Towards a visual tool for swath acquisition planning in multiple-mission EOSs

Jorge Galán-Vioque
Rafael Vázquez
Emilio Carrizosa
Carlos Vera Flores
University of Sevilla
Camino de los Descubrimientos s/n
41092 Sevilla, Spain

Federico Perea
Universidad Politécnica de Valencia
Camino de Vera s/n
46021 Valencia, Spain

Felipe Martín Crespo
Taitus Software
Via Cardinal Guglielmo Massaia 12A
00044 Frascati, Italy

Abstract

We report on progress made on the development of a real-time planning tool for multiple-mission Earth Observation Satellites (EOSs). The problem under consideration is to decide which acquisitions are needed to fulfill a series of criteria, such as the minimization of the total acquisition cost or the maximization of the area covered. Several models of increasing complexity have been developed; the most basic models are based on standard covering algorithms, which are adequate for the case in which no dynamical constraints, such as duty cycles, are imposed. The underlying Computational Geometry problem has been reduced to the computation of a matrix, which allows one to use standard Integer Programming tools and software. Given the complexity of the problem and the requirements of obtaining solutions in real time, heuristic algorithms, yielding (possibly sub-optimal) solutions to the problem are needed. A greedy and a GRASP algorithm have been implemented. Preliminary computational results are presented, comparing the heuristic algorithms with the exact solution.

Earth observation satellites (EOSs) are a class of geocentric satellites whose task is to collect data of the Earth using advanced sensing technology. Such data are useful in disciplines that study the Earth lands, oceans, and atmosphere and their interaction, such as cartography, meteorology, oceanography, biology, geology, geodesy, or atmospheric science. The field of EOSs is quickly evolving and new applications are emerging: fire detection, crisis management or fishing zone identification. For these reasons, EOSs have become an important resource for global Earth surveillance and research. Nowadays, many countries and companies all over the world are actively developing and deploying EOSs. However, although the fleet of EOSs is growing, their number is not high enough to satisfy the ever-increasing global requirements for remote sensing data. Hence, EOSs resources have to be efficiently managed to obtain the maximum possible benefit.

Currently, most remote sensing activities require manual coordination of satellites and observations by mission planners or shceduling algorithms (Sun, Wang, and Qin 2008; Muraoka et al. 1998; Chien, Smith, and Rabideau 1998). However, with the existing number of EOSs and demands for observation time, it is becoming unfeasible to manually plan coordinated EOSs activities. Instead, observation requests should be processed by automatic planning algorithms which select and schedule a subset of satellites yielding the maximum profit under operational constraints, such as satellite availability, power, thermal, data capacity, clouds, duty cycles of the sensors and the limited time each satellite spends over a target.

This problem has traditionally been studied for the case of one single satellite. However, nowadays Earth Observation operations are increasingly moving towards multi-satellite scheduling, looking for shorter revisiting times and quicker access to space resources, which is crucial for applications like disaster management and systematic large area mapping. Thus, since requests might be satisfied by *several satellites* belonging to the same constellation or even to a different mission, in more than one of their revolutions, the problem is not separable by satellite or orbit. Instead, planning must be performed simultaneously for *all satellites and orbits* considered.

In addition, many missions require rapid decision and management (for instance, humanitarian assistance or damage assessment), or they depend on rapidly changing data, such as clouds. Hence, planning algorithms should ideally find the optimal solution in (almost) *real-time*. However, taking into account the number of operating satellites, the total number of observations they can perform, and the number of existing constraints and options for each observation, the search space for EOSs scheduling problems might become too large. In general, exact algorithms cannot be expected to be feasible for real-time operation and instead *heuristic algorithms* have to be developed and applied. Taking advantage of new mathematical techniques for handling data and parallel computing, these algorithms are able to quickly find an almost-optimal solution.

An explicit example generated with SaVoir (Swath Acquisition Viewer, (Taitus 2011)) is shown in figures 1 and 2. In this example, the region of interest is Italy, and it can be seen how acquisitions from several satellites overlap, providing redundant images. The unoptimized solution in figure 1 is formed with acquisitions generated automatically with a simple ”maximal-coverage” algorithm, with no memory. Each acquisition is planned independently of the others. SaVoir selects the best sensor steering to maximize the coverage. Because there is no memory the output tends to overlap previous data takes leaving many parts uncovered. The results shown in figure 2 are selected with a similar strategy but with memory. The sensor steering will be selected with a ”maximal-coverage” criteria, but incrementally discarding coverage of previous data takes. It produces a solution with less acquisitions but not necessarily the best choice (in time, coverage, cost or a combination of these criteria).

