \( r \) on satellite \( s \)
- \( R_{ir}(t) \) – amount of resource \( r \) used by instrument \( i \) during time step \( t \)
- \( V_e \) – value of experiment \( e \) being on for one time step

83
• $M_e$ – loss of value of experiment $e$ due to effective experiment FOV

• $\rho_e$ – number of instruments in experiment $e$

• $I_e$ – set of instruments in experiment $e$

Decision Variables:

- $x_i(t) = \begin{cases} 1 & \text{if instrument } i \text{ is on during time step } t \\ 0 & \text{otherwise} \end{cases}$

- $w_e(t) = \begin{cases} 1 & \text{if experiment } e \text{ is active during time step } t \\ 0 & \text{otherwise} \end{cases}$

Objective Function

$$\max \sum_t \sum e V_e x_e M_e x w_e(t)$$

Constraints

$$x_i(t) = 0 \quad \forall \text{times } t \text{ when instrument } i \text{ cannot be on} \quad (5.1)$$

$$w_e(t) = 0 \quad \forall \text{times } t \text{ when experiment } e \text{ cannot be on} \quad (5.2)$$

$$x_i(t) + x_i(t) = 1 \quad \forall \text{instruments } i \text{ and } \hat{i} \text{ that cannot be on simultaneously} \quad (5.3)$$

$$w_e(t) + w_e(t) = 1 \quad \forall \text{experiments } e \text{ and } \hat{e} \text{ that cannot be on simultaneously} \quad (5.4)$$

$$\sum_{i \in s} R_{ir} x_i(t) \leq R_{is}(t) \quad \forall \text{rate limited } r, s, t \quad (5.5)$$

84
\[ \sum_t \sum_{i \in \mathcal{I}} R_{ir} \times x_i(t) \leq RT_{rs} \quad \forall \text{ volume limited } r, s, RC_{rs} \] (5.6)

\[ \rho_e w_e(t) \leq \sum_t x_i(t) \quad \forall \ e, \ t \] (5.7)

\( x_i(t), w_e(t) \) are \( \{0, 1\} \) variables.

All the resource related values and the experiment definitions and values are defined as they were previously. The one missing piece is the loss of experiment value due to the less than maximum effective experiment FOV. The calculation of this modulator, \( M_e \), will be discussed in the following section. Note, however, that with this formulation it is no longer possible to differentially value grid locations \( (V_{eg}(t)) \), only experiments and time steps \( (V_e(t)) \).

### 5.3 Effective Experiment FOV

Calculating the effective experiment FOV requires knowledge about the satellites, instruments, and experiments, an orbital propagator, and some spherical geometry. At the time of this research, no commercial software that would calculate footprint overlap was known, so the following algorithm was developed.

To keep the problem simple, the only case considered here is that for which all satellites follow each other in the same orbit (see Figure 5-2). For this case, the relative instrument positions, and therefore the effective experiment FOVs, remain constant with time. Note that modifying this to handle the other two cases would merely require an iteration of this methodology for every time step. In these cases, the relative instrument positions change every time step, necessitating a recalculation of the effective experiment FOV every time step.

#### 5.3.1 Definitions

The effective experiment FOV can be measured in a variety of different ways. One potential metric is the “height” of the FOV, where “height” is defined to be in the down range.
direction. Since the FOVs for successive time steps overlap, the FOV height is less important than the height at the outside edges. If the height is very small, it might indicate that the assumption that the entire FOV can be scanned by each instrument is no longer valid.

Another metric is area. However, because the effective experiment FOV remains constant as the satellites move forward in their orbits, the time integrated FOV is actually a swath across the ground (see Figure 5-4). Thus, the important metric is the width of this swath, which is the width of the effective experiment FOV for one time step. The algorithm described below calculates width, however, this software implementation also calculates area for comparison.

![Figure 5-4: The time integrated instrument (or experiment) FOV is a swath across the globe.](image)

In general, FOVs for science instruments tend to be circular. However, one of the instruments in the sample problem described in Section 4.3 has a longer down range than cross track FOV, which is an elliptical FOV. It is assumed that all the instrument FOVs can be approximated by ellipses.

One modeling approximation employed in the algorithm is that each instrument FOV is projected onto a flat earth instead of a spherical one (see Figure 5-5). For an instrument orbiting at 700 km with a horizon to horizon FOV, this results in a 3.3% error in the size of the FOV. Note that the size of the error scales with the size of the FOV, so a smaller
instrument FOV has less error. In addition, the image near the edge of the FOV gets elongated and distorted. The flat earth assumption that errs on the small side for the total FOV area can be thought of as excluding this lower quality data.

![Diagram](image)

Figure 5-5: The actual FOV for an instrument and its linear approximation.

The danger in this assumption comes not in approximating one instrument FOV, but in the calculation of the overlap. Care must be taken that the actual overlap is preserved, not linearly approximated. If two satellites are separated by an earth angle, $\alpha$, then the separation distance from the center of one FOV to the center of the next must remain $R_e \times \alpha$ in the model, not the linear approximation of that (see Figure 5-6). Note that the following algorithm does not depend on this assumption. It merely makes the calculations much clearer and easier to follow.

### 5.3.2 Two Instrument Case

Given these assumptions, the sizes of both the instruments FOVs, the separation between the instruments, and some algebra, the effective experiment FOV between two instruments can be calculated. Let $x$ be in the cross track direction, and $y$ be in the down range direction (see Figure 5-7). $a_n$ and $b_n$ are the semi-major and semi-minor axes of the $n^{th}$ ellipse corresponding to the FOV for the $n^{th}$ instrument. $c_n$ is the down range distance from its center to some reference point.
Figure 5-6: While each instrument FOV can be approximated as a flat surface, care must be taken to preserve the actual center to center separation distance.
Figure 5-7: Illustration of the coordinate system used to calculate the effective experiment FOV. $x$ is in the cross track direction, $y$ is in the down range direction. The elliptical instrument FOVs are aligned so that the semi-major axis, $a$, is in the $x$ direction while the semi-minor axis, $b$, is in the $y$ direction.

Solving the following series of equations which define the locus of points for instruments 1 and 2:

\[
\frac{x^2}{a_1^2} + \frac{(y - c_1)^2}{b_1^2} = 1 \\
\frac{x^2}{a_2^2} + \frac{(y - c_2)^2}{b_2^2} = 1
\]

gives the coordinates of the intersection points. Note that since the ellipses are always centered at $x = 0$, the problem is right/left symmetric. Only one point needs to be found.

\[
x = \sqrt{a_1^2(1 - \frac{(y - c_1)^2}{b_1^2})} \\
q = -\frac{1}{2}(B + \text{sign}(B)\sqrt{B^2 - 4AC}) \\
y = \frac{q}{A}; C/q
\]
where:

\[
A = \left( \frac{1}{b^2} - \frac{a_1^2}{a_2^2 b_2^2} \right)
\]

\[
B = \left( \frac{2c_1 a_1^2}{a_2^2 b_1^2} - \frac{2c_2}{b_2^2} \right)
\]

\[
C = \left( \frac{a_1^2}{a_2^2} - \frac{a_1^2 c_1^2}{a_2^2 b_1^2} + \frac{c_2^2}{b_2^2} - 1 \right)
\]

The effective experiment FOV width is simply 2x.

For the case with more than two ellipses, a different algorithm must be used. Like the two instrument algorithm, it uses the geometry of the instrument FOV ellipses to calculate the effective experiment FOV. Since the calculations are lengthy, the algorithm is described in Appendix A.

### 5.3.3 Effect of Separation Angle on Effective Experiment FOV

Given a particular experiment (i.e., the instruments and their FOVs are fixed), there are two factors that influence the effective experiment FOV: the order of the instruments in orbit; and their separation distance (or angle). The order of the instruments translates directly from the assignment of instruments to satellites. Remember that all the satellites are following each other in the same orbital plane. Instruments on the first satellite will be separated by \(1\alpha \) from the instruments on the second satellite but by \(2\alpha \) from the instruments on the third satellite. The order that results in the largest effective experiment FOV for any one experiment is based on ranking the instruments according to the size of their FOVs; the instruments with the smallest FOV in the center of the clustellation and the instrument with the largest FOV on the outside. In a system with more than one experiment, it is possible that each experiment's optimal order will be different. In this case, the mission designer has to determine to optimal order for the overall mission.

