

Task Scheduling for Satellite Based Imagery

S.A. Harrison¹, M. E. Price²

Defence Evaluation and Research Agency, Malvern

and

M. S. Philpott³

University of East Anglia, Norwich

Abstract

Observation satellites have a limited window of opportunity for imaging or taking measurements of a given target area, dictated by orbit considerations. In low Earth orbit (typically 400 to 1200km altitude), a ground object will be in view for a few minutes at most. Depending on the mode of operation of the satellite, the data-taking window may range from a few minutes down to only a small fraction of a second.

The scheduling problem then is one of achieving the maximum efficiency of use of a satellite. Traditionally, this has been achieved by teams of ground planners who write, check and recheck procedures.

Current missions are beginning to exploit the capability of automated decision support software. In some cases, automated scheduling results in not only improved efficiency, but in modes of use that would otherwise be impossible.

The principle questions of this study were:

- How many images can we expect to acquire in a single (three minute) overpass of a typical area of interest? and
- Given a lead time of between ten and fifteen minutes, can we automate the process of defining the schedule of imaging tasks?

The imaging tasks are selected according to a priority function from a larger set of imaging requests.

In order to address these issues, a specific problem was defined. This problem involves only a small number of parameters, but the principles involved are representative of many situations which could be encountered by optical or radar based Earth observing satellites. To obtain solutions we investigated a partial enumeration method. The enumeration method was able to generate optimal task schedules for the size problem we expect to encounter in practice within the lead times we expect to encounter in practice.

The success of the enumeration method for the simplified problem suggests that more sophisticated exact methods, such as branch-and-bound, will be equally successful for fuller formulations of the task scheduling problem. Hence, further work is to specify more fully the task scheduling problem and investigate branch-and-bound methods for these problems.

1. Introduction

Satellites have strict and demanding requirements for activity scheduling. They have numerous constraints, some hard, some soft, and many activities that need to take

¹ SAHARRISON@dera.gov.uk

² MEPRICE@dera.gov.uk

³ m.s.philpott@uea.ac.uk

place, often concurrently. Together with the multiplicity of planning and scheduling needs, the satellite environment, and its isolation, provide a harsh scenario with little in the way of a safety net. If something goes wrong, pressing the reset button may not be an option, with the cost of in-flight recovery or repair almost always prohibitive.

Moreover, due to their cost, Earth observing satellites are often co-funded by several agents, such as governmental organisations or commercial concerns. So, as well as physical constraints, exploitation of the Earth observing satellite must also satisfy constraints of efficiency and fairness. Efficiency means that each co-funding agent wants to maximise its return while fairness means each agent's return must be in proportion to its investment.

In general efficiency and fairness requirements are antagonistic. So, a compromise must be found. Bataille *et al.* [1] consider a number of approaches to sharing satellite resources. They conclude that no one approach dominates in all cases and that the best strategy is to present the agents with a number of alternatives, and their properties, a *protocol*, and let the agents themselves decide according to their individual and subjective preferences.

1.1 Earth Observation Satellites

Observation satellites have a limited window of opportunity for imaging or taking measurements of a given target area, dictated by orbit considerations. In low Earth orbit (typically 400 to 1200km altitude), a ground object will be in view for a few minutes at most. Depending on the mode of operation of the satellite, the data-taking window may range from a few minutes down to only a small fraction of a second.

The scheduling problem then is one of achieving the maximum efficiency of use of a satellite, while ensuring absolutely its safety, by meeting a number of constraints during operation. Traditionally, this has been achieved by teams of ground planners who write, check and recheck procedures, consisting of long lists of instructions, which are sent to the satellite. This requires long lead times, which must be traded off with the number of (expensive) ground planners, and the level of usage of the satellite.

1.2 Automated Mission Planning

Mission operations traditionally account for 40% of mission costs and are labour intensive. Hence, current missions are beginning to exploit the capability of automated decision support software, to allow efficiency and functionality improvements on the more traditional methods of pre-planning all activities. A number of approaches are reported in the literature and a number of systems have already flown on successful missions.

Bensana *et al.* [2] formulate a problem of task selection, for a multiple instrument (sensor) Earth observing satellite as an instance of Multi-Dimensional Knapsack, which is known to be *NP*-hard [5]. They compare branch-and-bound, greedy and tabu search methods on the resulting instances of Multi-Dimensional Knapsack, for single and multiple orbits, with rather inconclusive results. Their problem differs from the one presented in this paper in that task order has no effect on solution quality; whereas for the problem described in this paper task order has a significant effect on the number of tasks that can be scheduled. Hence, on the overall priority of a solution. Bensana *et al.* [3] extend their Multi-Dimensional Knapsack model to account for the uncertainties associated with cloud cover.