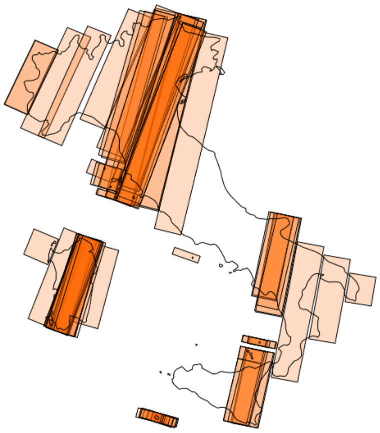

Figure 1: A real-world example produced with SaVoir (Taitus 2011). Swaths from different satellites produce a redundant and overlapping set of acquisitions. Each acquisition is planned independently of the others. SaVoir selects the best sensor steering to maximize the coverage. Because there is no memory the output tends to overlap previous data takes leaving many parts uncovered.

This contribution formulates the swath acquisition problem (SAP) arising in multi-satellite and constellation management as a mathematical programming problem and proposes a heuristic algorithm to provide a feasible solution. Current approaches to solve SAP are mostly based on simple enumeration of possible solutions, which may be too time consuming, mainly when different criteria or satellites priorities are taken into account. Moreover, duty cycle constraints and other similar constraints of dynamic nature (batteries, recorders, downlink capacity) are not frequently incorporated within the optimization process. Hence, the output of existing procedures may be far from optimal. However, if these dynamical constraints are not relevant, an adequate customization of well-known Mathematical Optimization methods can yield several feasible algorithmic strategies to address SAP.

We present our preliminary work, including a mathematical model for describing SAP, the study of several simplified models, a heuristic algorithm to solve the problem, and some preliminary results. Our ultimate goal is to design, develop and implement a real-time tool for EOSs planning, which includes several multiple-mission, multiple-constellation algorithms able to handle realistic operational constraints and its integration into SaVoir (Taitus 2011), a visual, simple-to-use tool for satellite mission planning and management. One of the advantages of using SaVoir is that its visual engine implements computational geometry algorithms that can compute (as in Figure 1) the subregions in which the region of interest is divided by the different available swaths, and what swaths cover each subregion. These computations are performed with embedded GIS capabilities and 3-D geometry. The geometrical algorithms include provisions to cope with Earth ellipsoid singularities over the poles and line of date (180 degree), such that the areas of interest may be positioned anywhere on the Earth without calculation or visualization restrictions.

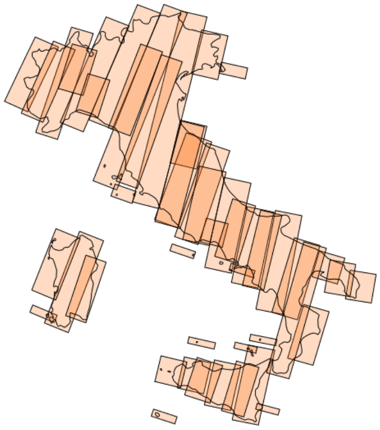

Figure 2: Optimized solution with an heuristic two steps approach. The acquisitions are selected with a similar strategy as in figure 1 but with memory. The sensor steering will be selected with a ”maximal-coverage” criteria, but incrementally discarding coverage of previous data takes. It produces a solution with less acquisitions but not necessarily the best choice (in time, coverage, cost or a combination of these criteria).

Methodology

Our approach towards an efficient, visual swath acquisition planning algorithm can be summarized in four steps:

1. Formulating the EOSs planning problem in an adequate mathematical setting, which allows us to use standard optimization algorithms.

   Several models of increasing complexity have been developed. First, a simplified model is constructed for simple regions, simplified satellite ground tracks, and fixed satellite sensors. Later, real-world regions and ground tracks, and steerable satellite sensors are included. Finally, complex constraints such as maximum sun angle or duty cycle limitations are included in the model.

2. Solving the involved computational geometry problem, which implies calculating intersections between Earth regions and different satellite swaths.

These problems are not difficult for the initial simplified models, but their complexity increases with the model. The algorithms should be fast and efficient, and they should be able to handle complex regions, possibly nonconvex and with holes. The algorithms should be free of projection distortion, and they should successfully work with regions located anywhere in the world.

