Swath-Acquisition Planning in Multiple-Satellite Missions: An Exact and Heuristic Approach

FEDERICO PEREA
Universitat Politècnica de València
Valencia, Spain

RAFAEL VAZQUEZ
JORGE GALÁN-VIOQUE
University of Sevilla
Sevilla, Spain

This paper deals with the swath acquisition planning problem for multisatellite Earth observation missions. Given a set of satellites and a mission time frame, the problem we solve consists of selecting a set of acquisitions from the satellites in order to cover a given region of interest during the requested time frame, optimizing a certain objective function. We show that the planning problem can be modeled as a set covering problem, using basic tools of mathematical programming. The formulation of the model requires the solution of a complex computational geometry problem, and therefore the use of heuristics and metaheuristics applies. In this paper, we discuss the efficiency of the constructive phase of a greedy randomized adaptive search procedure algorithm. Computational results comparing the heuristic algorithms with the exact approach are presented.

Manuscript received November 21, 2013; revised October 17, 2014; released for publication January 12, 2015.

DOI. No. 10.1109/TAES.2015.130751.

Refereeing of this contribution was handled by T. Vladimirova.

JGV acknowledges financial support from grant MTM2012-31821 and P12-FQM-1658, and FP from grants FQM-5849, P09-TEP-5022, and MTM2010-19576-C02-01.

Authors' addresses: F. Perea, Instituto Tecnológico de Informática, Universitat Politècnica de València, Spain, E-mail: (perea@eio.upv.es and fedperea@gmail.com); R. Vazquez, Departamento de Ingeniería Aeroespacial y Mecánica de Fluidos, Universidad de Sevilla, Spain; J. Galán-Vioque, Applied Mathematics II, University of Sevilla, ETSI, Camino de los descubrimientos s/n, Sevilla 41092, Spain, E-mail: (jgv@us.es).

I. INTRODUCTION

Among the class of geocentric satellites, Earth observation satellites (EOSs) are used to collect data of the Earth in order to study lands, oceans, atmosphere, etc. Such data might become useful for fire detection, earthquake and tsunami alerts, identifying fishing zones, etc. Because of their limited number, and their associated costs, efficiently managing their operations is a must.

Despite advances in technology, satellites are frequently manually managed by mission planners (see, for instance, [1–3]). The increasing number of EOSs, as well as the need of optimally coordinating several of them for a given set of observations, makes manual planning infeasible. Therefore, automatic mission coordination tools are necessary (see [4] for a review on the progress that has been made in the use of automated techniques for scheduling space mission operations).

In addition, decisions must be made rapidly or even in real time. Consider for instance humanitarian assistance or damage assessment [5]. Even when the response need not be that urgent, possible constant changes in input data might require the reevaluation of the situation more frequently (for instance, clouds might force a different assignment of EOS's with respect to the original plan). In fact, as satellites are not permanently visible for ground control stations, it would be advisable to design tools able to make autonomous decisions on board, especially in the case of agile satellites (see, for instance. [6, 7]).

In this paper, we start from a set of available EOSs, a region of interest, and a mission time frame, and then consider the problem of selecting a subset of EOSs in order to cover the region of interest at minimum cost during the requested time frame. We denote this problem as the swath acquisition problem (SAP), which we model as a set covering problem [8]. A preliminary version of our algorithms to solve the SAP was presented in [9].

Our approach to solve the SAP is as follows. First, we abstract the SAP as a mathematical programming problem with linear constraints and objective function and with binary variables. This constitutes an integer linear programming (ILP) problem. ILP is one of the most frequently used tools of operations research. It models problems in which a linear function must be optimized while some linear constraints are satisfied. The reader is referred to [10, 11] for a complete introduction on linear programming.

Then, in order to build the ILP model for our SAP, the region of interest is divided into the subregions generated by the available acquisitions. For each such subregion, we check which of the available EOSs can cover it. This problem might be intractable for realistic instances due to the NP-completeness of the set covering problem, see [12]. To be able to obtain a solution in a reasonable time, we propose a heuristic algorithm. A heuristic algorithm is a procedure that rapidly obtains a feasible solution to a large optimization problem, which is hoped to be close to an optimal solution; see [13, 14] for an introduction to heuristics. In this paper, we apply the constructive phase of a GRASP[1] algorithm to the SAP; the algorithm is tested in numerical experiments and shown to provide a feasible solution close to the exact solution of the ILP problem in a fraction of the time taken by an exact solver. Both the exact and heuristic algorithms could be easily integrated into existing multisatellite swath planners [16].