Muraoko *et al.* [8] describe a simple greedy approach for scheduling subsets of tasks selected on the basis of priority. Their aim was to generate a feasible schedule. However, no consideration was given to the benefits of efficient scheduling. The system was used by the Advanced Spaceborne Thermal Emission and Reflectance Radiometer (ASTER) which is due to be launched in early 2000 as part of NASA's Terra mission, formerly the Earth Observing Systems AM-1 (EOS AM-1). The European Space Agency's ERS-1 and ERS-2 mission employ a rule-based approach to the selection and scheduling of requests for satellite time.

The scheduling problems associated with both Earth observing satellites and astronomy generally requires several conflicting requirements to be met. Often the key modelling issue is how to trade-off between the different requirements. This is considered by Bataille *et al.* [1], in the context of resource sharing, and Bresina *et al.* [4,7] present an adaptive approach for obtaining high quality multi-objective schedules for fully automated, terrestrial telescopes as part of the Associate Principal Astronomer (APA) system. Sherwood *et al.* [10] describe a constraint based reasoning approach for automated planning and scheduling on NASA's Earth Orbiting 1 (EO 1) mission.

As well as scheduling when imaging requests are to be satisfied, it is necessary to determine how the requests are to be satisfied and how the raw data is then to be processed. NASA's VICAR tool is a suite of tools, for image acquisition and image processing, and framework for supporting the entire process from image acquisition through to generation of the final processed image.

The efficient operation of a satellite requires the scheduling and co-ordination of a large number of inter-dependent sub-systems subject to a wide variety of constraints. Gat [6] discusses the use of an AI planner for overall mission control accounting for sub-system failure and switching to backup systems. The system described (NewMAPP) is intended flew on the first of NASA's New Millenium mission, known as Deep Space 1 (DS1), launched in October 1998. Muscettola *et al.* [9] describe a heuristic scheduling testbed system (HSTS) project that is developing technologies for automated planning and scheduling and applying them to the generation of spacecraft commands. In particular, HSTS has been applied to science scheduling on the Cassini and Extreme UltraViolet Explorer (EUVE) missions.

1.3 Task Scheduling for Satellite Imagery

The two questions are to be addressed in this study are:

- How many images can we expect a low-Earth orbit imaging satellite to acquire in a single (three minute) overpass of a typical area of interest? and
- Given a lead time of between ten and fifteen minutes, can we automate the process of defining the schedule of imaging tasks?

The imaging tasks are selected according to a priority function from a larger set of imaging requests.

Clearly, any method that is able to address both these questions will form the basis of an automated decision support tool.

In order to address these issues, a specific problem was defined. This problem involves only a small number of parameters, but the principles involved are

representative of many situations which could be encountered by optical or radar based Earth observing satellites. To obtain solutions we investigated a partial enumeration method. The enumeration method was able to generate optimal task schedules for the size problem we expect to encounter in practice within the lead times we expect to encounter in practice.

The problem considers a satellite that overpasses an area of interest. The satellite is able to image only a small portion of the area of interest during the overpass time, in the form of a number of 10x10km blocks. Imaging is done via a synthetic aperture radar (SAR). There are a number of requests for such image blocks, which each have a numerical priority. The constraints consist of field of views of the imaging payload, and a limit on the scan rate of the imaging boresight.

1.3.1 Radar Model

The radar model considered is electronically scanned (ESCAN) in azimuth and mechanically scanned (MSCAN) in elevation. This means that the radar can be aligned instantly in the azimuth direction, but time is required to move it in the elevation direction. The ESCAN axis is aligned with the direction of motion of the satellite. It is necessary for this to be the case, because the scan rate in the MSCAN axis is insufficient to keep the radar boresight aligned with a specific area on the ground while it is being imaged.

Further constraints are on the range of angles the radar can scan over. In the MSCAN direction, there is a minimum angle and a maximum angle (both from the vertical) between which tasks can be viewed. In the ESCAN direction, there is an angle (also from the vertical), and only tasks that are within this angle (both ahead and behind the satellite) can be imaged. Understanding of this is aided by the following figures:

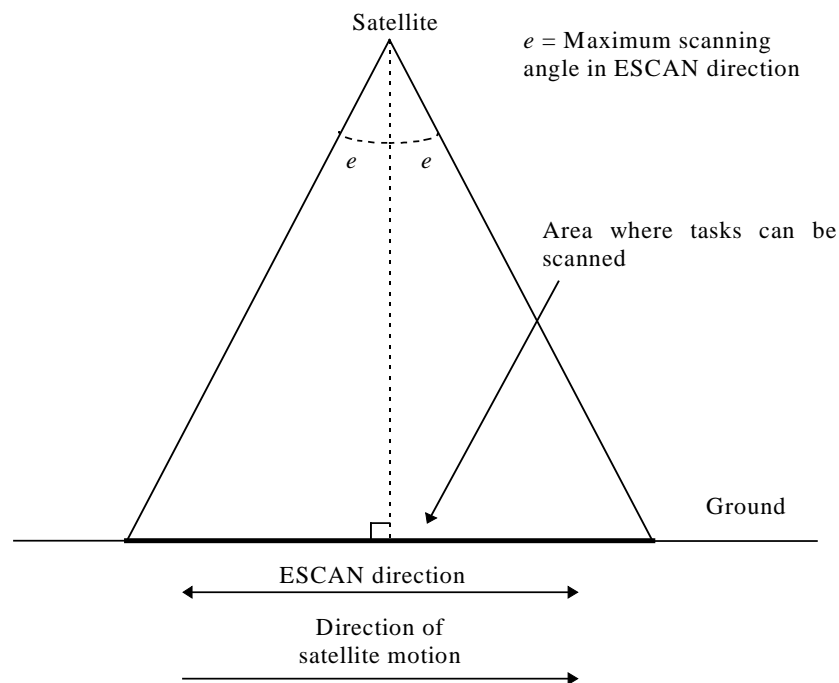


Figure 1-1 - Radar range in ESCAN direction

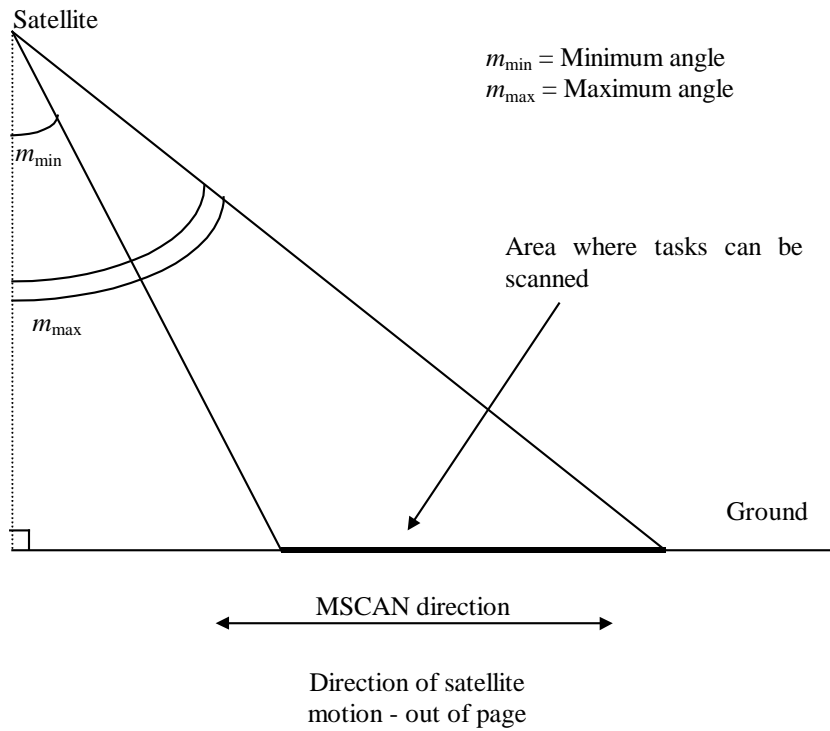


Figure 1-2 - Radar range in MSCAN direction

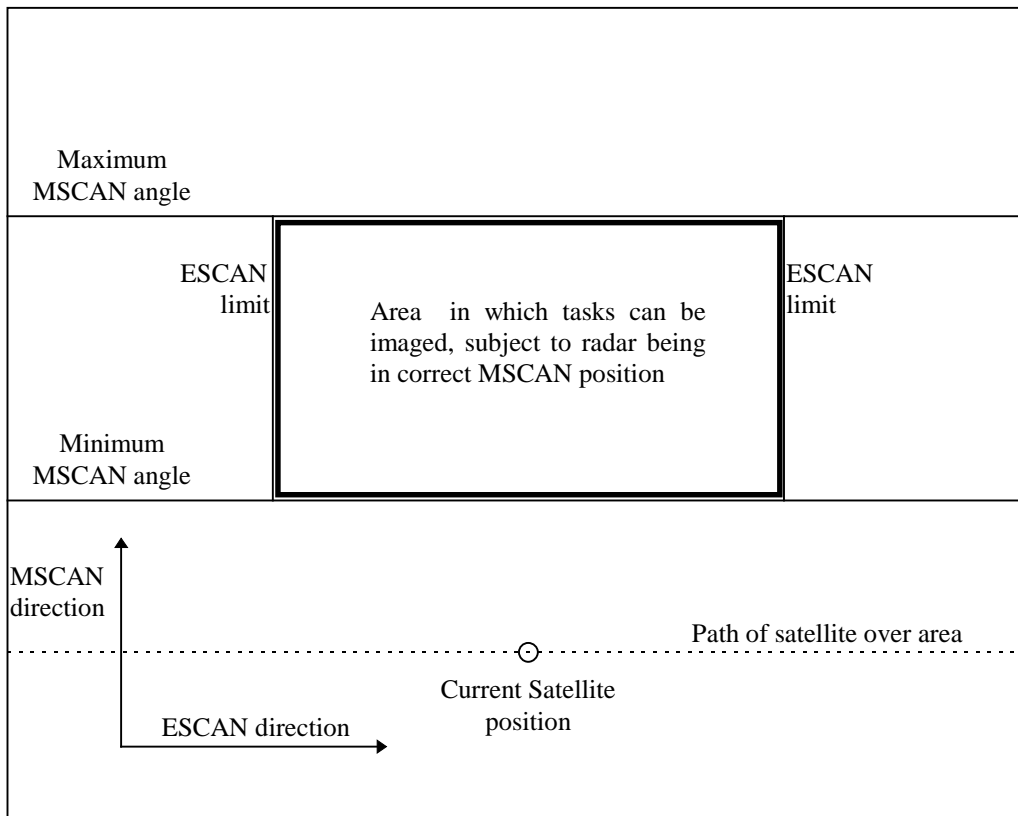


Figure 1-3 - Plan view of satellite motion and radar ranges

The satellite typically flies off to one side of the area of interest such that the minimum elevation angle aligns with the bottom edge of the area of interest. For simplicity, the satellite's orbit is assumed to lie parallel with the bottom edge of the area of interest.

However, as the satellite is constantly in motion, there will be tasks in the area highlighted in Figure 1-3 that cannot be imaged because they would take too long to acquire.

An important feature of the problem is that there will almost certainly be more tasks available than can be imaged in one pass of the satellite. This means that the schedule will be a sub-set of the requested tasks. It is also apparent that, because of the limitations of the radar in the MSCAN direction, there will be certain tasks that it is impossible to schedule, given a single pass of the satellite.

The objective of the problem is to maximise the total priority of the set of tasks that are imaged.

2. Formal Definitions

This section covers the formal definitions of the concepts outlined above. All measures of distance are in kilometres and all speeds are in kilometres per second.

2.1 Tasks

Associated with each task, t , is:

- A **number**, n : the index of the task;
- A **priority**, p : where p is a measure of how valuable t is;
- A **duration**, d : where d is the number of seconds required to image t ; and
- A **location**, l : where l is a Cartesian coordinate representing the position of the corresponding imaging area.

2.2 Satellite

Associated with the satellite, s , is:

- A **ground speed**, g : g is the speed at which the satellite covers the ground;
- An **altitude**, a : a is the constant height of the satellite above the ground;
- A **scan rate**, r : r is the rate at which the radar can be moved in the MSCAN direction, measured in degrees per second;
- An **azimuth angle**, az_θ :

