Planning and Scheduling of an Agile EOS Combining On-ground and On-board Decisions

Jufang Li, Yan Chi
School of Management, Hunan International Economics University, Changsha, 410205, China
lijf@hieu.edu.cn

Abstract. The Chinese new generation of agile Earth Observing Satellites (EOS) has more freedom and more choices on when and how to make an observation, which brings great challenges on planning and scheduling operations due to more complex and detailed mission dynamics constraints. This paper proposes a decision framework combining both on-board and on-ground part to solve the new problem. The on-ground part makes initial operation plans periodically based on estimated mission dynamics. The on-board part will revise the initial plan according to actual mission dynamics when there is no opportunity to communicate with the ground. The main management and control process is divided into four steps and the planning and scheduling problem is tackled in an integrative way. We also explain details of the rule-based heuristic algorithms which are efficient and fitful for complex constraints and limited on-board computing capability. The framework is applied to a simulation system which aims to optimize the satellite designing process. Results of experiments show that the framework and methods are reasonable and feasible.

1. Introduction
Traditionally, operational plan of a satellite is completely made on the ground ahead of time and then transmitted to the satellite by a ground station. It relies heavily on the geographical layout of the ground station network. When there are not enough globally dispersed ground stations, there may be long periods in which the ground cannot communicate with the satellite. This means that the satellite won’t have any opportunity to react to emergency tasks or unexpected situations. As a result, users are required to submit their observation requests days in advance and changes are usually not permitted after submission. This often causes a long delay for a user to retrieve his required data after he send the request [1]. Even worse is that many images retrieved are of less value due to clouds or other interferences.

The traditional way of making operational plan is a reasonable option for the old generation of satellites since they have little freedom to adjust their field of view. However, the new generation of agile earth observing satellite (AEOS) can move its line of sight more freely along roll, pitch, or yaw axis. That will enable it to observe a specific ground target from theoretically numerous view angles corresponding to different positions and ephemeris times. Thus an AEOS does have many chances to avoid cloud or other bad situations by choosing appropriate opportunities [2]. But what is an appropriate opportunity always depends on the real-time status of the satellite and the environments. Unfortunately, long-term or mid-term weather forecast is not accurate enough and detail status of an AEOS is quite difficult to be properly predicted.
On-board autonomy can certainly help an AEOS to perform better by scheduling tasks at the right moment when more accurate information can be got. It would finally benefit an AEOS by improving the quantity of valid images taken and shortening user’s waiting time [3]. But achieving entire on-board autonomy is difficult and the performance may be compromised due to limited on-board computing capability. Besides, people responsible for management of a satellite may have strong will to decide the satellite’s working plan. Therefore, to make decisions combining on-ground and on-board resources is a reasonable and necessary approach.

The AEOS considered in this paper has the capability to change its view angle fast along pitch and roll axis. We propose a framework to distribute planning and scheduling functions requiring different computing capabilities among the satellite and the ground management department. In fact, the ground department performs in a traditional way, gathering observation requests from different users at least one day ahead of time. Then all the opportunities of observing a given target and the corresponding time windows are calculated, as well as the required satellite attitudes. An initial plan is conceived using forecasted meteorological information and estimated mission dynamics, such as data storage, energy consumption or increase, etc. Manual intervention is permitted in this phase. The initial plan is uploaded to the satellite right before execution. Afterwards, the on-board part will make on-line decision to repair the initial plan in condition of finding big deviations from expected situations in order to avoid infeasible plan. We also explain the rules used for the heuristic planning and scheduling algorithm in some detail. The framework and algorithms have been verified in a simulation-based satellite design case.

2. The planning and scheduling problem
Planning and scheduling of earth observing satellite is defined here as: based on comprehensive considerations of satellite abilities and observing requirements, assign satellite resources to multiple might-in-conflict tasks, and determine specific activities and their begin and end times to fulfill the selected tasks, with an aim to eliminate conflicts and maximally satisfy users’ requirements.