Other SAPs have been formulated and solved in the literature. For instance, [2] defined the swath segment selection problem, which consists of selecting the areas to be observed and the acquisitions needed to observe each of them. This problem was solved via Lagrangian relaxation by [17]. In [18], the planning of two satellites with several modes to detect overflow regions in real time was undertaken with an optimization strategy of the swath covering, analogous to the one introduced in this paper. A multiple criteria approach, for a supporting planner in the selection of a satisfying feasible shot sequence, is addressed in [19] with a two-stage procedure. The authors of [20] address the scheduling problem of a fleet of EOSs, by means of an heuristic stochastic greedy algorithm. In [21], the planning and scheduling of the Cosmo-SkyMed constellation, including both the image up and down load, was analyzed. In [22], large satellite scheduling problems with complex constraints (duty cycle, ground station availability, or even coordination of multiple satellites) are studied. A genetic algorithm, simulated annealing, and stochastic hill climbing are compared, together with random and squeaky transmission. Simulated annealing is proved to perform best over a certain set of problems. A heuristic approach was introduced in [5] to solve the scheduling problem of an Earth-observing natural disaster-monitoring constellation in which several satellites compete to take images under complex operational constraints in a short time. The case of multisatellite, multiorbit, and multiuser management of Earth-observing satellites over a planning horizon is analyzed in [23], with a heuristic tabu search algorithm.

It must be noted that the SAP formulated in this paper is not exactly the same as in the above-referenced works, which makes it difficult to compare our results with those previously obtained by other authors. Our contribution mostly lies in formulating a simple model (which nevertheless captures the essentials of the SAP) and solving it with basic tools from computational geometry and operations research. Having a simple model allows us to solve the problem exactly (thus finding the best solution, albeit at the price of computational speed) and heuristically (much faster, but finding a feasible, but not necessarily optimal, solution), and compare both solutions. The ILP model and the constructive GRASP algorithm developed here can serve as a starting point for many other (similar) EOSs scheduling problems and thus constitute a useful addition to the literature on the SAP.

The rest of the paper is organized as follows. In Section II, we formally define our SAP. Section III introduces the ILP model we have developed for this problem. The heuristic designed for more rapidly finding a feasible solution to the SAP is explained in Section IV. Both the ILP model and the heuristic are computationally evaluated and compared in Section V.

II. THE SWATH ACQUISITION PROBLEM

A. Problem Inputs

To formulate the SAP, the following inputs are needed:

1) $\mathcal{R}$ is the region of interest, i.e., the region of the Earth that needs to be covered. If there are several regions of interest, then $\mathcal{R}$ is the union of all of them. However, note that different regions with different acquisition requirements are not included in the model.

2) The time frame for the planning problem is an interval $T = [T_0, T_f]$, given by the initial and final times T_0 and T_f.

3) S is the set of available EOSs.

4) For each satellite $s \in S$, the relevant sensor field of view α_s is also necessary.

5) P_s is the set of possible sensor angle positions for satellite $s \in S$ (for satellites with off-nadir viewing capabilities).

6) Using these concepts, an acquisition $a(s, k, t_0, t_1)$ is defined as the surface of the Earth covered by the swath of satellite $s \in S$, during $[t_0, t_1] \subset T$, when its sensor is in position $k \in P_s$. The set $\mathcal{A}$ of all possible acquisitions is another required input for the SAP, which however should be computed from the satellite orbits, as explained in the next section.

For simplification purposes, we assume that instruments rotate in-between acquisitions and not during a given acquisition. It is also assumed that satellite storage capabilities and downlink opportunities are sufficient to handle all acquisition data and do not need to appear explicitly in the constraints of the problem.

B. Computation of Acquisitions

For completeness' purposes, we next explain how to calculate the set of possible acquisitions for all satellites in S, when their orbits are given, for instance, as two-line elements for a certain epoch (which should be close to the time frame). However it must be emphasized that this is not part of the SAP algorithm itself, but rather a prerequisite computation.

To compute the acquisitions, the first step is to propagate the orbital elements of the satellites during the time interval of the mission. This can be done using any of the many possible methods available in the literature, which incorporate more or less accurate models of orbit perturbations (see, for instance, [24] and the references therein). Once the elements are known at all times $t \in T$,

[1] Greedy randomized adaptive search procedure (GRASP) algorithms were introduced by Feo and Resende [15] and have since been widely used for solving large-scale optimization problems.

the vector position $\vec{r}_s(t)$ can be computed [25]. Then, the subsatellite point (ground track) is calculated according to the Earth model that is used (typically ellipsoidal or spherical). From the ground track, and taking into account the sensor field of view α_s and the sensor angle position $k \in P_s$, the footprint of the swath is then obtained. As the satellite moves along its orbit, the swath describes a strip-like figure along the surface of the Earth (following the ground track). The intersection of this surface with the region of interest $\mathcal{R}$ is what we call an acquisition.

Because we are considering intersections with the region $\mathcal{R}$, satellites will generate an acquisition only each time the ground track passes close to $\mathcal{R}$ (at most once or rarely twice each satellite revolution). However, if $\mathcal{R}$ is nonconvex, the ground track might consecutively enter and exit the region more than once, generating a fragmented acquisition (consisting on two or more nonconnected parts with gaps in between); for all purposes of our model, each of these nonconnected acquisition fragments is considered a different acquisition by itself.