The computational geometry problem can be summarized as the computation of all the subregions generated by the intersections of the satellite swaths within the region of interest. Based on such intersections, one can compute a matrix Q, whose entry (i,j) takes the value 1 if subregion i is covered by swath j, and takes the value 0 otherwise.

3. Implementation of exact and heuristic algorithms to solve the problem.

   Given the matrix Q from the previous step, the SAP can be seen as a so-called set covering problem, (Schilling, Jayaraman, and Barkhi 1993), solvable by means of standard Integer Programming software. However, it has been shown that the swath segment selection problem is an **NP**-hard combinatorial optimization problem (Cordone, Gandellini, and Righini 2008), thus only small-size instances are expected to be solved exactly in short time.

   Hence, given the complexity of the problem and the need of obtaining solutions in real time, the use of exact algorithms is, in general, unfeasible. Therefore, heuristic algorithms have to be developed and implemented to rapidly find a (possibly sub-optimal) solution to the problem. To measure the quality of the solutions provided by those heuristic algorithms, it would be important to implement methods that produce the exact solution, though with a much higher computational cost. Then, comparisons between the solutions provided by exact and heuristic algorithms can be carried out to give an idea of what is lost in terms of costs when heuristics are used.

4. Integration of the planning algorithms in the SaVoir visual satellite simulation environment.

   Once the problem has been modeled and solved, the developed algorithms should be integrated in a visual satellite simulation tool, friendly for users (satellite planners). The tool should allow one to to select a set of real satellites, a region to be observed, a cost index and a set of constraints, and it should give in real time the optimal (or almost-optimal) subset of time-observation frames for the satellites. To do this, the tool will use realistic propagators to compute future satellite orbits based on existing orbital elements, solve the involved computational geometry problems, and apply the previously developed heuristic algorithms to find a good solution.

   In our opinion SaVoir gives a perfect match to the requirements above, since it is easy to use, inexpensive compared with other solutions in the market, and it already implements a computational geometry engine which can be used to compute the matrix Q.

Formulation of the EOSs planning problem

In this section we introduce several mathematical models that aim at describing the problem introduced before. We begin with several definitions of the key concepts that play a role in our model. Then, we show how to express different versions of the EOSs planning problem as optimization problems. We start with a simplified model, in which the involved satellites have fixed sensors; this model clarifies the integer programming formulation of the problem. The model is illustrated with several simple examples. Next, a straightforward extension in which satellites have steerable sensors follows. The last model describes a more realistic situation in which sensors can be used for a limited period of time only. Finally we comment on an heuristic approach to the solve the problem.

Statement of the problem and notation

To formulate the EOSs planning problem, the following concepts are defined:

- $\mathcal{R}$ is the region of interest, i.e., the region of the Earth that needs to be covered. No assumptions are imposed on the shape of $\mathcal{R}$.
- T is the time-frame for the planning problem. T is assumed to be an interval $[T_0, T_f]$ given by initial the and final times T_0 and T_f.
- S is the set of satellites considered in the planning problem. To avoid dealing with orbit propagators, it is assumed that the position of each satellite in S is known and it can be computed with enough precision for each time instant in T.
- For each satellite $s \in S$, P_s is the set of possible sensor angle positions for s.
- Given a satellite $s \in S$, a sensor position $p \in P_s$ and a time interval $[t_0, t_1]$, an acquisition $a(s, p, t_0, t_1)$ is defined as the surface of the Earth covered by the swath of satellite s during $[t_0, t_1]$ in its position p. Define also the cost of the acquisition a as $c_a > 0$.
- $\mathcal{A}$ is the set of all possible acquisitions given the set of satellites, their possible sensor positions and the time frame T,

$$\mathcal{A} = \{a(s, p, t_0, t_1) : s \in S, p \in P_s, [t_0, t_1] \subset T\}.$$

- Subregions SR: The intersection of the set $\mathcal{A}$ with the region $\mathcal{R}$ defines a set of subregions whose union is equal to the region of interest.

Several concepts from these definitions are illustrated in Figure 4.

Based on these concepts, we now define *admissibility* of acquisitions for the EOSs planning problem. We say a selection of n acquisitions $\{a_i(s_i, p_i, t_0^i, t_1^i), i = 1, \ldots, n\} \subset \mathcal{A}$ is admissible if:

- Each individual satellite s in the selection is not used more than once for any given time instant, i.e., $\forall\ i, j = 1, \ldots, n,\ s_i = s_j \Rightarrow [t_0^i, t_1^i] \cap [t_0^j, t_1^j] = \emptyset$.