As stated before, the maximum effective experiment FOV occurs when there is no separation between instruments. Given that there has to be some separation angle, the above algorithms can help determine the effect of non-zero separation angles. Assume that each satellite is separated by the same angle, \(\alpha\), from the next satellite in the clustellation.
Figure 5-8 shows the effective experiment FOV width and area as a function of the separation angle. The axes values are left out because they are highly dependent on the actual instrument FOVs. The general shape of the graphs remains constant from case to case. The separation angle spans the range from $\alpha = 0^\circ$ to the angle at which none of the instrument FOVs overlap. For a clustellation in low earth orbit (LEO) whose two instruments can view from horizon to horizon, $\alpha_{\text{max}}$ is approximately $24^\circ$ (dependent on the exact altitude). The effective experiment FOV size spans the range from the size of the smallest instrument FOV to zero. A limb to limb imaging instrument in LEO has an instrument FOV width of approximately $2330\text{km}$, and an area of approximately $4,263,800\text{km}^2$.

![Graph 1: FOV width as a function of separation angle.]

![Graph 2: FOV area as a function of separation angle.]

Figure 5-8: Effective experiment FOV width and area as functions of separation angle.

Note that while the satellites are each separated from their neighbors by an angle $\alpha$, this says nothing about the instrument distribution. If there are several instruments on the same satellite, they have a zero separation angle relative to each other.

Note from Figure 5-8 that the effective experiment FOV width stays constant at the maximum value until a threshold value is reached. Then it drops rapidly to zero. The specific
threshold value and how fast it drops to zero are functions of the instrument order and FOV sizes. Note that in the two ellipse case, the threshold corresponds to the physical point when the semi-major axis of one FOV passes out of the other FOV (see configuration C in Figure 5-9).

Figure 5-9: A: The two FOVs are completely overlapped. Both width and area are maximum. B: Area has decreased although width is still at a maximum. C: Width is now decreasing as well. D: There is no overlap. Both width and area are zero.

The effective experiment area, on the other hand, starts dropping off as soon as the smallest FOV is not entirely part of the effective instrument FOV any longer. It does not, however, drop off quite as rapidly.

It is the mission designer’s job to consider these effects of separation angle and decide on an appropriate mission separation angle. Smaller separation angles still allow for better co-registration, even if the model shows a threshold angle that is acceptable. However, there also other factors that impact the choice of separation angle. Some communications systems dictate that the satellites need to be a certain distance apart so their communications streams can be differentiated. The satellites also need to be far enough apart that they are not in danger of physical contact.

5.4 Case Study: EOS AM-2

The nominal architecture for EOS AM-2 is to have all five instruments on one large satellite (see Section 4.3). Since there is enough power and CPU capability onboard for all the instruments, the only resource that is limited is data rate. The logistics of achieving spatial co-registration of the data whenever possible is not an issue since all the instruments are
always imaging the same place. However, cost and reliability advantages are expected if the 
eos AM-2 mission is executed by a clustellation of smaller satellites.

The main disadvantage of the clustellation is that all the instrument FOVs are no longer 
co-located. It is possible that the clustellation architecture will produce less experiment 
data than the nominal architecture. While the scheduler employs no information about 
cost or reliability, it can help determine the relative amount of science data from candi-
date architectures. To illustrate the use of the scheduler in supporting the design of the 
clustellation, three different case studies are examined in the following sections:

- Nominal: 1 satellite (see Chapter 4)
- Case I: 5 satellites
- Case II: 3 satellites
- Case III: 5 satellites, different experiment set

Since it is cheaper to design one satellite bus and build it several times than to design 
several different satellite buses, all the satellites in any one case are assumed to have the 
exact same bus independent of which instruments are onboard. Therefore the amount of 
each resource available on each satellite is constant within each case study but can vary 
from case to case. The instruments themselves are the same as in the nominal 
eos AM-2 
architecture, and therefore are independent of satellite or case. Specifically, this implies 
that none of the resource usages or FOVs vary. None of the constraints from the nominal 
case are altered either. Tables 5.1 and 5.2 show the instrument FOVs and resource usages 
respectively [3, 31]. Note that EOSP and CERES have the same FOV.

Table 5.1: Down range (DR) and cross track (CT) dimensions for each of the instrument 
FOVs.

<table>
<thead>
<tr>
<th></th>
<th>CERES</th>
<th>EOSP</th>
<th>LATI</th>
<th>MISR</th>
<th>MODIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT Swath (km)</td>
<td>5400</td>
<td>5400</td>
<td>185</td>
<td>360</td>
<td>2330</td>
</tr>
<tr>
<td>DR Height (km)</td>
<td>5400</td>
<td>5400</td>
<td>185</td>
<td>5400</td>
<td>2330</td>
</tr>
</tbody>
</table>

The distribution of instruments on satellites, choice of separation angles, and satellite re-
source availabilities all vary from case to case and are derived in the following sections.
Table 5.2: Resource usages for each instrument. The two values listed for MISR denote its global and local modes respectively.

<table>
<thead>
<tr>
<th></th>
<th>CERES</th>
<th>EOSP</th>
<th>LATI</th>
<th>MISR</th>
<th>MODIS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rate Ltd</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7/8</td>
<td>7</td>
</tr>
<tr>
<td>CPU</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7/8</td>
<td>7</td>
</tr>
<tr>
<td><strong>Volume Ltd</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power (W)</td>
<td>58</td>
<td>19</td>
<td>81</td>
<td>88/96.8</td>
<td>170</td>
</tr>
<tr>
<td>Data Rate (Mb)</td>
<td>0.1</td>
<td>0.44</td>
<td>60</td>
<td>8.2/9.0</td>
<td>6.2</td>
</tr>
</tbody>
</table>

5.4.1 Case I: 5 Satellites

In this case, each of the 5 EOS AM-2 instruments is on an individual satellite. The instrument FOVs and resource usages are as in the nominal case. The satellite resource availabilities are set so that the schedule generated in the nominal case is possible in this case as well. In reality, these (unknown) values will be set by the satellite bus designers and will probably not match those of the nominal case.

Each satellite has enough of each rate limited resource and volume limited power onboard so that all the instruments can be on all the time. The volume limited power requirement is driven by MODIS, which requires more power than any of the other instruments. Data rate, on the other hand, is limited. The nominal case has enough data rate capacity so that all the instruments but LATI can be on all the time. LATI can be on 67% of the time (see Figure 4-5). The data rate for this case is set to the same rate: 67% of the LATI data rate requirement for one renewal interval. Note that 67% of the required LATI data rate is still more than 100% for any other instrument.

Table 5.3 shows the resource availabilities for this case. Note that since the instrument usages are so uneven (MODIS needing more power than any other instrument and LATI needing more data rate), the cost savings achieved in developing one satellite bus and building five copies might be overshadowed by the excess resource requirements placed on three of those buses. It might be cheaper to design two different buses. In an actual design situation, a trade should be done to determine the most cost effective set up.

While the experiments and their values, $V_e$, stay the same as in the nominal case, the
Table 5.3: Available satellite resource amounts and their renewal rates for the nominal case and Case I.

<table>
<thead>
<tr>
<th></th>
<th>Renewal Interval</th>
<th>Total Nominal</th>
<th>Total Case I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rate Ltd</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>36</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>36</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td><strong>Volume Ltd</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power (W)</td>
<td>1 orbit</td>
<td>7704</td>
<td>3400</td>
</tr>
<tr>
<td>Data Rate (Mb)</td>
<td>1 orbit</td>
<td>1040</td>
<td>803</td>
</tr>
</tbody>
</table>

values are adjusted by a modulator, $M_e$. As discussed earlier, the value of this modulator is a function of two factors: the instrument order; and the separation angle.