Notice that the considered satellites will frequently have sun-synchronous orbits [26] because this is the most commonly used orbit for EOSs due to constant lighting properties. Given that these satellites have almost-polar orbits, their ground tracks are approximately straight diagonal lines for latitudes far away from the poles. Thus, for nonpolar regions of interest, their acquisitions will approximate slanted rectangles, and if the region is small, they will generate at most one or two acquisitions per day; this very much simplifies the computational geometry subproblem.

C. Formulation of the Problem

To formulate the SAP, the following concepts are defined:

1) Each acquisition is denoted as $a_i(s_i, k_i, t_0^i, t_1^i)$, with $s_i \in S, k_i \in P_{s_i}, t_0^i \geq T_0, t_1^i \leq T_f$. We might denote acquisitions simply as a_i, whenever this creates no confusion. Therefore, given the set of satellites, their possible sensor positions and the time frame, let $\mathcal{A} = \{a_1, ..., a_n\}$ be the set of all possible acquisitions.

2) For each $a \in \mathcal{A},\ c_a > 0$ is the cost incurred when using a.

3) The intersection of the elements of the set $\mathcal{A}$ with themselves, and with the region $\mathcal{R}$, defines a set of subregions, whose union is equal to the region of interest. Let $\{R_1, \ldots, R_m\}$ be the set of all such subregions (note that the calculation of this set is a rather intensive task, which is performed by using algorithms of computational geometry).

These definitions are illustrated in Figs. 1 and 2.

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

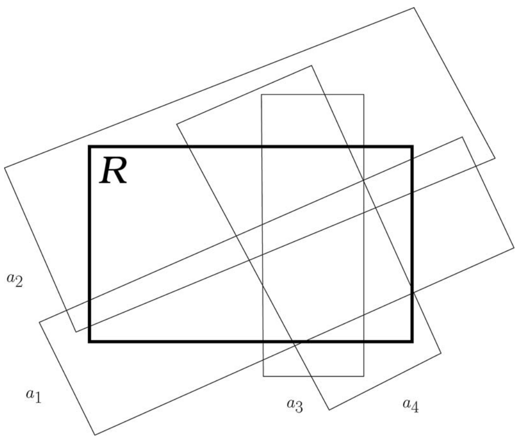

Fig. 1. Simplified example of SAP. Region $\mathcal{R}$ of interest (solid rectangle) is being covered by four acquisitions $\{a_1, a_2, a_3, a_4\}$ sorted in arriving time.

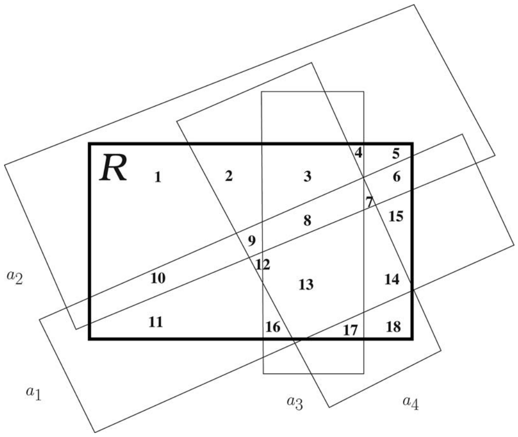

Fig. 2. Subregions generated in $\mathcal{R}$ by intersection of acquisitions with each other and with region of interest. In this simple example, 18 subregions are generated.

1) Each individual satellite s in the selection is not used more than once at any given time instant. This condition can be mathematically expressed as: $\forall\ i, j = 1, \ldots, n,\ s_i = s_j \Rightarrow [t_0^i, t_1^i] \cap [t_0^j, t_1^j] = \emptyset$.

2) If a satellite s in the selection is used more than once with different sensor positions, Δtk_s units of time are needed to change its sensor position. Mathematically,

$$\forall\ i, j = 1, \ldots, n,\ s_i = s_j, k_i \neq k_j \Rightarrow \\ \left[t_0^i, t_1^i + \Delta tk_i\right] \cap \left[t_0^j, t_1^j + \Delta tk_j\right] = \emptyset.$$

With this notation, we are in position to formulate our SAP as follows:

SAP: Find a selection of admissible acquisitions $\{a_i(s_i, k_i, [t_0^i, t_1^i]) \in \mathcal{A},\ i = 1, \ldots, n\}$ such that $\mathcal{R} \subset \cup_{i=1}^n a_i$ at minimum cost. The reader should note that other definitions of objective functions are possible: completion time, number of satellites, area covered more than requested, surface covered having a threshold value for the final time and/or maximum budget, etc. In the example shown in Fig. 1, it is easy to see that acquisition

a_3 is redundant. Therefore, an optimal solution minimizing overall costs is to choose acquisitions a_1, a_2, a_4.