- If a satellite s in the selection is used more than once with different sensor positions, a time Δtp_s is needed to change its sensor position, i.e.,

$$\forall i, j = 1, \ldots, n,\ s_i = s_j, p_i \neq p_j \Rightarrow$$

$$[t_0^i, t_1^i + \Delta tp_i] \cap [t_0^j, t_1^j + \Delta tp_j] = \emptyset.$$

- Depth of coverage d_j: if a subregion of $\mathcal{R}$, $\mathcal{SR}_j$, is specially relevant, it is advisable to have it recorded more that once. Parameter d_j is a non-negative integer which allows these "specially interesting" regions to be acquired more than once. It also allows to include subregions with $d_j = 0$, which means that they are "not so interesting", and therefore not required to be acquired at all.
- Duty cycle constraints: For each satellite s there is a set of n_s subintervals $T_1^s, T_2^s, \ldots, T_{n_s}^s$ such that $\cup_{j=1}^{n_s} T_j^s = T$, and a number $d_s \in [0, 100]$ such that s can be used in each T_i^s only a percentage d of the time, i.e.,

$$\forall\, s\, \forall\, j = 1, \ldots, n_s,$$

$$\sum_{s_i = s} length([t_0^i, t_1^i] \cap T_j^s) \leq d_s length(T_j^s).$$

With this notation we are in position to formulate the EOSs planning problem as follows.

EOSs planning problem: Find an optimal selection of admissible acquisitions $\{a_i(s_i, p_i, [t_0^i, t_1^i]) \in \mathcal{A},\ i = 1, \ldots, n\}$ such that $\mathcal{R} \subset \cup_{i=1}^n a_i$. If some of the subregions SR_j are marked as *not so interesting* through the depth of coverage parameter $d_j = 0$, the last condition should be changed to $\cup_{j:d_j \neq 0} SR_j \subset \cup_{i=1}^n a_i$.

The selection is optimal in the sense that a certain function $F(\{a_1, \ldots, a_n\})$ is minimized; F can have different definitions according to the objective, for instance:

- $F = n$ (minimal number of acquisitions).
- $F = \sum_{i=1}^n c_{a_i}$ (minimum cost of acquisitions).
- $F = \max_{i=1,\ldots,n} t_1^i$ (minimal final time).
- A combination of any of the above.

Alternatively, the objective could be to maximize the surface covered having a threshold value for the final time and/or maximum budget, etc.

Simplified model I: q-satellites and fixed sensor

The first model to be studied involves q-satellites with a fixed sensor, without duty cycle constraints; then the complete swath of the satellite can be used, and the set of useful acquisitions is a finite set consisting of the intersections of the complete swath with $\mathcal{R}$. In this case, given a maximum operation time $T_{\max}$, we consider a time frame $T = [0, T_{\max}]$. The goal is to select acquisitions during T so that the region of interest $\mathcal{R}$ is covered at a minimum cost. Let $a_1, ..., a_n$ be the set of possible acquisitions, increasingly sorted in time. Then, a_n is the last acquisition that can be used, that is, a_{n+1} would occur after T_{max}. As input data, the time in which acquisition a_i starts scanning $\mathcal{R}$ and the time in which it gets out of $\mathcal{R}$, respectively t_i^0 and t_i^1, are known. Note that, for the model to be meaningful, we need to assume that $t_i^0 \leq t_i^1$, $t_i^0 \leq t_{i+1}^0\ \forall\ i$, $t_1^0 \geq 0$, and $t_n^1 \leq T_{max}$.

As a simplified example of this situation consider the case depicted in Figure 3. The region of interest is the solid black rectangle $\mathcal{R}$, the four possible acquisitions are $\{a_1, a_2, a_3, a_4\}$ (thin rectangles) sorted in arriving time.