The instrument order that generates the largest effective experiment FOVs for the overall mission is: EOSP; MISR; LATI; MODIS; CERES. This puts the instrument with the smallest FOV to be in the center (LATI). Note that this order is not optimal for experiments 1 and 2. They require MISR, MODIS, and EOSP or CERES respectively. However, if these four instruments are placed in the order CERES, MISR, MODIS, and EOSP, the effective experiment FOV for any experiment that requires LATI and another instrument is extremely small. Overall, the order with LATI in the middle is better for the mission.

Two separation angles are examined. An angle of $5^\circ$ is below the threshold where the effective experiment FOV width starts dropping for any of the instruments (see Figure 5-8). Given this and the above resource allocations, this case should generate the exact same schedule as the nominal case. An angle of $11^\circ$ is used for comparison. This is the angle that gives $50\%$ effective experiment FOV width. Table 5.4 shows the experiment values and modulations for each experiment.

Both separation angles in Case I give the same optimal schedule as the nominal case (see Figure 4-5). The effective experiment values ($V_e \times M_e$) are not sufficiently different relative to each other to change the priorities of the scheduler. The science values do, however, change with the separation angle (see Table 5.5). As expected, the $5^\circ$ separation angle case is identical to the nominal. The $11^\circ$ separation angle case has a schedule that is worth less value because the modulator reduces the value of each experiment.
Table 5.4: The experiment and modulator values for the nominal case, and Case I with a 5° and 11° separation angle ($M_e$ based on FOV width).

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Required Instruments</th>
<th>Value</th>
<th>$M_e$</th>
<th>$M_o$</th>
<th>$M_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nominal</td>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5°</td>
<td>11°</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>EOSP, MISR, MODIS</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
</tr>
<tr>
<td>2</td>
<td>CERES, MISR, MODIS</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
</tr>
<tr>
<td>3</td>
<td>MODIS</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>LATI, MISR, MODIS</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>5</td>
<td>LATI, MODIS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>6</td>
<td>MODIS</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>LATI</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>LATI, MODIS</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 5.5: Science value of the nominal and each Case I schedule given in science points.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\alpha$</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal</td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>I</td>
<td>5</td>
<td>420</td>
</tr>
<tr>
<td>I</td>
<td>11</td>
<td>383</td>
</tr>
</tbody>
</table>
The modulators can also be calculated using effective experiment FOV area instead of width. Table 5.6 shows their values using the same separation angles as above.

Table 5.6: The experiment and modulator values for the nominal case, and Case I with a 5° and 11° separation angle ($M_e$ based on FOV area).

<table>
<thead>
<tr>
<th>Experiments</th>
<th>Required Instruments</th>
<th>Value</th>
<th>$M_e$</th>
<th>$M_e$</th>
<th>$M_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>EOSP, MISR, MODIS</td>
<td>5</td>
<td>1</td>
<td>0.90</td>
<td>0.39</td>
</tr>
<tr>
<td>2</td>
<td>CERES, MISR, MODIS</td>
<td>4</td>
<td>1</td>
<td>0.86</td>
<td>0.07</td>
</tr>
<tr>
<td>3</td>
<td>MODIS</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>LATI, MISR, MODIS</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.12</td>
</tr>
<tr>
<td>5</td>
<td>LATI, MODIS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.12</td>
</tr>
<tr>
<td>6</td>
<td>MODIS</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>LATI</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>LATI, MODIS</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Using these effective experiment values, the schedule is still the same as in the nominal case. However, the science value of the schedule changes even more drastically with separation angle (see Table 5.7).

Table 5.7: Science value of the nominal and each Case I schedule in science points ($M_e$ based on FOV area).

<table>
<thead>
<tr>
<th>Case</th>
<th>$\alpha$</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal</td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>I</td>
<td>5</td>
<td>406</td>
</tr>
<tr>
<td>I</td>
<td>11</td>
<td>252</td>
</tr>
</tbody>
</table>

5.4.2 Case II: 3 Satellites

All of the data produced by MISR can be duplicated with several MODIS instruments if they can be tilted to give different sun angles. The premise is that designing and building three MODIS instruments could be cheaper than designing and building one MODIS and one MISR (development costs being a large portion of the cost of an instrument). Accordingly, this case has three satellites with six instruments: CERES, EOSP, LATI, and three MODISs. One MODIS is nadir pointing as before. Any experiment that calls for MODIS will use this
instrument (a nadir pointing instrument has better resolution than a tilted instrument). The other two MODISs are tilted at -50° and +50° respectively. Any experiment that calls for MISR will use all three MODISs.

To ensure that co-registration is possible, the three MODISs are set up so they tilt towards each other. If all three instruments are on the same satellite, there is very little co-registration due to the large tilt angle. So each MODIS is on a separate satellite. The other three instruments are assigned one to each satellite, with LATI (the instrument with the smallest FOV) in the center. The satellite order and instrument allocation is then:

- Satellite 1: MODIS1 (-50° tilt), CERES
- Satellite 2: MODIS2 (0° tilt), LATI
- Satellite 3: MODIS3 (+50° tilt), EOSP

The tilted MODISs also affect the separation angle calculations. The original calculations described in Section 5.3 assume all the instruments are nadir pointing, so some alterations have to be made.

The instruments are only tilted in the down range direction, not cross track. As a result, the cross track FOV is only slightly enlarged, and will be assumed to stay constant. The down range FOV, however, is elongated and the center is no longer at the sub satellite point. The calculation for the new down range FOV length is shown in Appendix B. Table 5.8 shows the resulting instrument FOV dimensions and the positions of the center of the FOV relative to the sub satellite point. The latter is given only in the down range direction since the FOV is assumed to be centered in the cross track direction.

<table>
<thead>
<tr>
<th></th>
<th>CERES</th>
<th>EOSP</th>
<th>LATI</th>
<th>MODIS1</th>
<th>MODIS2</th>
<th>MODIS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT Swath (km)</td>
<td>5400</td>
<td>5400</td>
<td>185</td>
<td>2330</td>
<td>2330</td>
<td>2330</td>
</tr>
<tr>
<td>DR Height (km)</td>
<td>5400</td>
<td>5400</td>
<td>185</td>
<td>2762</td>
<td>2330</td>
<td>2762</td>
</tr>
<tr>
<td>Center (km)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1319</td>
<td>0</td>
<td>+1319</td>
</tr>
</tbody>
</table>

An interesting point is that for tilted instruments, a zero separation angle no longer insures the optimum co-registration. For the three MODISs, the optimum is at $\alpha = 12^\circ$ (see
Figure 5-10. Schedules were generated for separation angles of 5° and 12°. Table 5.9 shows the experiment values and modulators.

![Figure 5-10: Effective experiment FOV width as a function of separation angle for an experiment requiring only the three tilted MODISs.](image)

The resource availabilities on each satellite are set up in the same manner as in Case I. This time there is 4615W of power per orbit and 803Mb per orbit. Note that in this case, satellite 2, carrying MODIS2 and LATI, is the driver for both power and data rate.