III. AN ILP MODEL FOR SAP

Assume we have a number of EOSs available with steerable sensor, i.e., the sensor angle with respect to the nadir can be changed within a certain range, if necessary. We assume that each EOS must keep the chosen mode for the whole acquisition, while the satellite is scanning the region of interest. Let K be the number of possible sensor positions, N the number of acquisitions, and M the number of subregions generated by the intersections of all available acquisitions and the region of interest $\mathcal{R}$. Let R_1, ..., R_M be the subregions. Parameter q_{ij}^k is defined as one if acquisition i in position k covers subregion j, and zero otherwise, $\forall\ i = 1, \ldots, N, j = 1, \ldots, M, k = 1, \ldots, K$. Let Q be the tridimensional matrix whose entries are the parameters q_{ij}^k. Our SAP is modeled as an ILP program using the following variables:

$$x_i^k = \begin{cases} 1 & \text{if acquisition } i \text{is selected in mode } k, \\ 0 & \text{otherwise} \end{cases}$$

$$\min \quad \sum_{k=1}^{K} \sum_{i=1}^{N} c_i x_i^k \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \sum_{i=1}^{N} x_i^k q_{ij}^k \geq 1, \quad \forall\ j = 1, \ldots, M, \tag{2}$$

$$\sum_{k=1}^{K} x_i^k \leq 1, \quad \forall\ i = 1, \ldots, N, \tag{3}$$

$$x_i^k \in \{0, 1\}, \quad \forall\ i = 1, \ldots, N, \forall\ k = 1, \ldots, K. \tag{4}$$

Equation (1) is the objective to be minimized, the cost of the acquisitions. Constraints (2) ensure that all subregions must be covered by at least one acquisition at one of its possible modes.[2] Constraints (3) ensure that the same acquisition cannot be used in more than one position.

A bottleneck in the previous formulation is the calculation of parameters q_{ij}^k, which requires the calculation of all possible intersections. The complexity entailed by the underlying computational geometry problem might make it very hard (and slow) to explicitly formulate problem (1). To tackle this issue, in the next section, we propose a faster method for selecting a set of acquisitions so that the region of interest is covered, without having to compute Q.

IV. A HEURISTIC ALGORITHM

The need to obtain a good solution in a short time asks for the use of heuristic and/or metaheuristic algorithms. Heuristic algorithms are procedures to rapidly obtain a solution, which is hoped to be (nearly) optimal.

In this section, we design the construction phase of a GRASP algorithm that consists of randomly adding elements to the problem's solution set, out of the set of ℓ elements that individually yield the largest improvement in the objective function, when added to the solution obtained in the previous iteration (the so-called restricted candidate list). This procedure is repeated until a certain stop criterion is met, and each of the (possibly) different obtained solutions forms a set of feasible solutions. Note that, the randomness of the procedure makes it possible that solutions obtained in different runs differ from one another. The final solution, chosen by the heuristic, is the best one out of the feasible solution set obtained in the different runs. When $\ell = 1$, that is, when we choose at each iteration the element that individually yields the largest improvement in the objective function, the procedure obtained is called a greedy algorithm. A pseudocode of our heuristic is given in Algorithm 1.

Algorithm 1 Pseudocode of our heuristic.

```
Data: Input data: $Q$, $\ell$, $\mathcal{R}$, $\{a_1, \ldots, a_N\}$.
Assumption: $\mathcal{R} \subset \cup a_i$.
Set $Reg = \mathcal{R}$, $Aq = \{a_1, \ldots, a_N\}$, $Sol = \{\}$;
while $STOP = False$ do
    if # of elements of $Aq \leq \ell$ then
        Set $Fq = Aq$
    else
        $Fq$ is the set constituted by the $\ell$ acquisitions in
        $Aq$ whose strips individually cover the maximum
        area of $Reg$
    end
    Randomly pick one acquisition $a_j$ in $Fq$;
    $Sol = Sol \cup \{a_j\}$;
    Set $Aq = Aq - \cup \{a_k\}$, where the $\{a_k\}$ encompass
     all different modes of acquisition $\{a_j\}$ (including
     $\{a_j\}$ itself);
    Set $Reg = Reg \setminus S_a$ ($S_a$ is the region covered by
     acquisition $a$);
    if $Reg = \emptyset$ then
        STOP = True
    end
end
Result: A feasible solution $Sol$.
```

This procedure gives a feasible solution to the SAP, Sol_1, i.e., 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. Note that, different runs of this algorithm may yield different solutions, due to the randomization introduced when picking one acquisition in Fq. Let $\{Sol_1, \ldots, Sol_w\}$ be the set of feasible solutions calculated. If $Cost_j$ denotes the cost of solution Sol_j, i.e., $Cost_j = \sum_{i \in Sol_j} c_i$, 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 $\min_{j\ =\ 1,\ \ldots,\ w} Cost_j = Cost_{j*}$.