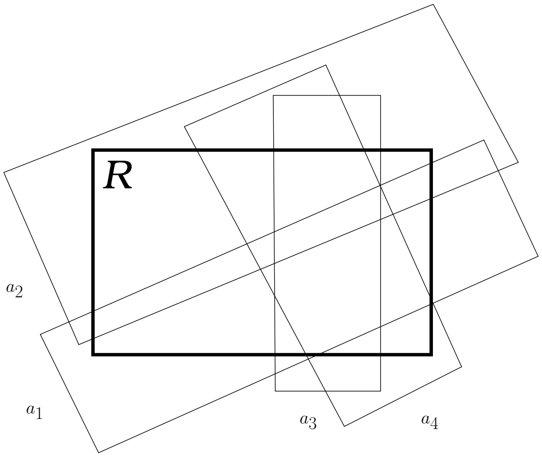

Figure 3: Simplified example of swath acquisition problem. The region $\mathcal{R}$ of interest (solid rectangle) is being covered by 4 acquisitions $\{a_1, a_2, a_3, a_4\}$ sorted in arriving time. Acquisitions a_3 is redundant and should not appear in the optimal solution.

We could think of a very simple algorithm that would first choose a_1, since it covers a certain area of the region of interest. Then, it would iteratively pick acquisitions a_2, a_3 and a_4, since all of them cover a certain new area of the rectangle. This algorithm would stop as soon as the whole region of interest $\mathcal{R}$ is covered.

With this process, the four acquisitions are necessary to cover the whole region of interest. Note that this example has been introduced just to show that any *ad hoc* non-optimal approach will certainly provide feasible solutions (if enough acquisitions are available), but the best one may not be obtained.

In order to solve our problem optimally, we formulate it as a mathematical programming (MP for short) problem. For a complete introduction on MP see (Bazaraa, Jarvis, and Sherali 1990; Wolsey 1998).

Let x_i be a binary variable (i.e., it can only take the values 0 and 1) saying if acquisition a_i is to be used ($x_i = 1$) or not ($x_i = 0$), and let c_i be the cost of using acquisition a_i ($c_i > 0$). Then, the EOSs planning problem can be posed as

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{s.t.} \quad & \bigcup_{i:x_i=1} a_i \supset \mathcal{R} \qquad (1) \\ & x_i \in \{0,1\}, \quad \forall\, i = 1, 2, \ldots, n. \end{aligned}$$

Note that this problem can be infeasible. For instance, if there is a subregion of $\mathcal{R}$ not covered by any of the acquisitions available, then the constraint of (1) can never be satisfied.

To be able to compute the solution of (1), we model this problem as an Integer Linear Programming (ILP) problem (Bazaraa, Jarvis, and Sherali 1990; Wolsey 1998). To that end, the constraints in (1) must be expressed as linear constraints. We show now how to do it. Let $\{SR_i, ..., SR_m\}$ be the subregions in which $\mathcal{R}$ is divided considering all intersection of the acquisitions a_i with $\mathcal{R}$ and among themselves. From this set of subregions, we obtain a matrix Q, whose entry q_{ij} takes the value 1 if subregion SR_j is covered by acquisition a_i, and 0 otherwise. With this new matrix, Problem (1) can be formulated as

$$\begin{array}{ll} \min & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & \sum_{i=1}^{n} x_i q_{ij} \geq d_j, \quad \forall\, j = 1, ..., m \\ & x_i \in \{0, 1\}, \quad \forall\, i = 1, 2, \ldots, n. \end{array} \tag{2}$$

Taking advantage of this formulation, in the constraints of (2) we have included depth of coverage constraints, forcing each subregion SR_j to be covered by at least d_j different acquisitions.

We will now apply our ILP model (2) to the example of Figure 3, showing that some of the acquisitions might not be needed. We consider that all costs are equal and therefore, by the linearity of the problem, we can set $c_i = 1$ for all $i = 1, ..., 4$. In this example, the region of interest $\mathcal{R}$ is subdivided in 18 subregions as shown in Figure 4.

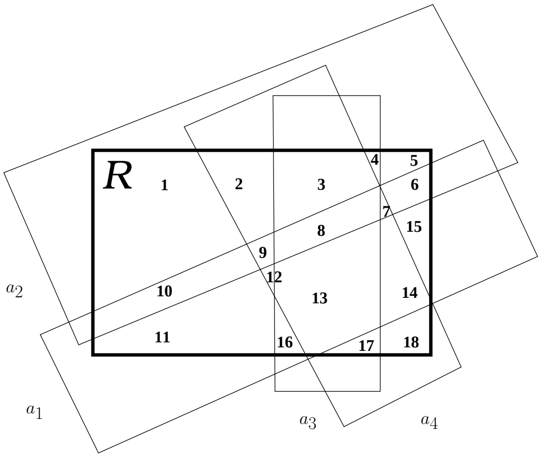

Figure 4: Subregions generated in $\mathcal{R}$ by the intersection of the acquisitions with each other and with the region of interest. In this simple example 18 subregions are generated.