Table 5.10 lists the resulting schedule value in science points. The resulting schedule, shown in Figure 5-11, is identical for both separation angles. It looks different than the nominal schedule because of the redefined instruments. Keeping in mind that MODIS2 replaced MODIS and MODIS1, MODIS2, and MODIS3 replaced MISR, the schedule does, in fact, match the nominal.
Table 5.9: The experiment definitions, values and modulators for the nominal case, and Case II with a $5^\circ$ and $12^\circ$ separation angle.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Required Instruments</th>
<th>Value</th>
<th>$M_e$</th>
<th>$M_o$</th>
<th>$M_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>EOSP, MODIS1, MODIS2, MODIS3</td>
<td>5</td>
<td>1</td>
<td>0.83</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CERES, MODIS1, MODIS2, MODIS3</td>
<td>4</td>
<td>1</td>
<td>0.83</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>MODIS2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>LATI, MODIS1, MODIS2, MODIS3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>LATI, MODIS2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>MODIS2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>LATI</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>LATI, MODIS2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.10: Science value of each schedule (in science points) in Case II.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\alpha$</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal</td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>II</td>
<td>5</td>
<td>401</td>
</tr>
<tr>
<td>II</td>
<td>12</td>
<td>420</td>
</tr>
</tbody>
</table>

Figure 5-11: Schedule for one orbit of Case II.
5.4.3 Case III: 5 Satellites, Different Experiment Set

Since we have seen that the schedule does not change with the separation angle or mission architecture, this particular set of experiments has a very stable optimal schedule. This is an excellent situation for the mission scientists, but not so good when a proof of concept for the scheduler is desired. This case is based on Case I, but is altered to make it a less stable problem with more interesting resultant schedules.

Using the original five EOS AM-2 instruments, there is one instrument per satellite. Since part of the reason the schedules for the nominal case and Cases I and II are the same is the fact that there is no lack of resources, each satellite will have 50% of the required power and date rate: 1700W and 600Mb per orbit.

The main factor contributing to the similarity of the schedules is that while LATI and MODIS are the most expensive instruments in terms of resources, they are also the most valuable. In addition, the experiment definitions and values are designed so that MODIS will have a 100% duty cycle. For this new case, however, MODIS is not on all the time and the experiment set is somewhat redefined. Table 5.11 shows the new set of experiments and their values. Note that the schedule value for the 5° separation angle Case III will not match the nominal value even if the schedules are identical since not all the experiment modulators have values of unity.

Table 5.11: The experiment definitions, values, and modulators for for the nominal case, and Case III with a 5° and 11° separation angle. The double experiment value denotes a day and a night value for the experiment.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Required Instruments</th>
<th>Value</th>
<th>$M_x$</th>
<th>$M_x$</th>
<th>$M_x$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nominal</td>
<td>III</td>
<td>III</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5°</td>
<td>11°</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>EOSP, MISR, MODIS</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
</tr>
<tr>
<td>2</td>
<td>CERES, MISR, MODIS</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
</tr>
<tr>
<td>3</td>
<td>EOSP, MISR</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>LATI, MISR, MODIS</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>5</td>
<td>CERES, EOSP</td>
<td>5</td>
<td>1</td>
<td></td>
<td>0.91</td>
</tr>
<tr>
<td>6</td>
<td>MODIS</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>LATI</td>
<td>1/5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>LATI, MODIS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Since the experiment definitions and values have changed, the schedule for the nominal case with all five instruments on one satellite must also be regenerated. Table 5.12 shows the schedule values for Case III. Figures 5-12, 5-13, and 5-14 show the schedules themselves. Note that Case III is much more sensitive to changes in the input data than the first two. The schedule does not match the nominal one, and it varies with separation angle.

![Schedule Diagram]

**Figure 5-12:** Schedule for one orbit of the nominal case using the new experiments and resource amounts from Case III.

**Table 5.12:** Science value of each schedule (in science points) in Case III.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\alpha$</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal</td>
<td></td>
<td>401</td>
</tr>
<tr>
<td>III</td>
<td>5</td>
<td>308</td>
</tr>
<tr>
<td>III</td>
<td>12</td>
<td>236</td>
</tr>
</tbody>
</table>
Figure 5-13: Schedule for one orbit of Case III with a separation angle of 5°.

Figure 5-14: Schedule for one orbit of Case III with a separation angle of 11°.
Chapter 6

Uses for an Automated Scheduler

Formulations for schedulers for several different classes of satellites have been presented in the preceding chapters. The goal has been to develop a scheduler that runs quickly and is generic enough to be easily adapted for different satellites. As noted in those earlier chapters, the scheduler does run quickly. Naturally the run time is dependent on the time step size and schedule horizon, but the schedulers in the preceding chapters generally find the optimal schedule for a week in under a minute of real time. This chapter includes a discussion of how generic the scheduler is, and presents some additional uses for such an automated scheduler.

6.1 Mission Scheduling

The most basic use for a scheduler is to schedule baseline satellite missions. If a mission is at all complicated, perhaps with several instruments, experiments, satellites, time step types, and data taking modes, the number of possible schedules increases exponentially. A human scheduler can have trouble creating a feasible schedule, much less an optimal one. This scheduler has been developed to perform that job quickly and efficiently.

By developing good schedules quickly, the automated scheduler realizes gains in human productivity as well as in science data that would be lost with a less efficient schedule. The cost of the scheduler is its development and implementation for a specific mission. If the
development and implementation costs are too great, they will overshadow any savings in scheduling. One way to cut these costs is to develop generic schedulers that can be easily adapted to any mission. This way the development cost is a non-recurring cost and the implementation cost is kept to a minimum.

As stated in Chapter 1, the development of a generic scheduler that can automatically create a schedule for any conceivable satellite is beyond the scope of this thesis. Instead, it has been the objective of this thesis to develop a scheduler for “classes” of satellites that share many of the same characteristics. That way the scheduler can take advantage of the inherent structure of the satellite mission and design. The difficulty of modifying the scheduler to adapt to any other class is directly related to how different the classes are.

Despite the large design difference between the spin stabilized science satellite class of Chapter 3 and the 3-axis stabilized science satellite class of Chapter 4, the two classes have been shown to be quite similar in terms of scheduling. Some of the variables do get redefined. For instance, a time step of data in the class of Chapter 3 includes data that will later be discarded because the instrument was not facing the earth. An equivalent time step of data in the class of Chapter 4 contains entirely useful data. However, the fundamental structure of the scheduler remains the same. It is conjectured that the same scheduler might also generalize easily to communications missions, which also require constant, global coverage. In its current form, the scheduler might not be applicable to a military surveillance or deep space comet tracking missions. Missions of this type require targeting a certain area or object in preference to other points.

Adapting the scheduler to another satellite of the same class, however, is fairly easy. Adding or subtracting constraints or instruments involves changing the prototype constraint matrix for every time step time that is affected. An example is the set of three scenarios for the case studies in Chapter 5. For each scenario, the number of instruments changed, as did some of the constraints, but the basic problem remained the same. These changes were implemented easily.

Changes that do not alter the shape of the constraint matrix, but merely alter values in the matrix, are trivial to implement. These are changes such as an increase in an instrument’s resource usage, or a decrease in a resource’s total availability. This type of change is what
was needed in Chapter 5 to change the separation angle in each of the scenarios. The only change to the scheduler was a change in the input data. These types of alterations can be made in minutes.

6.2 Rescheduling

Being able to develop the optimal schedule for a mission quickly and efficiently, especially for complicated missions, is extremely useful. However, in missions where the time line or instrument constraints are constant from one scheduling horizon to the next, the scheduler is only used once. The real benefit from an automated scheduler is seen in situations where rescheduling is necessary.

Rescheduling is often required as a result of stochastic events in the environment. The nominal schedule is defined as the optimal schedule assuming that events proceed exactly according to plan, which is often not the case. For instance, although each orbit has been modeled as a constant ratio of day to night, this ratio changes during the year as the earth tilts towards and away from the sun. This not only changes when certain instruments can be on if they are subject to day/night constraints, but also affects the total amount of available power.

An unpredicted event, such as an unforeseen data taking opportunity, will also change the problem so that the nominal schedule is far from optimal. An instrument which only measures solar flares will only be scheduled when there is a strong likelihood that there will be a flare. If there is an unexpected flare up, it would be beneficial to turn the instrument. In such cases, rescheduling is, if not necessary, then at least advisable.