[2] Note that we could substitute the one in the right-hand side of (2) by a parameter. The modified constraint implies that the jth region is covered at least dj times, thus allowing redundant acquisitions over more important areas.

Fig. 3. Extremadura within Spain.

Fig. 4. Andalucía within Spain.

V. COMPUTATIONAL RESULTS

This section is divided into two subsections. The first one aims at checking whether or not the exact approach is valid for large instances. It also gives a rough estimation of what the value of the parameter ℓ should be. In these preliminary experiments, three geographical regions are tested. The second subsection aims at validating the accuracy of our heuristics over one region in which the exact approach can be run to optimality, and thus the goodness of the heuristic solutions can be tested. All experiments were run on a laptop, Intel Core2 Duo CPU T7100 1.8 GHz and 2 Gigabyte of RAM memory, operating system Windows 7 Professional 32 bits. All algorithms were implemented in Matlab 7.7.0.

For the experiments, a set of real sun-synchronous satellites were considered, and their orbital elements were extracted from the publicly available Space-Track database [27]. To simplify the computations, a spherical model of the Earth was used, and the elements of the satellites were propagated according to the mean J2 propagation model (i.e., using only the secular variations of the mean elements due to oblateness of the Earth). To perform the computational geometry calculations, the Matlab code Polygon Clipper (based on the gpc-library) was used [28].

A. Experiments I

In order to certify the validity of our procedures and the need of heuristics, we performed experiments over three different regions of interest, the three of them located in Spain: the autonomous regions of Extremadura (41 635 km^2, medium size; see Fig. 3) and Andalucía (87 597 km^2, large size; see Fig. 4), and the Andalusian province of Cádiz (7435 km^2, small size; see Fig. 5).

Over each region of interest, we analyzed four possible combinations of satellites:

1) 19 satellites with one mode only;

2) 19 satellites with seven modes, sensor rotating angle 1 degree;

3) 19 satellites with three modes, sensor rotating angle 4 degrees;

4) eight satellites with seven modes, sensor rotating angle 1 degree.

Fig. 5. Cádiz within Andalucía.

We therefore have 3 × 4 = 12 possible scenarios. Over each scenario, we solved the corresponding problem by means of the ILP problem, and the heuristic for values of $\ell = 1, 3, 5, 9, 13$. Note that, the heuristic with $\ell = 1$ coincides with the classic greedy algorithm. The results obtained are presented in Table I. The meaning of the headings in the table is:

1) REGION and SAT are the region and the combination of satellites, respectively.

2) AC is the number of acquisitions considered, and SR the number of subregions, for each of the tested scenarios.

3) SOL_EX and T_EX are the optimal value (number of acquisitions) obtained by the ILP problem, and the computation time needed to calculated such solution, when there was one such optimal solution.

4) T_1 and GAP_1 are the computational time needed by the heuristic with $\ell = 1$, and the gap with respect to the optimal solution measured in percentage as follows:

$$GAP = 100\frac{SOL\_GRASP - SOL\_EX}{SOL\_GRASP}. \quad (5)$$

Gap's can be interpreted as distances to the optimal solution (0% gap means optimality). T_3, GAP_3, T_5, GAP_5, T_9, GAP_9, and T_13, GAP_13 are analogously defined, for the other values of ℓ.

We first note that, in four of the tested scenarios (33%), the proposed ILP problem could not find an optimal solution because the computer ran out of RAM memory. More specifically, these four instances were two of the medium sized areas of interest (Extremadura) and

TABLE I
Results Obtained in the 12 Tested Scenarios over Cádiz, Extremadura, and Andalucía[a]

REGION	SAT	AC	SR	SOL_EX	T_EX	T_1	GAP_1	T_3	GAP_3	T_5	GAP_5	T_9	GAP_9	T_13	GAP_13
CADIZ	1	26	220	6	3.35	2.59	0.00%	1.43	0.00%	1.71	0.00%	2.91	0.00%	2.77	25.00%
CADIZ	2	158	3239	3	767.10	3.54	0.00%	3.92	0.00%	4.52	0.00%	5.65	0.00%	6.25	0.00%
CADIZ	3	76	983	3	14.28	2.22	25.00%	2.12	0.00%	2.28	0.00%	2.68	0.00%	3.16	0.00%
CADIZ	4	74	897	3	10.70	1.69	0.00%	2.15	0.00%	2.36	0.00%	3.41	25.00%	3.10	0.00%
EXTREMADURA	1	32	784	12	2.37	3.73	0.00%	4.47	0.00%	5.89	0.00%	8.94	0.00%	16.09	25.00%
EXTREMADURA	2	281	8723	0	38045.00	7.89	–100.00%	18.39	–100.00%	20.77	–100.00%	28.10	–100.00%	28.05	–100.00%
EXTREMADURA	3	133	6338	6	3502.00	7.89	14.29%	10.04	14.29%	11.72	14.29%	12.77	14.29%	16.27	25.00%
EXTREMADURA	4	136	5021	0	1537.00	7.02	–100.00%	8.83	–100.00%	10.66	–100.00%	15.80	–100.00%	17.25	–100.00%
ANDALUCIA	1	136	5021	0	1537.00	71.21	–100.00%	99.82	–100.00%	114.73	–100.00%	104.60	–100.00%	119.72	–100.00%
ANDALUCIA	2	67	1971	17	28.01	17.56	15.00%	24.48	10.53%	23.92	15.00%	39.50	10.53%	46.61	26.09%
ANDALUCIA	3	136	5021	0	1537.00	36.47	–100.00%	45.02	–100.00%	58.36	–100.00%	62.59	–100.00%	76.00	–100.00%
ANDALUCIA	4	196	5021	11	23.07	27.41	15.38%	37.16	21.43%	58.95	15.38%	47.98	26.67%	54.02	35.29%