For traditional non-agile earth observing satellite, alternatives to complete a specific task are limited. Especially within one pass of a satellite, the possible start and end time for watching a ground target could be fixed on beforehand. Planning and scheduling is more like a pure resource assignment problem allowing job rejection. The plan generated is mainly about which target will be observed. Such a planning and scheduling problem is often modeled as a common problem, such as a knapsack problem [2][4], a packing problem [5], a single-machine scheduling problem [6], or a network problem [7]. Algorithms for these models include constraint programming [2][8], greedy algorithms [2][5], dynamic programming [2][7], and meta-heuristics [5][9][10].

In almost all these theoretical literatures, some practical constraints, like energy consumption and replenishment, were neglected. For non-agile satellites, these constraints might be taken into account later when the plan will be transformed into executable commands. In that period, detailed operations such as turning on or turning off a camera, electricity power control, attitude control, and so on, will be added. Such a sequential process was feasible, because energy or data storage variance could be reasonably estimated in advance for non-agile satellite without causing many conflicts later. In some cases, energy constraint was considered as longest working time or so [11].

But for an AEOS, its rolling ability enables the satellite to observe wider areas at both sides of its ground track, while pitching ability enables it to observe a ground target in a much longer time window. Theoretically speaking, there will be numerous ways to watch a target from different view angles, with different attitude maneuver and energy consumption demands influenced by their precedent and subsequent observing activities. This makes it never feasible to use some constraints like longest working time during one pass or else, because it is quite hard to estimate energy consumption or replenishment and attitude maneuver times in advance without big error [12]. However, energy must be considered as a constraint, since frequent attitude maneuver will consume a lot of energy. Besides, the satellite will change attitude as a whole instead of only swing the on-board camera. So the solar panel will also change its direction, rather than always has opportunity to get
largest amount of electric energy charged. Thus planning and scheduling of the AEOS must explicitly consider energy and attitude factors, which means all operations consuming or generating energy or changing satellite attitude must be deliberately taken into account in a synchronized way when making a plan [13].

Usually customer requests can be pre-processed into candidate payload activities, mainly observation activities and download activities. For traditional non-agile satellite, these are the activities to be planned and scheduled. But for the AEOS, as mentioned before, more detailed operations need to be considered synchronously to ensure a feasible plan.

We mainly consider the following kinds of operations: imaging, attitude / antenna maneuvering, data downloading, memory erasing, sun-pointing, geocentric pointing. These operations are the lowest level of a plan that will be technically discussed in this paper. With such a plan, the final command sequence understandable by the satellite can be easily generated by some auxiliary or translating processes. To make it clear, we tried to describe the problem more formally as follows.

2.1. Decision variables
For convenience, we first define some symbols.

- \( H \) is the end time of a plan horizon starting from 0.
- \( J = \{J_1, J_2, \ldots, J_n\} \) is the set of observation tasks that are in correspondence with user requests, including single strip, jointing strips and stereoscopic strip, of which \( J_j \) denotes task \( j \). \( J \) has a revenue \( c_j \) associated with it.
- \( A_j = \{a_{j1}, a_{j2}, \ldots, a_{j\nu_j}\} \) is the set of basic operations essentially to complete task \( J_j \). The number of these operations is \( \nu_j \).
- \( \hat{A} = \{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p\} \) is the set of auxiliary operations not directly associated with \( J_j \), such as attitude or antenna maneuvering, data erasing, sun-pointing and geocentric-pointing. The number of these operations is \( \hat{\nu} \).
- \( A = \hat{A} + \bigcup_{j \in \{1,2,\ldots,\nu\}} A_j \) is thus the set of all the operations. \( a_i \) denotes the \( i_{\text{th}} \) operation belonging to \( A \). The number of operations in \( A \) is \( V \).
- \( R' \) is the set of master resources including the satellite and available ground stations.
- \( \bar{R} \) is the set of all the subordinate resources, such as platform, payload, data storage, battery, antenna of the satellite, and antennas of ground stations.
- \( R^s \) is the set of reservoir type of resources, such as data storage and battery. \( R^s \subset \bar{R} \).
- \( R = \bar{R} + R' \) is the set of all the resources.
- \( Q_r \) is the maximum capacity of a resource. \( r \in R \).
- \( L_r(t) \) is the accumulated amount of occupation, consumption or regeneration of resource \( r \) at time \( t \).
- \( O_{ir} \) is the set of all the opportunities for operation \( i \) on resource \( r \).
- \( TW_k \) is the time window related with opportunity \( k \), \( TW_k = [lb_k, ub_k] \).
- \( d_{ir}^k \) is the duration of operation \( i \) completed by resource \( r \), \( r \in R_i \). \( R_i \) is the set of alternative resources that can complete \( i \).
- \( t_{ir}^k \) is the transition time needed from operation \( i \) to operation \( i' \) on resource \( r \). This transition time can be calculated by a known function of \( i, i' \) and \( r \).
- \( E_r \) is a pair of operations that might be executed sequentially on resource \( r \). If \( i \neq i' \in V \) meet the condition of \( st(i) + d_{ir}^k + d_{i'r}^k \leq st(i') \), then \( (a_i, a_{i'}) \in E_r \). It can be seen as an edge of a directed graph related to \( r \).