Matrix Q results:

$$Q = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Let us explain the meaning of rows and columns. The first row states that subregion SR_1 is only covered by acquisition a_2. In contrast, the third column says that acquisition a_1 covers subregions $SR_3, SR_4, SR_8, SR_{13}, SR_{16}$ and SR_{17}.

Note that matrix Q can be simplified by merging all identical rows into one row only (which implies joining the corresponding subregions). The same can be done with columns. For instance, rows 12 and 14 are identical, meaning that subregions SR_{12} and SR_{14} are covered by the same satellites, and they can therefore be treated as one single subregion. This way the size of the ILP problem is reduced. Whether or not this reduction is worthy is still an open question.

The computational complexity of matrix Q could become a bottleneck of the algorithm, even bigger than the ILP, and an efficient implementation of computational geometry operations (union, intersection, subtraction) is needed. Besides, a well structured *area accounting system* should be introduced to deal with frequent area related constraints in the applications (acquisitions with negligible covering should not be included, for instance).

Setting $d_j = 1 \;\forall\; j$, the solution to Problem (2) with the data of this example is $x_3 = 0$, $x_1 = x_2 = x_4 = 1$, which means that, as we had anticipated, acquisition a_3 is not needed to cover the area of interest.

Simplified model II: q-satellites and steerable sensor

Assume now that the satellites have a steerable sensor, that is, the sensor angle with respect to the nadir can be changed in a certain range in order to better acquire the region of interest. In the simplified model, we assume that the mode (angle) of the satellite's sensor can be changed for each revolution, but it has to be maintained during all the period that the satellite is scanning the region of interest. We also assume that the sensor has K possible discrete modes (typically $K \leq 256$). The previous simplified model (2) can be easily adapted to this case just by adding a superscript k that specifies the mode and a set of constraints that avoid the change of mode along the same acquisition. Therefore, let x_i^k be a binary variable, which takes the value 1 if acquisition i is chosen with the sensor in position k, and 0 otherwise. We now have to build the subregions of $\mathcal{R}$, taking into account that every satellite revolution generates up to K different acquisitions (at most one of which is to be chosen). Matrix Q now becomes a 3-dimensional matrix. Its entry q_{ij}^k is 1 if subregion SR_j can be photographed by acquisition a_i using its k^{th} position and it takes the value 0 otherwise. Therefore, Problem (2) becomes

$$\begin{aligned}
\min \quad & \sum_{k=1}^{K}\sum_{i=1}^{n} c_i x_i^k \\
\text{s.t.} \quad & \sum_{k=1}^{K}\sum_{i=1}^{n} x_i^k q_{ij}^k \geq d_j, \quad \forall\, j = 1, \dots, m \\
& \sum_{k=1}^{K} x_i^k \leq 1, \quad \forall\, i = 1, \dots, n \\
& x_i^k \in \{0,1\}, \quad \forall\, i = 1, \dots, n, \forall\, k = 1, 2, \dots, K.
\end{aligned} \tag{3}$$

The first set of constraints states that all subregions must be covered by at least one acquisition at one of its possible modes. The second set of constraints forces that the same acquisition cannot be used in more than one position.

Constrained model I: q-satellites, fixed sensor and duty cycle constraints

Assume further that each acquisition a_i can only be used for a limited period of time $SLOT_i$. Variables are now *functions* of time and constraints become integrals and sums. This way, x_i^t is 1 if acquisition a_i is used at instant t and 0 otherwise, and q_{iy}^t is 1 if acquisition a_i can photograph point $y \in \mathcal{R}$ at time t. The problem is much more difficult because we have to find *functions* instead of just the values of *variables*. Now the model becomes:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i \int_{t_i^0}^{t_i^1} x_i^t dt \\
\text{s.t.} \quad & \int_{t_i^0}^{t_i^1} x_i^t dt \leq SLOT_i, \quad \forall\, i = 1, \dots, n \\
& \sum_{i=1}^{n} \int_{t_i^0}^{t_i^1} q_{iy}^t dt > 0, \quad \forall\, y \in \mathcal{R} \\
& \frac{1}{A(\mathcal{R})} \int_{\mathcal{R}} q_{iy}^t dy \leq x_i^t, \quad \forall\, t \in [0, T_{max}] \\
& x_i^t \in \{0,1\}, \forall\, i = 1, \dots, n, \forall\, t \in [0, T_{max}].
\end{aligned} \tag{4}$$

The first set of constraints ensure that acquisitions are not used more than what is needed. The second set of constraints forces that each point of the region of interest is acquired by at least one acquisition for at least one nondegenerate interval. Note that this way we avoid the paradox of acquiring only a finite set od instants, having the whole area of interest scanned at zero cost. The third set of constraints says that if point $y \in \mathcal{R}$ is to be covered by acquisition a_i at time t, then x_i^t must be 1.