[a] When the ILP failed in finding an optimal solution, we set SOL_EX = 0, and GAPs = –100%.

TABLE II
Average Computational Times (in s) and GAPs for Instances in Which ILP Found Optimal Solutions (5)

	ILP	GRASP1	GRASP3	GRASP5	GRASP9	GRASP13
Time	543.86	8.33	10.72	13.91	15.48	18.53
GAP	–	8.71%	5.78%	5.58%	9.56%	17.05%

two of the large sized areas of interest (Andalucía). This fact justifies the need of heuristics algorithms. Table II shows the average computational times and average GAPs for the six algorithms tested (ILP, and the heuristic with five different values of ℓ) in the scenarios in which the ILP was able to find an optimal solution. A first observation from this table is that, even in the instances in which ILP managed to find optimal solutions, the average computational times of the heuristics are inferior by two orders of magnitude. Logically, the time spent by GRASP to calculate one solution increases with the value of ℓ (more potential solutions are evaluated). The GAPs are quite controlled, being the heuristics with $\ell = 3$ and $\ell = 5$ being the lowest of them (around 5% from the optimal solution, on average). Note that GRASP's performance improves if $\ell > 1$, except for the cases in which ℓ is too large. A reason for this poor performance for larger values of ℓ is that the increasing number of possible solutions means that the algorithm might choose solutions that are far from optimal. A reason for the larger GAPs for $\ell = 1$ is given by the (general) bad quality of the solutions given by the greedy algorithm, see [29].

B. Experiments II

Once we have checked that the ILP model might fail in calculating a solution on medium-large areas of interest, we here report on the experiments performed over one small geographical area (comparable with Cádiz) independent of the other three, so we guarantee that the ILP model can find the optimal solution in a reasonable amount of time, allowing us to therefore test the goodness of the solutions obtained by the heuristic. The chosen area is the Spanish autonomous region of Asturias (10 604 km^2, see Fig. 6).

Fig. 6. Asturias within Spain.

Also, because we observed that neither $\ell = 1$, nor large values of ℓ, yielded good results, we only tested our heuristic for values $\ell = 2, 3, 4$. We tested our procedures for five different number of satellites (8, 9, 10, 11, and 12) and three possible number of modes for each satellite (3, 4, and 5), having this way 5 × 3 = 15 different instances. The obtained results are shown in Table III. The meaning of the column headings in the table is:

1) SAT and MOD denote the number of satellites and the number of sensor modes for each satellite, respectively, while N_AD is the number of acquisitions.
2) T_EX, T_2, T_3, T_4, is the computational time to obtain a solution for each algorithm.
3) GAP_2, GAP_3, GAP_4 are the corresponding gaps when compared with the solution obtained by the ILP model, which is denoted as Z_EX.

In Table IV we have the average computational times and average GAPs.

After checking the normality hypotheses by means of a Kolmogorov-Smirnov test (see last row in Table IV), we noticed that the computational times of the four algorithms here tested, on this region of interest, can be considered normally distributed. An analysis of variance (ANOVA,

TABLE III
Results Obtained in the 15 Tested Scenarios over Asturias[a]

SAT	MOD	N_AD	T_EX	Z_EX	T_2	GAP_2	T_3	GAP_3	T_4	GAP_4
8	3	48	5.76	5	4.29	0	2.52	0.2	3.21	0
9	3	55	10.98	5	3.42	0	3.4	0.2	4.78	0
10	3	59	8.72	5	4.03	0	3.98	0	4.73	0.2
11	3	64	13.26	5	3.8	0	4.48	0	5.23	0.2
12	3	72	25.64	5	6.08	0	5.17	0	5.57	0.2
8	4	81	56.46	5	4.46	0	5.25	0	5.52	0.2
9	4	93	127.31	5	5.37	0	5.99	0	6.51	0.2
10	4	101	180.02	5	9.04	0	7.04	0	7.7	0.2
11	4	114	316.14	5	7.42	0	7.47	0	7.71	0
12	4	124	532.78	5	9.05	0	12.6	0	8.83	0
8	5	108	299.14	5	6.64	0	6.07	0	7.45	0
9	5	124	623.84	5	8.08	0	7.96	0	9.17	0
10	5	139	1133.59	5	8.81	0	9.11	0	10.14	0
11	5	154	1956.09	5	9.67	0	10.17	0	11.18	0.2
12	5	168	3093.68	5	11.16	0	11.53	0	12.07	0

[a] The ILP model obtained the optimal solution in the 15 scenarios tested.