Since most unexpected events alter parameters such as the time line or the total amount of resource available, but not the fundamental satellite configuration or design, rescheduling with the scheduler is easy and, above all, very fast. Thus, over the entire life of the satellite, less valuable data will be lost and less manpower will be needed than would be if the rescheduling were done by hand.
6.3 Failure Analysis

A failure can be classified as an “unexpected event” requiring immediate rescheduling. The faster the new schedule can be generated and implemented, the lower the amount of valuable data that is lost. Clearly there will be some reduction in data due to the failure even if a new optimal schedule is generated, but not as much as if the original schedule were continued. As an example, the amounts of science data lost due to several different failures were calculated for the nominal EOS AM-2 mission in Section 4.3 and are described below.

Failures that result in either a total loss of satellite functionality (such as a total failure of the communications system) or a quick total recovery (such as a single event upset) are not very interesting to schedule. The result is that either the mission is over, or the nominal schedule is still optimal. Instead, this analysis concentrates on failures that result in some partial loss of capability, either temporarily or permanently.

The scheduler is used to generate a nominal schedule for a 24 hour period. This is the optimal schedule assuming no failures of any kind and is used as reference. Then various failures are injected.

Many satellites have safety precautions that send the satellite into sleep mode if any failures are encountered. Once a sleep mode is detected, the failure must be diagnosed by mission control and a new schedule, which is optimal for the new situation, created and uplinked. For this discussion, it will be assumed that the failure cannot be fixed and the relative effects of continuing to use the old schedule and uplinking a new schedule will be examined.

When the old schedule is uplinked, it can no longer be carried out exactly. For instance, if there has been a 25% power loss, the instruments will run out of power before the schedule is complete, and any remaining scheduled experiments will not generate data. The new schedule, on the other hand, takes these failures into consideration. Table 6.1 shows the difference in science generated by each schedule for various failures. The first four example failures; a partial battery failure, a partial ROM failure, an instrument failure, and an instrument adjustment; are all assumed to occur midway between ground contacts and result in the satellite being in safe mode until the next contact. The last failure, missing a ground contact for some reason that does not imply a permanent failure of the communications
system, does not result in the satellite entering safe mode, merely that the data storage capacity is not renewed for one renewal period. Note that it is always better to uplink the new schedule tailored to the latest situation.

Table 6.1: Science generated by the old optimal schedule and the new optimal schedule due to various failures (given as a percent of nominal).

<table>
<thead>
<tr>
<th>Failure</th>
<th>Old Schedule</th>
<th>New Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>25% Power Loss</td>
<td>73%</td>
<td>85%</td>
</tr>
<tr>
<td>25% Memory Loss</td>
<td>67%</td>
<td>90%</td>
</tr>
<tr>
<td>LATI Fails</td>
<td>66%</td>
<td>66%</td>
</tr>
<tr>
<td>MODIS needs</td>
<td>67%</td>
<td>81%</td>
</tr>
<tr>
<td>100% More Power</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Contact Missed</td>
<td>50%</td>
<td>78%</td>
</tr>
</tbody>
</table>

If all the expected failures and the frequency of their occurrence were known for a mission, the total expected amount of science lost during a mission could be calculated. This might be a way to help compare different architectures for a mission, such as the EOS AM-2 nominal and the first two proposed scenarios.

Other failure analyses can be done with the scheduler as well. For instance, suppose a failure occurs and the scientists are told that the satellite can be brought out of safe mode with a range of percentages of the original resources. Assume that the higher that percentage, the more likely a second failure will occur and damage the satellite even further. The scientist must now weigh the probability of incurring further damage against the value gained from the availability of a higher percentage of the resource. With the scheduler, it is possible to calculate the resultant value (as a percentage of the value for the nominal schedule) as a function of different levels of available resource. Instead of having to guess the value from the chosen level of post failure resources, the scientist knows the exact value (assuming no other unexpected events happen) and, in fact, the schedule for the new resource level.
6.4 Sensitivity Analysis and Design Trade-offs

Often not all of the scheduling parameters are known exactly before launch. Estimates are available and more accurate values will be determined as the satellite starts its operations. This means that there is a certain amount of uncertainty to the scheduling problem. The scheduler, however, can generate schedules for all the values that are considered likely so that the satellite operators have a better understanding of how the system is likely to behave over a range of values. As more accurate information becomes available, better schedules can be generated.

The scheduler can also be helpful in the design phase of the satellite. There are models the designers can use to estimate the cost and reliability of different design concepts, but no easy way to estimate the value of a mission. If enough of the inputs can be estimated with reasonable accuracy, the scheduler can be used to perform parametric studies of the value of the gathered data versus various parameters on the satellite. An example might be setting the life span of the instruments. Generally, the life span of the entire satellite is set by the mission requirements. However, if the instruments are not on all the time, their life span does not need to be as long. With the scheduler, the designers can figure out how long the life span for each individual instrument should nominally be.

To demonstrate the scheduler’s abilities further, a sensitivity study of total available volume limited power on the overall TERRIERS schedule value is shown here. Note that only the amount of volume limited power is changing for this analysis. Rate limited power and all the power usages are held constant.

Performing a sensitivity study on discrete points is straightforward. The user changes the scheduler input to reflect the new power level. The scheduler then produces the optimal schedule for the new configuration, and the associated schedule value. Figure 6-1 shows some results for 50% to 160% nominal power.

If the variables were not constrained to be integers, these points could be extrapolated to straight lines. However, since increasing the power a fraction of the amount needed to turn on an additional instrument will not increase the number of instruments that are on, the graph is actually a step function, not a straight line. This step function can be generated
by studying the underlying structure of the problem. With the resource usages and totals used in this problem, volume limited power is always the only limiting resource (this is also confirmed by the non-zero slack values for every resource except power from the scheduling IP). An IP problem can be created using only volume limited power whose optimal value will be the optimal schedule value for the full problem. It is, however, important to note that this method produces only the schedule value, not the schedule itself (while power might be the limiting resource most of the time, another resource that this analysis does not take into account might be important when there is far too much power). The IP then has the form:

Objective Function:

\[ \text{maximize } (5p + 7n + 4d) \]

Subject to:

\[ 0.0747p + 0.396n + 0.60d \leq \text{ total power} \]

\[ 0 \leq p \leq 8N \]
\[ 0 \leq n \leq 16N \]
\[ 0 \leq d \leq 8N \]

\(P, n,\) and \(d\) are integers representing the number of photometer-time steps, night spectrometer-time steps, and day spectrometer-time steps respectively. \(N\) is the number of renewal periods in the schedule duration. As there are only four night spectrometers and four night time steps in a two orbit period, \(n\) cannot be greater than \(16N\). Similarly there are only two photometers and four time steps in which they can be active in a two orbit period, so \(p\) cannot be greater than \(8N\). Also, there is only one day spectrometer and eight time steps in which it can be active, so \(d\) must be less than \(8N\).

Figure 6-2 shows the graph of the maximum amount of science that can be generated as a function of the available power. Note that the practical limits are 0% power, at which all the instruments are off and 160% power at which all the instruments are on for the entire renewal period. Another interesting item to note is the break in the graph at 85.6% power. Because the day spectrometer uses more power and has a lesser value than any of the night instruments, all the night instrument-time steps are activated first and then the day instrument-time steps if there is any power left over (see Lemma 6.1). At 85.6% power, all the night instruments and none of the day instruments are on.

**Lemma 6.1** If \(\exists\) two instruments, \(i\) and \(j\), with science value \(v_i\) and \(v_j\) respectively and resource usage \(a_i\) and \(a_j\) respectively, and

\[ v_i > v_j \]
\[ a_i \leq a_j \]

then if instrument \(j\) is used in any optimal solution, \(x^*\), instrument \(i\) will also be used. Specifically, if \(x_j^* > 0\) then \(x_i^* > 0\).
Figure 6-2: Schedule value as a function of available power as generated by theory (both given as a percentage of nominal). The discrete points represent the value of optimal schedules generated for various power levels.