The decision variables considered are thus defined as follows.

- \( X_j (J_j \in J) \) is a Boolean variable indicating whether a task \( j \) appears in the final plan, 1 if yes, 0 if no.
- \( Y_i (a_i \in A) \) is a Boolean variable indicating whether an operation \( i \) appears in the final plan, 1 if yes, 0 if no.
- \( CO_i \) is a discrete variable representing the opportunity chose by operation \( i \) in the final plan.
\( st(i) \) is a discretized continuous variable representing the start time point of operation \( i \) in the final plan. \( st(i) \in TW_{CO_i} \).

2.2. Constraints
Using the formerly defined symbols, we can explain the constraints in our problem as the following equations.

\[
X_j = \prod_{i \in \{1,2,...,n\}} Y_i, \quad \forall J \in J \tag{1}
\]

\[
Y_i \Rightarrow Y_{tr}, \quad \forall a_t \in \cup_j A_j, a_t \in \bar{A} \tag{2}
\]

\[
Y_i \Rightarrow st(i) \in TW_{CO_i}, \quad \forall a_t \in \cup_j A_j \tag{3}
\]

\[
Y_i \cdot (st(i) + d_i^r + t_{ij}^r - st(j)) \leq 0, \quad \forall r \in R, (i,j), (a_t, a_t) \in E_r \tag{4}
\]

\[
0 \leq L_T(t) \leq Q_T, \quad \forall t \in [0,H], \forall r \in R^2 \tag{5}
\]

\[
X_j, Y_i \in \{0,1\}, CO_i \in \cup_j O_{i,r}, st(i) \in [0,H] \tag{6}
\]

Equation (1) means that key operations supporting one task should whether all appear or none appears in the final plan. Equation (2) implies processes of adding auxiliary operations with the opportunity it uses. The expression also implies instantiations of other operations in support of key operations. For example, if an imaging operation requires a certain satellite attitude, then an attitude maneuvering operations might be instantiated and added. Equation (3) means that if an operation is in the final plan, then its start time must be among the time window in corresponding operations. Equation (4) means that if two operations appear in the final plan successively, then there must be enough transition time left between them. Equation (5) means that at any time throughout the plan horizon, the accumulated amount of occupation, consumption or regeneration of reservoir resource should not exceed its maximal capacity. Equation (6) just states domains of the decision variables.