TABLE IV
Average Computational Times (in s), GAPs for Experiments over Cádiz (5), and p-Values of Kolmogorov-Smirnov Normality Tests for the Computational Times of Each Algorithm

	ILP	GRASP2	GRASP3	GRASP4
Time	558.89	6.75	6.85	7.32
GAP	–	0%	2.67%	9.32%
p-value	0.21	0.86	0.94	0.88

see [30]), taking as dependent variable the computational times, and as unique factor the type of algorithm, yielded a $p = .0015$. This means that, at 95% confidence level, we cannot admit that the average computational times are equal. Actually, the average computational times using the heuristic are two orders of magnitude lower than the average computational time of the ILP model (558 s against 6–7 s). At the same level of confidence, we cannot reject that the average computational times of the three heuristics are the same ($p = .827$), but we do reject that the average GAPs are the same ($p = .0031$), with significant differences between $k = 2$, 3 and $k = 4$. In other words, the heuristics with $\ell = 2$, 3 yielded statistically significant better results than the heuristic with $\ell = 4$ in terms of quality of solutions. Note that, the heuristic with $\ell = 2$ obtained the optimal solution in all instances tested.

VI CONCLUSIONS

In this paper, we have introduced several algorithms to solve the SAP, which consists of selecting a number of satellites in order to cover a given region of interest at a minimum cost. Such problem's complexity is increasing continuously, as a result of the ever raising number of observation satellites, which makes EOS management more and more complicated. Our procedures have proven very effective for 24 realistic scenarios, over three different regions of interest. The exact approach proposed, an ILP, failed in computing a solution for several scenarios (when the number of satellites, and subregions generated by the exact algorithm, is too large). Therefore, the introduction of a heuristic approach (a fast algorithm which gives good solutions although not necessarily optimal) is justified. We have also introduced the construction phase of a GRASP algorithm. In a second step of our computational experience, we tested the accuracy of our GRASP algorithms over instances that could be solved to optimality by means of the ILP model. The results showed that, the heuristics proposed, yielded good solutions (most of the times optimal) in fairly lower computational times than the exact approach. Future research includes extending the heuristic algorithm to include more complex and realistic constraints such as instrument slew times, onboard storage limitations and downlink opportunities.

ACKNOWLEDGMENT

The authors want to thank Professor Emilio Carrizosa for fruitful discussions and Sergio Rubio Madroñal for his help during the experiment phase. Special thanks are due to Taitus software and its Founder and CEO (Felipe Martín Crespo) for posing this problem to us during a modeling week organized by the Institute of Mathematics of the University of Sevilla.

REFERENCES

[1] Chien, S., Smith, B., and Rabideau, G.
Automated planning and scheduling for goal-based autonomous spacecraft.
IEEE Intelligent Systems and Their Applications, **13**, 13 (1998), 50–55.

[2] Muraoka, H., Cohen, R., Ohno, T., and Doi, N.
Aster observation scheduling algorithm.
In: *Proceedings of SpaceOps*, Tokyo, Japan, 1998.

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

[4] Chien, S. A., Johnston, M., Frank, J., Giuliano, M., Kavelaars, A., Lenzen, C., and Policella, N.
A generalized timeline representation, services, and interface for automating space mission operations.
In: *SpaceOps 2012 Conference.* Stockholm, Sweden, 2012.

[5] Wang, P., Reinelt, G., Gao, P., and Tan, Y.
A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation.
Computer & Industrial Engineering, **61** (2011), 322–335.

[6] Beaumet, G., Verfaille, G., and Charmeau, M. C.
Decision-making onboard an autonomous agile earth-observing satellite.
Computational Intelligence, **27** (2011), 123–139.

[7] Lemaitre, M., Ferfaillie, G., Jouhaud, F., Lachivier, J., and Bataille, N.
Selecting and scheduling observations of agile satellites.
Aerospace Science and Technology, **6** (2002), 367–381.

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

[9] Galán-Vioque, J., Vázquez, R., Carrizosa, E., Vera, C., Perea, F., and Martín, F.
Towards a visual tool for swath acquisition planning in multiple-mission eoss.
In: *IWPSS 2011Workshop Proceedings*, Darmstad, Germany, 2011.

[10] Bazaraa, M., Jarvis, J., and Sherali, H.
Linear Programming and Network Flows. New York, NY: Wiley, 1990.

[11] Wolsey, L. A.
Integer Programming. New York: Wiley, 1998.