az_θ is the maximum angle the satellite can view in the ESCAN direction, both ahead and behind it;

Two **elevation angles**, $m_{\theta_{\max}}$ and $m_{\theta_{\min}}$:

$m_{\theta_{\max}}$ is the maximum angle that can be viewed in the MSCAN direction, and $m_{\theta_{\min}}$ is the minimum;

An **entry point**, e_p : e_p is the coordinate at which the satellite is first available to be used;

An **exit point**, x_p : x_p is the coordinate where the satellite ceases to be available; and

An **entry time**, t_e : t_e is the time at which the satellite enters the area. Typically this will be zero.

For the scope of the current study:

- The task priorities are randomly allocated in the range 1 to 100;
- The ground speed is 6.6 kms^{-1} ;
- The altitude is 840 km;
- The scan rate is 0.1 degrees per second;
- The azimuth angle is ± 30 degrees;
- The elevation angles are 35 and 70 degrees; and
- The area of interest is 1200 x 1200 km.

Furthermore, the satellite is assumed to enter at the left hand edge of the area of interest; exit at the right hand edge; fly parallel to the bottom edge and such that the minimum elevation angle is aligned with the bottom edge of the area of interest, that is, 588 km off the bottom edge.

By a simple calculation, based on the size of the area of interest and the ground speed of the satellite, we can see that a single overpass will only last 180 seconds.

2.3 Problem Definition

Instance: We are given a satellite, s ; a set of tasks, T , and associated with each task $t \in T$ is a priority, p_t .

Objective: Find T' such that $T' \subseteq T$, and the total priority, $\langle \sum p_i \mid t_i \in T' \rangle$, is maximised

Constraints: The length of the schedule (in time) must not exceed the time the satellite is between its entry and exit points (e_p and x_p).

For a satellite at vertical position y_1 , any task in the schedule must have a y co-ordinate in the following range:

$$y_1 + a \tan(m_{\theta_{\min}}) \leq y \leq y_1 + a \tan(m_{\theta_{\max}}),$$

that is, for a task to be imaged it must lie within a band defined by the minimum and maximum elevation angles of the MSCAN axis.