**Proof** The optimal solution is a solution to the problem of the form

\[
\begin{align*}
\text{maximize} & \quad v'x \\
\text{subject to} & \quad a'x \leq b \\
& \quad 0 \leq x \leq \hat{x}
\end{align*}
\]

as was shown previously. Assume that \( x^* \) is an optimal solution with \( x_i^* = 0, x_j^* > 0 \). Construct a new solution, \( \hat{x} \), that is defined as \( \hat{x}_i = x_i^* + 1 \) and \( \hat{x}_j = x_j^* - 1 \). Then

\[
\begin{align*}
a'\hat{x} &= a'x^* + a_i - a_j \leq a'x^* \leq b \\
v'\hat{x} &= v'x^* + v_i - v_j > v'x^*
\end{align*}
\]

Therefore \( \hat{x} \) is feasible and has a higher science value which contradicts the optimality of \( x^* \).
6.5 The Ideal Level of Automation

One purpose of a computerized scheduler is to help automate scheduling. It makes intuitive sense that increasing the level of automation decreases the cost and increases the reliability of the system (see Figure 6-3). The intuition is that, in comparison to automated systems, human operators are expensive, comparatively slow, and prone to mistakes in repetitive tasks. However, as the tasks become more complex and less repetitive, the cost for a reliable automated system rises (again, see Figure 6-3). The open question is, what is the ideal automation level for a given task? A recent study has begun to address this issue [40].

![Figure 6-3: A presumed curve of life cycle cost versus the level of automation.](image)

In addition to cost and reliability, there is a third important factor in determining the ideal level of automation: mission value. Ultimately, the products of a mission (in the science satellite case, the data) must justify its cost. A mission that produces very little value for a small cost might still be less cost effective than an expensive mission that produces an enormous amount of value. As the level of automation can affect the value of a system as well as the cost and reliability, an objective comparison between scheduling systems would be extremely useful.

While the scheduler currently contains no cost or reliability information, it does provide a metric with which to compare different levels of automation. The scheduler can produce the optimum values that can be gained from the same mission with different levels of automation. This information, coupled with the cost and reliability information for each level of automation, can show which level is ideal for the mission. Note that cost could also be included in the scheduler either as another constraint or in the objective function.
6.5.1 Levels of Automation

There is no single accepted definition of automation, much less a set of definitions for levels of automation. One definition that applies directly to satellite systems was developed by Schwarz [40]. A completely automated system is one in which the satellite can operate on its own, without ground intervention; a system with no automation is one in which the human operator takes care of all functions. Naturally, there are levels in between these two extremes.

Regardless of the level of automation, some processor must perform the scheduling task. There are three possible processors in a satellite system: a human operator; the ground station computer; or the satellite onboard computer. It is also possible that the task be performed jointly by two processors. The sensible combinations are (1) the satellite computer and the ground station computer, and (2) the ground station computer and the human operator. Note that in each case, the processor mentioned second is considered less automated.

The amount that each of the respective processors contributes to the performance of a task can be used to define the level of automation. While there are gradations between not automated and fully automated, they will not be considered here. Instead three discrete levels of automation are defined and analyzed: “no automation” corresponding to the human processor doing the scheduling; “full ground automation” corresponding to the ground station computer doing the scheduling; and “full onboard automation” corresponding to the satellite computer doing the scheduling.

6.5.2 Comparing Levels of Automation

The following discussion assumes that the schedule developed is the optimal schedule, no matter which processor developed it (i.e., assuming no failures, the schedule is identical in all three cases). The difference, then, is in the time to reschedule. Furthermore, the speed of the schedulers is assumed to be as follows. The onboard scheduler can operate in real time (i.e., scheduling takes a negligible amount of time). The ground scheduler is not aware of a failure until the next ground contact. Once the contact occurs, however, it can generate
a schedule fast enough that it can be uplinked during the same contact period. The human scheduler is also not aware of the failure until the next ground contact, and is assumed to require the time between two contacts to generate a new schedule.

Since these classes of satellites all have missions that require continuous global coverage, there will be few unforeseen viewing opportunities. Thus the majority of the unexpected events will be failures, which is what this analysis will focus on. Also note that only certain failures are interesting in this context. Failures that prematurely end the satellite’s usefulness also end the need for scheduling and completely recoverable failures can utilize the nominal schedule. The interesting failures are those that result in some degradation in performance, such as losing one instrument entirely or losing some percentage of the total available power. Failures that do not result in any permanent degradation but that take time from which to recover are also included in this analysis.

Given these assumptions, it is possible to determine how much science is lost due to a failure for each of the three levels of automation that are considered. Remember that “science” is a metric for the amount and value of the data taken by the instruments. The onboard scheduler takes a negligible amount of time to reschedule, so the only science that is lost is due to the degraded state of the satellite (or it can be assumed that one time step’s worth of data is lost). The ground scheduler can reschedule during the contact period, so in addition to the science lost due to the satellite degradation is the science lost between when the failure happened and the next ground contact. This is because the optimal schedule has changed due to the failure but the nominal schedule is still being used. The human scheduler loses the same science as the ground scheduler, but also runs the nominal schedule instead of the new schedule for an additional contact interval due to longer time to reschedule.

For example, the amount of science lost in a 24 hour period can be calculated for the TERRIERS satellite for each of the three different levels of automation. For the no automation case, the nominal schedule is optimal for the first 6 hours (the instruments generate 312 science points). Then the failure happens and the payload goes into sleep mode. After another six hours, this information is downlinked to the human scheduler (0 science points). The scheduler examines the data, issues the restart command and generates the new optimal schedule. However, the human scheduler cannot do this fast enough to uplink it on the same contact, so it must wait until the next contact, 12 hours later (the payload is still in sleep
mode, so 0 science points are generated). This results in a total schedule value of 312 science points, or 25% of the science points generated by the nominal schedule. Note that after the next contact, the payload follows the new optimal schedule, generating a reduced amount of science. However, the scheduler for each level of automation will continue this way, so only a 24 hour period was examined.

For full ground automation, the scenario is much the same. The nominal schedule is used for the first six hours (312 science points). Then the failure happens and the payload goes into sleep mode until the next contact (6 hours, generating 0 science points). At that point the scheduler issues the restart command and generates the new optimal schedule. Unlike the human scheduler, however, the ground automated scheduler can do this fast enough to uplink it on the same contact. So for the next 12 hours, the new optimal schedule is run (532 science points). This results in a total schedule value of 844 science points, or 68% of nominal.

For full onboard automation, the scenario is slightly different. The nominal schedule is still used for the first 6 hours before the failure occurs (312 science points), however as soon as the failure happens, the scheduler knows about it. This is the obvious advantage to a real time, onboard scheduler. The scheduler can immediately issue the restart command and generate the new optimal schedule. There is a possible loss of one or two time steps worth of science as the scheduler runs, but this is insignificant for this analysis. This schedule is then run for the next 18 hours (840 science points), resulting in a total schedule of value 1152 science points or 89% of nominal.

The same type of calculation was run for several different failures: the power loss described above, a computer problem resulting in a loss of data storage capability (25% loss of memory), a failure in one of the night spectrometers (total loss of the one instrument), and a failure in the photon collection mechanism for one spectrometer resulting in a need for more power (100% more power needed for one spectrometer). Table 6.2 shows the amount of science generated by each scheduler assuming various failures.

There are also conceivable failures that do not fit this time line. For instance, when a contact is missed due to atmospheric problems, it is unreasonable to assume that the entire payload should go into sleep mode. Instead, the payload keeps functioning until it runs out
Table 6.2: Amount of science generated, given that various failures have occurred (as a percentage of nominal).