2.3. Optimization Objective
Even for AEOS, it must be emphasized that not every customer requests with available time windows could be satisfied. There are always contests and rejections. So the optimization objective of our problem is maximizing the weighted sum revenue of completed observation tasks. It can be expressed as follows.

\[
\max_{j \in [0,n]} \sum c_j \cdot X_j
\]

3. The decision framework
The planning and scheduling process goes on mainly in a periodical way with a regular horizon. Logically and practically, each period can be divided primarily into four steps. Step1 will first decompose complex tasks into single strips with certain order or logical relations. Then observing time windows or opportunities and the corresponding satellite attitudes for each single strip will be calculated using orbit dynamics. Also visible time windows between the satellite and all available ground stations will be calculated in a similar way. Step2 will solve the planning and scheduling problem in an integrative way. Some estimation models must be used to predict mission dynamics or satellite status throughout the time horizon, such as attitude maneuvering model for estimating maneuvering time, payload or downlink bit rate model for estimating data storage consumption or duration of data downloading, battery model to estimating energy capacity, etc. Step 3 is a dynamic replanning process. It will check the initial plan with more accurate and recent information, such as cloud and real resource status. Conflicts or changes will be handled to assure a feasible plan ready to be executed. Step4 is the final step which will translate the final plan into command sequences that can be understood by the satellite. Some auxiliary operations are added, such as switching on or off the camera, heating before imaging, and so on.

Taken together, we propose a planning and scheduling framework combining on-board and on-ground decision as Figure 1 shows. The formerly mentioned step 1 and step 2 are processed at the on-ground part where only rough estimation models will be used due to the difficulty of getting accurate
inputs. Step 3 and step 4 are completed at the on-satellite part. At this time, status information of sub-system on board such as camera, battery, attitude controller, temperature controller and downlink antenna are collected in near real-time and distributed through buses on board. If supervisors find that there are big differences between estimated status and real status, or there are resource failures that will cause the initial plan infeasible, re-planning process will be triggered to solve the conflicts. Since on-ground decision used rough estimated models, re-planning may always happen, but will not change the plan much more. The revised plan will then be translated into executable commands and distributed to those sub-systems on board. Such a framework is convenient to take advantage of on-ground capabilities of computer resources and relieve the calculation burden on board.

![Figure 1. The combinatorial decision framework for planning and scheduling.](image1)

![Figure 2. Primary work-flow for the planning and scheduling algorithm.](image2)

4. Planning and scheduling algorithms

Due to the mixed planning and scheduling features and many dynamics needed to be estimated and considered, it is really hard to model the problem as some usual formats such as mixed integer programming and then use sophisticated methods to solve it. To get through these difficulties, we use some rule-based heuristics that are efficient for the problem.

The algorithm start from time point 0 which is the beginning of a planning horizon and go ahead along the timeline while using some rules for decision. All the opportunities for observation and data downloading are ordered according to their earliest start time. So we get a queue of key operations with always the first one as a current operation to be considered. Primary flow of the algorithm can be demonstrated in Figure 2. If current operation is an observing opportunity, the algorithm will look ahead in the queue within a certain length to decide if current operation or an operation be ahead would be accepted. In this process, a planning procedure will begin to add auxiliary operations and also check all resource constraints. The selected operation and its related auxiliary operations will be settled down. If current operation is a data downloading opportunity, looking ahead is not needed.
Certainly, the algorithm needs to maintain information about all the resource status dynamics all along the time.

The initial ground-generated plan is based on many resource models and may not be very precise. So, re-planning on board will use newly collected information about all the resource status and weather conditions to check the initial plan’s feasibility. The detailed process can be described here. When the algorithm finds a conflict at a certain time point, it turns to the key operation that directly causes the conflict and tries to slide the key operation in its long time window. If the obvious conflicts could then be resolved, then other auxiliary operations will be added to get a feasible part-plan. If conflicts still exist, then the operation is just cancelled. The process is repeated until getting a feasible new plan. Such a simple method is fit for the poor computing capability on board.

5. Application

The planning and scheduling process and methods proposed in this paper are designed primarily for a specific AEOS. Since the satellite has some individual features, we don’t have benchmark problems to have some tests and comparisons. Practically, the main purpose of this paper is not to provide sophisticated algorithms, but a framework and processes as a whole. Our results have been applied in some simulation cases to assist design optimization of the satellite. Basic structure of the simulation system is demonstrated in Figure 3.