Consider a satellite at horizontal position x_1 , scheduling a task t at horizontal position x_2 . Imaging for t cannot begin until the following constraint is satisfied:

$$x_2 \leq x_1 + a \tan(az_{\theta})$$

that is, a task cannot be imaged if it lies beyond the forward limit of the ESCAN axis. In such circumstances the satellite must idle until the task comes into range.

In addition, imaging for the same task can only be completed while the following constraint is true:

$$x_2 \geq x_1 - a \tan(az_{\theta}),$$

that is, a task cannot be imaged if it passes out beyond the backward limit of the ESCAN axis before imaging is complete.

2.4 Image Acquisition

The time required to acquire an image can be split into three distinct segments:

- **scan time** - time when the radar is moving to the correct elevation;
- **idle time** - time when the satellite is idle as the task is not yet in ESCAN range; and
- **image time** - the time required to acquire an image of the target.

These are specified as follows.

2.4.1 Scan Time

The scan time is calculated by the following method: First, calculate the angle that must be traversed in moving the radar in the MSCAN direction:

$$d\theta = \left| \arctan\left(\frac{x_t}{a}\right) - \arctan\left(\frac{x_M}{a}\right) \right|$$

where:

- x_t = the y co-ordinate of the task to be scanned;
- x_M = the y co-ordinate of the location on the ground currently illuminated by the radar boresight; and
- a = satellite altitude.

The scan time duration is then found by dividing this by the scan rate of the satellite:

$$d_s = \frac{d\theta}{r}$$

where

- r = MSCAN scan rate.

2.4.2 Idle Time

Once the scan time has been calculated, the position of the satellite at the end of the scanning can be ascertained as follows:

$$p_s = p_i + (d_s \cdot g)$$

where:

- p_i = initial horizontal satellite position, before scanning started; and
- g = satellite ground speed

This allows the x co-ordinate of the maximum forward location - the furthest forward the satellite can image - to be calculated:

$$x_{forward} = (a \tan(\alpha z_{\theta_{max}})) + p_s$$

where

- $\alpha z_{\theta_{max}}$ = maximum ESCAN angle.

Now, if the horizontal position of the task, x_t is greater than $x_{forward}$, some idle time is necessary. The idle time is given by:

$$d_w = \frac{(x_t - x_{forward})}{g}$$

Idle time is the time the satellite must wait until the task is within range.

2.4.3 Image Time

We define a minimum imaging time, d , as the time required to image a task that is as close to the satellite as possible, i.e. the same azimuth, and at the inner MSCAN limit. The imaging time is then scaled according to the distance the task is from the satellite when the imaging begins. This is given by the following formula:

$$d_i = \left(\frac{d}{e_{min}} \right) \left(\sqrt{|y_s - y_t|^2 + |x_s - x_t|^2} \right)$$

where

- e_{min} = the closest that a task can be in the MSCAN direction and still be imaged;
- y_s and y_t = y co-ordinates of the satellite and the task; and
- x_s and x_t = x co-ordinates of the satellite and the task (after time d_s and d_w).

The imaging time is the time it takes the satellite to complete the task. In practice this is a function of the grazing angle and the image's required resolution. For the sake of the current piece of work this has been approximated by the distance on the ground between the point the satellite is over and the current task. The minimal duration, d , is fixed at seven seconds. This results in a maximum duration of 22 seconds.

2.4.4 Total Time

The total time required for the task is now given as:

$$d_t = d_s + d_w + d_i$$

This allows the final position of the satellite to be found:

$$p_f = p_s + (d_t \cdot g)$$

Once this is known, the furthest the satellite can image behind it in the ESCAN direction is given as:

$$x_{backward} = p_f - a \tan(az_{\theta_{max}})$$

If $x_{backward}$ is greater (further right) than the horizontal position of the task, then the task cannot be imaged, as it would pass past the ESCAN tolerance before the imaging was complete.

2.4.5 Additional Idle Time

Presently, the only time the satellite is idle is when it has chosen to do a task that is beyond the forward ESCAN limit, and so it must wait for it to come into range. It may be that, as the duration of a task depends on its horizontal distance from the satellite, the satellite may want to remain idle for longer. Consider a duration, d_1 , which is the

duration of a task if it is done immediately. Also consider an idle time, i , and a second duration, d_2 , which is the duration if the task is imaged after i . The satellite should wait for i , if there exists an i such that:

$$i + d_2 < d_1$$

For the model described in this paper, the maximum additional idle per task is approximately two seconds. Therefore, it is necessary to have several tasks idling in order to generate sufficient idle time to insert an additional task. It is therefore reasonable to ignore idle time in this initial study.