Heuristics

Due to the complexity of the problems to be solved, and the need to obtain a "good" solution in a relatively short time, heuristic and/or metaheuristic algorithms apply. A heuristic algorithm is a method used to rapidly obtain a solution that is hoped to be close to an optimal solution. See (Michalewicz and Fogel 2000) or (Vazirani 2001) for an introduction to heuristics.

In this work we will apply to the EOSs planning problem a GRASP algorithm. GRASP algorithms (Greedy Randomized Adaptive Search Procedure) have been widely used for solving large-scale optimization problems since the pioneering work by Feo and Resende (Feo and Resende 1989).

A GRASP procedure consists of randomly adding elements to the problem's solution set out of the set of $k \in \mathbb{N}$ elements that individually yield the largest improvement in the objective function when added to the previous solution. This procedure is repeated, and each of the (possibly) different obtained solutions form a set of feasible solutions. The final solution chosen by GRASP is the best out of the feasible solution set previously obtained. When $k = 1$, that is, when we choose at each iteration the element that individually yields the largest improvement in the objective function, the procedure obtained is a greedy algorithm.

We have used the algorithm for problem (2), although the same philosophy could be used to handle problems (3) or (4) after small modifications.

EOSs planning GRASP algorithm:
Input data: Q, k, $\mathcal{R}$, $\{a_1, ..., a_n\}$.
Set $Reg = \mathcal{R}$, $Aq = \{a_1, ..., a_n\}$, $Sol = \{\}$

1. If $|Aq| \leq k$, set $Fq = Aq$. Else, Fq is the set constituted by the k acquisitions in Aq whose strips individually cover the maximum area of Reg.
2. Randomly pick one acquisition a_i in Aq. $Sol = Sol \cup \{a_i\}$. Set $Aq = Aq - \{a_i\}$ and $Reg = Reg \setminus S_i$ (S_i is the region covered by acquisition a_i).
3. If $Reg = \emptyset$, STOP. Sol is a feasible solution. Else, go to 1.

This procedure gives a feasible solution to our problem, Sol_1, that is, a set of satellites whose strips cover the whole region of interest $\mathcal{R}$. In order to explore the feasible solution set, we repeat this problem until we run out of computational time, or we have calculated a fixed maximum number of solutions. Let $\{Sol_1, ..., Sol_m\}$ be the set of feasible solutions calculated. If $Cost_j$ denotes the cost of solution Sol_j, that is, $Cost_j = \sum_{i \in Sol_j} c_i$, our algorithm finishes by choosing the best feasible solution among all that have been calculated. That is, the final solution is Sol_{j^*}, where j^* is such that $\min_{j=1,...,n} Cost_j = Cost_{j^*}$.

In case the acquisitions have different costs, in step 1 we could choose the acquisitions that maximize the ratio (area covered/cost of acquisition).

Inst.	n	K	m	OPT	GR(1)	GR(2)	GR(3)	TQ
1	28	20	5966	5	8	6	6	784
2	22	6	2503	9	9	9	9	51
3	22	7	2900	8	9	8	8	70
4	23	10	4020	7	11	7	7	161

Table 1: Preliminary results.

Preliminary Results

We have implemented and solved the second model (a number of satellites and steerable sensor) for four relatively large random instances: the number of satellites (n) ranging from 22 to 28, the number of possible modes that each satellite could work (K) ranging from 6 to 20, and the number of subregion in which the region of interest $\mathcal{R}$ was divided, denoted by m, ranging from 2503 to 5966. The computational effort goes more to calculating matrix Q rather than to solving the ILP, which was done using CPLEX 11 and the modeling system GAMS 23. The GRASP procedure we have designed was implemented in MATLAB. The results obtained in this preliminary experience are shown in Table 1. The first four columns denote the instance label, its number of available satellites, number of available modes for each satellite and number of subregions generated in $\mathcal{R}$, respectively. Column OPT is the minimum number of satellites needed to cover the whole region (optimal, calculated with CPLEX). Columns GR(1), GR(2) and GR(3) are the number of satellites of the best solution found by by our GRASP algorithms for $k = 1, 2, 3$. Note that GRASP(1) is the classical greedy algorithm. The last column, TQ, is the time needed to compute the Q matrix, the bottleneck of the process.