<table>
<thead>
<tr>
<th>Failure</th>
<th>No Automation</th>
<th>Ground Automation</th>
<th>Onboard Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>25% Power Loss</td>
<td>25%</td>
<td>68%</td>
<td>89%</td>
</tr>
<tr>
<td>25% Memory Loss</td>
<td>25%</td>
<td>71%</td>
<td>95%</td>
</tr>
<tr>
<td>1 Night Spectrometer Fails</td>
<td>25%</td>
<td>70%</td>
<td>92%</td>
</tr>
<tr>
<td>1 Night Spectrometer Needs 100% More Power</td>
<td>25%</td>
<td>71%</td>
<td>95%</td>
</tr>
<tr>
<td>1 Contact Missed</td>
<td>56%</td>
<td>56%</td>
<td>84%</td>
</tr>
</tbody>
</table>

of data storage. Whether it overwrites previous data or not is irrelevant. The important point is that only so much data can be stored at once. For both the no automation and full ground automation cases, the payload keeps functioning on the nominal schedule until there is no more free memory. So the payload generates 12 hours plus 1.3 orbits of data, resulting in a schedule worth 704 science points or 56% of nominal.

In the full onboard automation case, the payload also generates the initial 12 hours of science (624 science points). As soon as the failure happens, though, the scheduler generates a new optimal schedule that uses the remaining memory in a much more efficient fashion. This results in 12 hours of reduced data taking (420 science points), for a total schedule value of 1044 science points or 84% of nominal.

Clearly, the real time, onboard scheduler has a large advantage over the other two. However, remember that the scheduler has no information about the cost or reliability of the overall mission. For a small mission like TERRIERS, it is probably not worth the cost and time to develop an onboard scheduler. The satellite does not have a computer onboard that can support such a scheduler anyway. Instead the most benefit would probably be obtained from a full ground automated scheduler.

The main reason the real time scheduler has such a large advantage is the long ground contact interval. If something goes wrong, it takes an average of 6 hours for the ground to find out about it, much less do anything about it. For a system like EOS AM-2 with a ground contact every 100 minutes, the onboard scheduler still has the same advantage but
to a lesser degree.

An estimate of the total amount of value lost by each scheduler during the satellite's life (or some other benchmark time interval) can then be obtained by multiplying each failure by the number of times it is estimated to happen in the time interval and summing over all failures. This gives a satellite designer a concrete number for the value gained from each scheduler. When this information is coupled with the cost and reliability of each scheduler, a design trade can be made as to which system will be most cost effective.
Chapter 7

Conclusions and Future Work

7.1 Overview

Good schedules increase the value of a system without increasing the cost by using the available resources efficiently. However, generating the best schedule, especially for a complicated system, is not a trivial process and the associated manpower costs required for manually generating such a schedule can easily overrun any savings the schedule might garner. Automating this process increases efficiency and decreases personnel costs.

Moreover, an automated scheduler that is adaptable to other systems provides additional cost savings. Since the main cost of an automated scheduler is its development, reusing the scheduler from one satellite program to another would make this a one time only cost. In addition, the scheduler can make rescheduling a much less time consuming task by adapting to changing conditions.

In the past, satellite schedulers have typically used either a form of the envelope method or heuristics (near-optimal instead of optimal) and have been highly problem-specific. This thesis presents three automated schedulers that produce optimal schedules for three different classes of satellites: spin stabilized science satellites; 3-axis stabilized, earth observing science satellites; and constellations of 3-axis stabilized, earth observing science satellites. Each scheduler models its scheduling problem as an integer programming problem. In each
model, there is a set of experiments that can be carried out, along with their attendant constraints. Each experiment has some associated value and the goal is to find the time-ordered sequence of experiments that maximizes the overall schedule value. In addition, the schedule must honor all the constraints. These come in three categories: instrument constraints such as "instrument A cannot be on in sunlight;" resource constraints such as power; and system constraints such as "instruments A and B are not allowed to be on simultaneously."

Each scheduler is formulated for the general case of that class and then demonstrated in a case study. Note that this thesis concerns itself with modeling the problem as an integer programming (IP) problem, not with solving the IP problem. The actual solution method used in the case studies is a branch and bound method as implemented by a commercial solver. No attempt was made to tailor the solution method or optimize the code for speed or memory consumption.

7.2 Capabilities

Since each scheduler is formulated for the general case of its class, it is easily adaptable to any satellite that fits those class requirements. In fact, the majority of this adaptation effort is in gathering the data on the particular satellite so that it can be modeled properly. Once all the necessary data is in one place, implementing the scheduler is the work of a day or two. This makes it a convenient tool, especially for smaller satellites with only a few instruments onboard. These programs might not have the budget to develop their own schedulers, however, they can spend a couple of hours adapting this one to their satellite.

Once the scheduler is adapted, it takes little time to create the schedule itself. For the TERRIERS case study, a week's schedule was created in under a minute and for the EOS case studies, 100-minute long schedules were created in a couple of seconds. This speed creates new opportunities for mission control. Not only can the manpower needed to run the satellite be reduced but the system response time to a failure or unexpected opportunity can be improved. While an optimal schedule produces the greatest cost savings if all events happen exactly as anticipated, the ability to reschedule in the face of unexpected events is
more important.

The scheduler can also be used as a design and analysis tool. Other models exist for comparing cost and reliability of different architectures, but they all assume the schedule value remains constant. The scheduler, however, is based on the notion that the relative value of each schedule can be computed. Note that the scheduler contains no information about the satellite cost or reliability.

Currently, the scheduler is only intended to schedule payload events. However, there is no intrinsic difference between payload events and bus events that would prevent the scheduler from being doubly useful. Note, however, that the bus events probably need a more fine-grained schedule than the payload events, so they should govern the choice of time step size.

7.3 Limitations

This thesis concerns itself with the scheduling, not the planning of satellite missions. It is assumed that all the other planning tasks can be carried out in some efficient, expedient manner. The time it takes to reschedule, for example, is actually the scheduling time plus however long the rest of the planning system takes. If the planning system is a human operator with a poorly designed user interface, this could be substantial. Note that packaging the scheduler in an automated planning system will take some forethought but is not conceptually difficult.

For the most part, all the other assumptions in the thesis are concerned with the definitions of each class and only impact which class a satellite falls into. It is, however, assumed that relative values can be placed on one time step of data from each experiment. No serious thought has been given to how this might be accomplished. For a relatively focused mission with a handful (5-10) of experiments that are repeated again and again, like TERRIERS or EOS, the principal investigator was able to assign the values himself. However, for a wider-focused mission with many (thousands) one-time only observations, like the Hubble Space Telescope, no one person can perform this task.
7.4 Future Work

Many of the science satellite missions, like EOS and TERRIERS, are intended to take continuous measurements around the globe for some period of time to give scientists a set of baseline measurements. For such missions, the satellite needs to respond to failures in a reasonable amount of time, but there are no unexpected events that would necessitate an immediate response. So the scheduler can remain on the ground, which has two benefits: ground computers will always be a generation or two ahead of space computers so the scheduler will run faster; and there is no need to spend already limited satellite resources (power, memory, and CPU time) on scheduling.

For missions which observe unpredictable events, especially events with short life spans, however, the opposite is true. Now the scheduler must run in near real-time. For instance, a weather satellite that is tracking a tornado must be able to identify the tornado, focus on it, and report back to earth in time for the area to be evacuated. To do this, the scheduler must constantly know about the state of the environment around the satellite (whether or not an unexpected event is taking place). In this case, it is better to move the scheduling onboard the satellite so that the mission is not so dependent on the satellite-ground communications link.

Real time schedulers developed for other autonomous vehicles, like the Autonomous Mine-hunting and Mapping Technologies (AMMT) submarine [36] might find an application to satellite systems. AMMT is an hierarchical scheduler: there are different schedules for each level of detail. It can schedule in real time because the scheduling problem is broken up into small pieces that each level can solve quickly. The top level arranges the operator specified goals on a time line and has a rough approximation of their resource usages. The second level breaks each high level goal into the pieces that are carried out by each subsystem, schedules them, and collects details on how accurate the resource allocations of the top level were. The lowest level actually commands the hardware. Actual resource usages and terrain data from the sensors are passed back to the second level scheduler.