![Figure 3. Basic structure of the simulation system](image)

We did some experiments based on the simulation systems. The AEOS orbit parameters at time 2016-07-26 00:00:00 is shown in Table 1 and key parameters of the on-board instruments were list in Table 2. We chose five ground stations as the available ones which are all inside China. Scenarios used are randomly generated which cover different numbers of customer requests, and different distribution of ground targets. Targets in instance 2 and 3 are intensively distributed, whereas they are randomly distributed in other instances.

| Table 1. Satellite orbit parameters |
|-------------------------------------|
| Semimajor Axis (m) | Eccentricity e | Inclination (d) | RAAN (d) | Arg. of Periapsis (d) |
|---------------------|----------------|-----------------|-----------|----------------------|
| 7051203             | 0.002344       | 97.308688       | 249.784   | 0                    |

| Table 2. Key Parameters of On-board Instrument |
|-----------------------------------------------|
| Max Roll Angle (degree) | Max Pitch Angle (degree) | Best ground resolution (m) | Battery Capacity (relative value) | SSR Capacity (TB) |
|-------------------------|--------------------------|---------------------------|-------------------------------|------------------|
| 45                      | 45                       | 5                         | 100                           | 1                |

Table 3 gives the key parameters of 6 typical scenarios, in which \( N_R \) denotes the number of customer requests, \( N_S \) denotes the number of strips, \( N_M \) denotes the number of jointing strips requests, \( S_V \) denotes the sum values of all the requests. Here each request has a value between 1 and 10 which was assigned by the ground decision maker according to emergency and importance of the request.
Table 4 shows the ground planning results in 24 hours horizon. Profit denotes the total revenue of all completed requests. Time denotes the duration needed for the algorithm to generate a result in seconds. P denotes the number of completed requests. Actions denote the number of all the key operations and auxiliary operations in the final plan. For computing, we use a computer with a P4 CPU of 3.0 GHz and 2G RAM.

| Table 3. Parameters of instances | Table 4. Planning results of 24 hours |
|----------------------------------|-------------------------------------|
| Instances | $N_R$ | $N_S$ | $N_M$ | $S_V$ | | Instances | Profit | Time(s) | P | Actions |
|-----------|-------|-------|-------|-------|-----------|-------|-------|-----|-------|
| 1         | 47    | 122   | 3     | 254   | 1         | 261   | 24    | 46  | 384   |
| 2         | 46    | 122   | 4     | 295   | 2         | 169   | 19    | 29  | 249   |
| 3         | 42    | 112   | 3     | 278   | 3         | 179   | 19    | 27  | 249   |
| 4         | 46    | 101   | 4     | 297   | 4         | 297   | 18    | 46  | 372   |
| 5         | 200   | 480   | 70    | 1137  | 5         | 1009  | 81    | 180 | 1263  |
| 6         | 300   | 734   | 150   | 1855  | 6         | 1396  | 126   | 239 | 1633  |

It can be seen that there are relatively more non-completed requests in instance 2 and 3, because targets are too intense and the satellite is still not agile enough to maneuver so quickly. For instance 5 and 6, data collected for some targets could not be downloaded within the planning horizon since the satellite had flown out of the range of all the ground stations. It also shows that there are many auxiliary operations added in the plan to complete the requests. That is just the planning feature of our problem that is different from common scheduling problems. We also find that the look-ahead algorithm and revising algorithm are both efficient in computing time. For all these scenarios, computing times used are almost all within tens of seconds for look-ahead algorithm and always several seconds for re-planning algorithm.

Table 5 shows the results of on-board decision, in which AT denotes the average time consumed for re-planning in seconds, NC denotes the times of re-planning triggered in each 24 hours planning horizon, VP denotes the variant in total revenue finally got. The computer used is also with a P4 CPU of 3.0 GHz and 2G RAM. From Table 5, we can also find that there were frequent status deviations that would trigger the re-planning. Since our simple re-planning algorithm mainly aims to solve conflicts but not to insert new targets, the variant in revenue is always negative, meaning that some requests are cancelled. But in fact, without the on-board re-planning, such deviations may cause damage to the satellite health or cause more useless data.