3. Solution Methods

A partial enumeration method that exploited time constraints to prune the search tree was found to provide adequate performance and is described in Figure 3-1.

```

initialise remaining_schedules to empty
initialise best_solution to zero
for each task,  $t$ 
    generate a schedule starting with  $t$ 
    place schedule in remaining_schedules
end for
repeat
    remove a schedule,  $s$  from remaining_schedules
    for each task,  $t$ 
        if  $t$  is not in  $s$  and  $t$  can be added to  $s$ 
            generate new schedule  $s_{new}$  by adding  $t$  to  $s$ 
            evaluate the priority of  $s_{new}$ 
            calculate the schedule duration
            if the duration is less than the time limit
                add  $s_{new}$  to remaining_schedules
                if value of  $s_{new}$  is better than best_solution
                     $best\_solution = s_{new}$ 
                end if
            end if
        end if
    end for
until remaining_schedules is empty

```

Figure 3-1: The Schedule Enumeration Algorithm

Potentially, the maximum number of possible schedules for a given problem is $n!$, where n is the total number of tasks. Although the theoretical maximum number of possible solutions is very large the actual number may be significantly less than this.

The various time constraints mean that in general there will be insufficient time to schedule all the requested tasks. This will impose an upper limit on the number of tasks that can appear in the schedule. If the maximum number of tasks is m , the number of possibilities is reduced to $n!/(n-m)!$. Also, if a schedule is created with p tasks, and no further tasks can be added to it, then the $(n-p)!$ ways to complete the schedule can be discarded. This observation is employed to significantly reduce the size of the enumeration tree. It is this constraint pruning that, as we shall see, is key to the effectiveness of the enumeration method.

Consider a problem with 50 tasks. $50!$ is roughly 10^{64} , but if there is a maximum of six tasks that can appear in a schedule then there are at most 10^{10} schedules to consider which is far more acceptable.

For this reason, an enumeration algorithm will be developed. This will simply enumerate all the possible solutions and pick the best one. It is outlined in Figure 3-1 where the *value* of a schedule is the total priority of all its tasks.

4. Empirical Results

In this section we present the results obtained for the enumeration algorithm on a number of randomly generated, but representative data sets.

Separate studies suggest that we do expect to have more than twenty requests to consider at any one time. However, to more fully exercise the enumeration method and to more fully answer the questions posed we consider data sets containing up to fifty requests, although there are no circumstances where we will be able to satisfy fifty requests in a single overpass.

Even in the most favourable circumstances, where tasks are evenly distributed along the bottom edge, so that they can all be imaged with a minimum duration of seven seconds, a simple calculation shows that we can satisfy at most twenty five tasks in a single three minute overpass.

4.1 Summary of the Data Sets Used

The first table summarises the data sets used for the initial set of experiments. For each data set it lists the number of tasks available (total tasks) and the optimal solution: total priority and number of tasks.

There are two classes of data set considered. Uniform data sets where the tasks are uniformly distributed over the area of interest. These data sets are distinguished by the prefix *uni*.

Problem	Total Tasks	Optimal Priority	Optimal Tasks	Problem	Total Tasks	Optimal Priority	Optimal Tasks
uni.10a	10	229	5	uni.40a	40	590	9
uni.10b	10	222	5	uni.40b	40	650	8
uni.10c	10	227	5	uni.40c	40	584	8
uni.10d	10	461	6	uni.40d	40	635	9
uni.10e	10	240	5	uni.40e	40	484	10
uni.20a	20	439	7	uni.50a	50	574	8
uni.20b	20	436	5	uni.50b	50	567	8
uni.20c	20	499	7	uni.50c	50	649	10
uni.20d	20	390	7	uni.50d	50	774	9
uni.20e	20	404	7	uni.50e	50	633	7
uni.30a	30	449	8	cluster.5-5a	28	420	6
uni.30b	30	480	6	cluster.5-5b	34	443	5
uni.30c	30	572	7	cluster.5-5c	28	333	5
uni.30d	30	469	8	cluster.5-5d	28	411	7
uni.30e	30	427	7	cluster.5-5e	16	299	4

Table 1: Summary of Data Sets

In the second class of problems, the tasks were centred on a number of *cluster points*. The number of these points was given by the user, as was the mean number of tasks at each cluster point. The points were generated from a uniform distribution, as above. The tasks were then generated from a Normal (Gaussian) distribution, with the co-

ordinate of the cluster point acting as the mean. The clustered data sets considered each have five cluster points with tasks Normally distributed around these.

4.2 Results

The table in this section describes the convergence times for the enumeration method.

The solution times marked with an asterisk (*) indicate experiments done on a slightly more powerful computer, that is, a machine with more memory and a slightly greater processor speed (100 MHz rather than 75 MHz).