Such a scheduler could be adapted for a satellite. Each “goal” is an image that needs to be taken, while the satellite “travels” from one time step to the next. The scheduler could also control all the satellite bus functions as well.
This scheme does, however, require one central scheduler. For one weather satellite, this might be ideal. However, for a constellation of weather satellites, this forces the very communication that was avoided by putting the scheduling onboard the satellite in the first place. Now instead of having to talk to the ground to get their respective schedules, each satellite must talk to the scheduling satellite. Note that not all the satellites in a global constellation can necessarily communicate directly with all the others at any given time because there may be no direct line of sight.

Instead, the scheduling process can be broken up into individual pieces on each satellite. All the separate satellites can be modeled as players in an n-player coordinated game. They are all working toward one goal (getting the best global weather report) but are considered as separate entities. The amount of communication between the individual satellites depends on the system architecture. For now, assume that all the satellites can signal each other (either directly or via a forwarding path through another satellite). Each satellite knows the overall system value of each of its instruments, and knows how that value changes under different circumstances. For instance, if two satellites can coordinate an experiment, the overall value increases. However, if the instrument measures clouds and there are no clouds in view, then its value decreases. So the satellites communicate with each other to coordinate their strategies. The problem is that while the individual satellite’s resource levels can be predicted and accounted for, the weather is a stochastic event and moreover is not necessarily the same for any two satellites. That information must be relayed fast enough to allow for the schedule generation. Given that there is some chance the satellites’ communications will be interrupted or not occur in time, how should each satellite behave? This model can be considered a “coordinated attack problem” [21]. Some thought needs to be given to how to optimize over many time steps instead of just one, however.

7.5 Conclusion

This thesis has demonstrated that it is possible to develop an adaptable automated scheduler for a class of satellites. Moreover, it has been shown in the case studies that such a scheduler can be implemented in a practical manner for these three classes. For any spin stabilized science satellite, or 3-axis stabilized earth observing science satellite (or constellation
This scheduler is small and fast enough to be extremely useful.

It is also possible to formulate a similar scheduler for any satellite class. The only question is whether or not that scheduler would be cost effective. A satellite belongs to a certain class because of its fundamental characteristics. The most important of these characteristics to the scheduler is the form of the decision variables: what they are and how many of them there are. The three classes described here all have the same decisions: should instrument $x$ be on during time step $y$. As such they can all be considered part of a super class of satellite scheduling problems that is solvable using integer programming methods. Other classes might have different variables, such as where the instrument should be pointing in addition to whether or not it should be on. Because of the large number of decision variables this leads to, integer programming techniques are probably not practical solution techniques. Other techniques, such as the genetic algorithms of Abbott [1] might be more suitable.
Bibliography


127


[60] http://ls7pm3.gsfc.nasa.gov/objective.html
[71] http://www.urban.uiuc.edu/Courses/varkki/msProject/msproject.html
Appendix A

Effective Experiment FOV for Three or More Instruments

Section 5.3 details the algorithm for calculating the effective experiment FOV for any case involving two instruments. For the case with more than two instruments, a different algorithm must be used.

The inputs to the algorithm are the sizes and down track spacings of the ellipses. The definitions in Section 5.3 apply here, as does the coordinate system and notation shown in Figure 5-7. Note that since the geometry of the problem is right/left symmetrical, the algorithm only needs to consider half of the intersection points.

Since the effective experiment FOV could be bounded by any number from one to all the involved ellipses, the first step is to identify all the bounding ellipses. The bounding ellipses at the top center and bottom center can be found very easily. Each ellipse has two points that lie on the y axis, which given this coordinate system, correspond to the two endpoints of its minor axis (see Figure 5-7). They can be calculated by $y_n = c_n \pm b_n$.

The effective instrument FOV is bounded in y by the the minimum top minor axis endpoint, $(c_n + b_n)$, and the maximum bottom minor axis endpoint, $(c_n - b_n)$ (see Figure A-1). The ellipses these two points are part of are the top and bottom bounding ellipses.
Figure A-1: The top and bottom bounding points of the effective experiment FOV.

Note that if the top bottom minor axis endpoint is above the bottom top minor axis endpoint (see Figure A-2), then there is no overlap between the instrument FOVs at all. In this degenerate case, the effective experiment FOV is zero and there is no need to continue with the algorithm.

Figure A-2: Zero effective experiment FOV.

Once the top and bottom-most bounding ellipses have been found, the intervening bounding ellipses can also be found. Using the two ellipse algorithm, it is possible to calculate all the intersection points between the top bounding ellipse and all the other ellipses. The next bounding ellipse is the one whose intersection point is the closest in the clockwise direction (see Figure A-3). Note this technique is a modified version of a Graham’s scan [13].

Using this new ellipse as the top bounding ellipse, this procedure is repeated until the
Figure A-3: The next bounding point is the point closest to the old point in a clockwise direction.

bottom bounding ellipse had been reached. The result is a list of all the bounding ellipses and their intersection points (see Figure A-4).

Figure A-4: The finished algorithm produces a list of bounding points and ellipses.

The algorithm requires at most $\frac{n(n-1)}{2}$ two-ellipse intersection point calculations. Note that integrating over the appropriate ellipses from intersection point to intersection point gives the area of the effective experiment FOV.

Now that the outside edges of the effective experiment FOV are known, the maximum width can be calculated. Naturally if there is only one bounding ellipse (the top and bottom bounding ellipses are the same and it does not intersect with any other ellipse), the effective experiment FOV is that ellipse. The effective experiment FOV width is the major axis of that ellipse (see Figure A-5).

If there is more than one bounding ellipse, finding the effective experiment FOV width is a bit more complicated. There are two different scenarios: the maximum width is the major axis of one of the ellipses; or the maximum width is at an intersection point. To test for
Figure A-5: If no intersection points are found, the effective experiment FOV is the FOV of the center instrument.

In each case, first find the two boundary points with the largest $x$ values and the bounding ellipse that connects them. If the center of the connecting ellipse falls between the two boundary points, the effective experiment FOV width is the major axis of the ellipse (see Figure A-6).

Figure A-6: The ellipse that connects the two boundary points is centered between the two points.

If, however, the center of the bounding ellipse is either above or below the effective experiment FOV, the maximum width is twice the largest of the $x$ values (see Figure A-7).

Note that, as described here, this methodology does not adequately handle the case where more than two ellipses intersect in the same place.
Figure A-7: The ellipse that connects the two boundary points is centered either above or below the two points.
Appendix B

Tilted FOV Calculation

The original FOV calculations all assume that all the instruments are nadir pointing. Further calculations have to be done if this is not true.

If the instruments are only tilted in the down range direction, the cross track FOV is only slightly enlarged, and will be assumed to stay constant. The down range FOV, however, is elongated and the center is no longer at the sub satellite point. The new down range FOV length can be calculated by applying the law of sines (see Figure B-1).

Figure B-1: The FOV of a tilted instrument is longer than that of a nadir pointing instrument.
If $\beta$ is the tilt angle and $\eta$ is the instrument look angle:

$$\epsilon = 90 - \eta$$
$$D = 180 - \epsilon$$
$$C = 180 - (D + \beta)$$

$$\frac{\text{untilted FOV}}{\sin C} = \frac{\text{tilted FOV}}{\sin D}$$

Since the untilted FOV length is known, the tilted FOV length can be calculated. However, it is important to remember that this derivation is done assuming a flat earth. While the error in the FOV size is not all that great, care must be taken that the tilted footprint does extend past the horizon. The horizon is a function of altitude and can be easily calculated.

![Figure B-2: The tilted footprint calculated above needs to be truncated at the horizon.](image)

From Figure B-2:

$$F = |\eta - \beta|$$
$$x = H \tan F$$

where $H$ is the altitude of the satellite. Then:

if $\eta > \beta$ : actual footprint = min(tilted footprint $- x, 1/2$ horizon) + $x$

if $\eta < \beta$ : actual footprint = min(tilted footprint + $x, 1/2$ horizon) $- x$
Note that although the footprint really has an elongated and truncated shape, it is still assumed to be elliptical. It is centered at $x = 0, y = 1/2$ tilted footprint $\pm x$ depending on whether $\eta$ is greater than or less than $\beta$. 