In all instances but the first one, the GRASP algorithm found the optimal solution for $k = 2, 3$, whereas the greedy algorithm only found the optimal solution in one instance. This reinforces the idea that, greedy algorithms are not an accurate option and therefore other more elaborated heuristics (such as the GRASP algorithms we present here) are needed in order to obtain good solutions.

Concluding Remarks

In this work, we have modeled an EOSs planning problem in a mathematical setting which allows us to use well-known optimization algorithms. Several models of increasing complexity have been developed. The underlying computational geometry problem has been reduced to the computation of a matrix Q, and we have shown how to formulate the problem as a standard Integer Programming problem. However, given the complexity of the problem and the need of obtaining solutions in real time, heuristic algorithms have to be designed and implemented. We have developed a greedy and a GRASP algorithm to solve the problem.

Preliminary tests with greedy and GRASP algorithms have yielded encouraging results. We plan to test other heuristic methods, such as genetic algorithms, tabu search or variable neighborhood search. Exhaustive computational experiments on real data sets are planned to learn which procedure is the most suitable for different scenarios (possibly depending on features such as region size, or geometrical properties of the region). We also plan to include duty cycle constraints in the optimization algorithms. Since it does not seem easy to reduce the resulting optimization problem to well-studied models such as the set covering problem, we will adapt the heuristic algorithms developed for the problem without duty cycle constraints. The insight obtained will guide us to decide which heuristic(s) will be chosen and how constraints should be modeled.

The heuristics yielding the best performance in terms of quality of the solutions and speed will be integrated in SaVoir. SaVoir currently provides an engine capable of performing the necessary computational geometry calculations and an approximated solution based on a simple sequential search of a valid solution that avoids repeated area acquisitions as much as possible. This algorithm is reliable and relatively fast, but lacks any capabilities of tuning and configuration to optimize given criteria. We expect that integrating the developed heuristic algorithms in SaVoir will result in a visual tool to solve the EOSs planning problem (including realistic constraints) that can be extremely useful to mission planners worldwide, for a large number of scenarios.

Acknowledgements

The authors wish to thank the Institute of Mathematics of the University of Sevilla for organizing the *IMUS modelling week*, where this collaboration was started. JGV acknowledges finacial support from MEC through grant MTM2009-07849 and Junta de Andalucía grant P08-FQM-03770 and TIC-130. EC is supported by grants MTM2009-14039, Spain, and FQM-329, Junta de Andalucía.

References

Bazaraa, M.; Jarvis, J.; and Sherali, H. 1990. *Linear programming and network flows*. Wiley.

Chien, S.; Smith, B.; and Rabideau, G. 1998. Automated planning and scheduling for goal-based autonomous spacecraft. *IEEE Intelligent Systems and Their Applications, Vol. 13,* 13:50–55.

Cordone, R.; Gandellini, F.; and Righini, G. 2008. Solving the swath segment selection problem through lagrangian relaxation. *Computers & Operations Research* 35:854–862.

Feo, T., and Resende, M. 1989. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8:67–71.

Michalewicz, Z., and Fogel, D. B. 2000. *How to Solve It: Modern Heuristics*. Springer Verlag.

Muraoka, H.; Cohen, R.; Ohno, T.; and Doi, N. 1998. Aster observation scheduling algortihm. *Proceedings of SpaceOps 1998, Tokyo, Japan.*

Schilling, D. A.; Jayaraman, V.; and Barkhi, R. 1993. A review of covering problems in facility location. *Location Science* 1(1):25–55.

Sun, B.; Wang, W.; and Qin, Q. 2008. Satellites scheduling algorithm based on dynamic constraint satisfaction problem. *2008 International Conference on Computer Science and Software Engineering* 167–170.

Taitus. 2011. Savoir: swath acquisition viewer. *www.taitussoftware.com*.

Vazirani, V. 2001. *Approximation Algorithms*. Springer.

Wolsey, L. A. 1998. *Integer Programming*. Wiley.