Problem	Time ms	Problem	Time ms
uni.10a	110	uni.40a	*1,322,210
uni.10b	90	uni.40b	*501,410
uni.10c	110	uni.40c	*1,952,280
uni.10d	520	uni.40d	*320,580
uni.10e	90	uni.40e	*1,362,710
uni.20a	9,150	uni.50a	5,380,050
uni.20b	10,240	uni.50b	*97,184
uni.20c	12,040	uni.50c	*3,616,070
uni.20d	12,080	uni.50d	⁴
uni.20e	15,800	uni.50e	*1,923,420
uni.30a	256,500	cluster.5-5a	83,250
uni.30b	48,490	cluster.5-5b	10,240
uni.30c	109,730	cluster.5-5c	2,540
uni.30d	88,320	cluster.5-5d	63,370
uni.30e	243,550	cluster.5-5e	4,930

Table 2: Results

The enumeration method can enumerate all the possible tasks in less than the required ten to fifteen minute lead time for problem sizes up to thirty. For problems larger than this the time requirement of enumeration means the method is impractical. It should be noted however, that the requirement is to task up to twenty tasks - a requirement that is met.

4.3 Pictorial Representation of Task Schedules

But what do the task schedules look like? Figure 4-1 and Figure 4-2 represent solutions generated for data sets *uni.10a* and *uni.20a* respectively.

Each figure encompasses the entire area of interest. All the tasks are plotted, with their associated priority. The optimal schedule is represented by the path taken by the radar's boresight across the ground with the start and end points indicated by the letters S and E respectively. Only the scheduled tasks lie on the path of the boresight. The satellite's path is parallel to the bottom of the area of interest and lies approximately 600 km to the "south" of the area of interest.

If we decompose the motion of the boresight into azimuth and elevation for each solution; we observe that the elevation component of the motion tends not to change

⁴ Insufficient computing power

direction. For problem *uni.10a* the elevation component scans downwards with only a small element of backtracking. However, for problem *uni.20a* the elevation component scans upwards without backtracking.

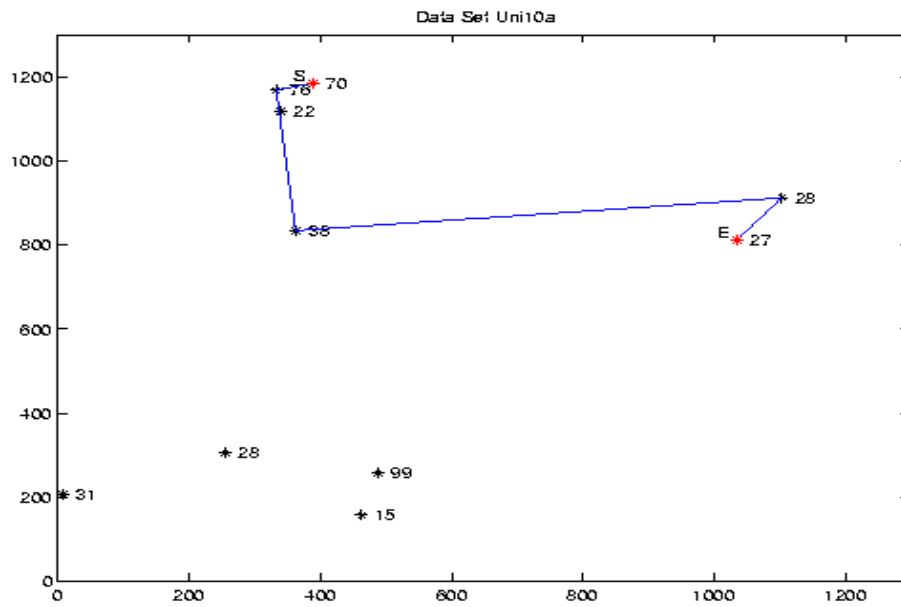


Figure 4-1: Schedules for problem *uni.10a*.