Also the simulation system was used to help designing the satellite by varying some key parameters to see what will happen on the application effects. In fact, it shows that increasing agility will greatly improve the satellite’s ability to satisfy customer requests as 4 times as much. Such experiments and applications also show that the planning and scheduling framework and algorithms provided in this paper are reasonable in a certain sense.

| Table 5. Planning results of 24 hours |
|-------------------------------------|
| Instances | AT(s) | NC | VP |
|-----------|-------|----|----|
| 1         | 5.2   | 15 | -23|
| 2         | 5.1   | 18 | -50|
| 3         | 4.9   | 14 | -37|
| 4         | 5.0   | 16 | -25|
| 5         | 6.5   | 28 | -75|
| 6         | 6.7   | 30 | -82|

6. Conclusion
This paper primarily provides a planning and scheduling framework for a specific AEOS which has some individual features. Mostly ascribe to insufficient communication time windows and limited on-board computing capabilities, decision processes are distributed both on ground and on board. The on-ground part is more complicated and uses a heuristic look-ahead algorithm, whereas the on-board part
is quite simple and uses some rules to make decision. Those ideas and algorithms are realized within a simulation system. Results of some experiments show that the framework and methods are reasonable and feasible.

In fact, since there are more and more powerful ground computing resources that can be used, ground decision part may use more sophisticated algorithms that generate better solutions. We are trying to develop other techniques for this problem, like constraint reasoning and meta-heuristics that are combined together to improve the computing quality and efficiency.

References
[1] W.-C. Lin, et al., "Daily Imaging Scheduling of an Earth Observation Satellite," IEEE Transactions on Systems, Man, and Cybernetics—part A: Systems and Humans, vol. 35, pp. 213-223, 2005.
[2] M. Lemaître, et al., "Selecting and scheduling observations of agile satellites," Aerospace Science and Technology, vol. 6, pp. 367-381, 2002.
[3] P. Grandjean, et al., "What on-board autonomy means for ground operations: An Autonomy Demonstrator conceptual design," space operations symposium, 2004.
[4] M. Vasquez, J. Hao, “A Logic-Constrained Knapsack Formulation and a Tabu Algorithm for the daily photograph scheduling of an earth observation satellite,” Computational Optimization and Applications, vol. 20, pp. 137-157, 2001.
[5] W. Wolfe, S. Sorensen, “Three scheduling algorithms applied to the earth observing systems domain,” Management Science, vol. 46, pp. 148-166, 2000.
[6] B. Sun, L. Mao, W. Wang, et al., “Satellite mission scheduling algorithm based on genetic algorithm,” Kybernetes, vol. 38, pp. 1255-1261, 2010.
[7] W. Martin, “Satellite image collection optimization,” Optical Engineering, vol. 41, pp. 2083-2087, 2002.
[8] E. Bensana, M. Lemaître, and G. Verfaillie, “Earth observation satellite management,” Constraints, vol. 4, pp. 293-299, 1999.
[9] J. Pemberton, F. Galiber, “A constraint-based approach to satellite scheduling”, in DIMACS workshop on constraint programming and large scale discrete optimization, Washington DC, 2001, pp. 101-114.
[10] A. Globus, J. Crawford, “A comparison of techniques for scheduling fleets of earth-observing satellites,” in National Conference on Artificial Intelligence - AAAI , 2008, pp. 836-843.
[11] Li Jufang, Yao Feng, et al., “A decomposition-based algorithm for imaging satellites scheduling problem”, in ICIECS 2009, pp. 1-6
[12] G. Beaumet and G. Verfaillie, "Estimation of the minimal duration of an attitude change for an autonomous agile earth-observing satellite," in Proc. of the 13th international conference on Principles and practice of constraint programming, 2007, pp. 3-17.
[13] G. Beaumet, "Continuous Planning for the Control of an Autonomous Agile Satellite,"2006.