[12] Karp, R.
Reducibility among combinatorial problems.
In: R. Miller, J. Thatcher (eds.) Complexity of Computer Computations, New York, NY: Plenum Press, 1972, 85–103.

[13] Michalewicz, Z., and Fogel, D. B.
How to Solve It: Modern Heuristics. Berlin: Springer Verlag, 2000.

[14] Vazirani, V.
Approximation Algorithms. Berlin: Springer Verlag, 2001.

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

[16] Taitus Software. SaVoir: Swath Acquisition Viewer (2011).
[Online] www.taitussoftware.com/ (2011)

[17] Cordone, R., Gandellini, F., and Righini, G.
Solving the swath segment selection problem through Lagrangian relaxation.
Computers & Operations Research, **35** (2008), 854–862.

[18] Elía, E. D., Oglietti, M., Masuelli, S., and Romero, E.
Planning of satellite images applied to early warning hydrological models.
In: *7th IWPSS 2013Workshop Proceedings*, 2013.

[19] Gabrel, V., and Vanderpooten, D.
Enumeration and interactive selection of efficient paths in a multiple criteria for scheduling an earth observing satellite.
European Journal of Operational Research, **139** (2002), 533–542.

[20] Frank, J., Jónsson, A., Morris, R., and Smith, D. E.
Planning and scheduling for fleets of earth observing satellites.
In: *Proceedings of Sixth International Symposium on Artificial Intelligence, Robotics, Automation & Space*, 2001.

[21] Bianchesi, N., and Righini, G.
Planning and scheduling algorithms for the cosmo-skymed constellation.
Aerospace Science and Technology, **12** (2008), 535–544.

[22] Globus, A., Crawford, J., Lohn, J., and Pryor, A.
Scheduling earth observing satellites with evolutionary algorithms.
In: *International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, 2003.

[23] Bianchessi, N., Cordeau, J. F., Desrosiers, J., Laporte, G., and Raymond, V.
A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites.
European Journal of Operational Research, **177** (2007), 750–762.

[24] Vallado, D., and McClain, W.
Fundamentals of Astrodynamics and Applications (3rd ed.). El Segundo, CA: Microcosm Press/Springer, 2007.

[25] Curtis, H. D.
Orbital Mechanics for Engineering Students (2nd ed.). Oxford: Butterworth-Heinemann, 2009.

[26] Barret, E., and Curtis, D.
Introduction to Environmental Remote Sensing (3rd ed.). New York: Springer, 1992.

[27] Space-Track.org. [Online]. http://www.space-track.org/.

[28] Holz, S.
Polygon Clipper for Matlab (2006). [Online] http://www.mathworks.com/matlabcentral/fileexchange/8818-polygonclipper.

[29] Bang-Jensen, J., Gutin, G., and Yeo, A.
When the greedy algorithm fails.
Discrete Optimization, **1**, **2** (2004), 121–127.

[30] Montgomery, D., Runger, G., and Hubele, N. F.
Engineering Statistics (5th ed.). Hoboken, NJ: John Wiley & Sons, 2011.

Federico Perea graduated with a B.S. degree in mathematics in 2001 from the Universidad de Sevilla; in 2007, he obtained his Ph.D. degree in applied mathematics from the Universidad de Sevilla. He is an associate professor at the Universitat Politècnica de València since October 2010 in the department of Applied Statistics and Operations Research, and Quality. His academic career has previously developed at the Universidad de Sevilla and Universidad de Zaragoza. Out of academia, he has also worked as a researcher for Thales in 2005–2006. His research focuses on transportation planning, game theory, scheduling, and other related areas within operations research. So far, he has coauthored over 20 research papers, which have been published in internationally recognized journals, and contributed to many conferences within his areas of interest.

Rafael Vazquez (S’05-M’08) received the M.S. and Ph.D. degrees in aerospace engineering from the University of California, San Diego, and B.S. degrees in electrical engineering and mathematics from the University of Seville, Spain. He is currently an associate professor in the Aerospace Engineering Department, University of Seville. His research interests include control theory, distributed parameter systems, and optimization, with applications to flow control, ATM, UAVs, and orbital mechanics. He is coauthor of the book Control of Turbulent and Magnetohydrodynamic Channel Flows (Birkhauser, 2007). He currently serves as Associate Editor for Automatica.

Jorge Galán-Vioque is professor of mathematics at the Applied Mathematics Department and subdirector of the Institute of Mathematics of the University of Sevilla, Spain. He graduated with a B.S. in physics at the Autónoma University of Madrid, Spain, in 1989 where he also obtained his Ph.D. degree in 1993. He was awarded a Fulbright Fellowship for a post-doctoral stay at Ohio State University (USA). He has coauthored more than 50 research papers and the book *Numerical Continuation Methods for Dynamical Systems*, Springer 2007. His research interests include solid state theory, Hamiltonian dynamical systems with application to quantum and classical mechanics, and, recently, he has been involved in industry applied projects combining optimization and orbital mechanics.