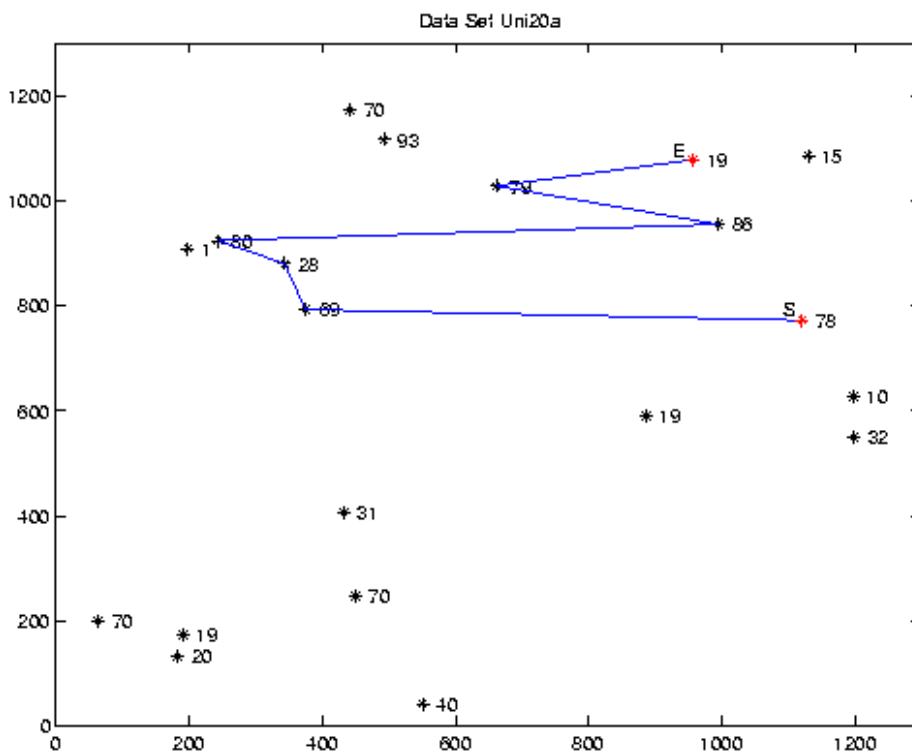


Figure 4-2: Schedules for problem *uni.20a*.

It is easily seen that by minimising the backtracking in the elevation component we minimise the amount of time that is spent aligning the radar's boresight in the MSCAN direction. However, as we can see from Figure 4-1 that it is not always possible to completely eliminate backtracking in elevation.

The task schedule in Figure 4-2 displays another interesting feature. The satellite waits until one of the right hand most tasks comes within the area it can image and does this task first and then sweeps backwards to image tasks on the extreme left hand side. The satellite is able to do this because it is electronically scanned in the azimuth axis. Hence, it is able to sweep across its entire range instantaneously.

5. Conclusions

In this paper we have considered a simplified, but realistic, formulation of a task scheduling problem associated with Earth observing satellites. We have considered two questions:

- How many images can we expect a low-Earth orbit imaging satellite to acquire in a single (three minute) overpass of a typical area of interest? and
- Given a lead time of between ten and fifteen minutes, can we automate the process of defining the schedule of imaging tasks?

In order to address these questions a simple enumeration method has been developed.

Data sets containing up to fifty imaging requests have been considered. This exceeds the numbers of requests we expect to encounter in practice.

The enumeration method has been shown to be able to generate optimal task schedules for the largest data sets in less than twelve minutes. Hence providing answers to the first question and demonstrating that it is possible to automate the task scheduling within the lead times required.

The results have shown that, on average, we cannot expect to satisfy more than ten imaging requests in a single overpass.

The work reported in this paper forms part of an initial feasibility study. Hence, the results are only preliminary but already they show promise.

On the basis of these results it is recommended that more fully specified models of the task scheduling problem are formulated and that more sophisticated optimisation methods are investigated. The aim being to develop automated decision making tools that are able to generate optimal task schedules rapidly.

6. References

1. N.G. Bataille, M. Lemaître and G. Verfaillie. Efficiency and Fairness when Sharing the Use of a Satellite. *ISAIRAS99*, 1999.
2. E. Bensana, G. Verfaillie, J. C. Agnèse, N. G. Bataille and D. Blumstein. *Exact and Inexact Methods for the Daily Management of an Earth Observation Satellite*, *SpaceOps96*, 1996.
3. E. Bensana, G. Verfaillie, C. Michelon-Edery and N. G. Bataille. Dealing with Uncertainty when Managing an Earth Observation Satellite, *ISAIRAS99*, 1999.

4. J. L. Bresina, R. A. Morris and W. R. Edginton. Optimising Observation Scheduling Objectives. *Proceedings of the 1997 NASA Workshop on Planning and Scheduling for Space*, 1997.
5. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.
6. E. Gat. News From The Trenches: An Overview of Unmanned Spacecraft for AI Researchers. *AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, 1996.
7. R. A. Morris, J. L. Bresina and S. M. Rodgers. Automatic Generation of Heuristics for Scheduling. *Proceedings of the 15th IJCAI*, 1997.
8. H. Muraoka, R. H. Cohen, T. Ohno and N. Doi. ASTER Observation Scheduling Algorithm. *SpaceOps98*, 1998.
9. N. Muscettola, B. Pell, O. Hansson and S. Mohan. Automating Mission Scheduling for Space-Based Observatories in *Robotic Telescopes: Current Capabilities, Present Developments, and Future Prospects for Automated Astronomy*, G. W. Henry and J. A. Eaton (editors), Astronomical Society of the Pacific, Provo, UT, ASP Conference Series No. 79, 1995.
10. R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien and A. Fukunaga. Using ASPEN to Automate EO-1 Activity Planning. *Proceedings of the 1998 IEEE Aerospace Conference*, Aspen, CO, 1998.