Mission planning for a non-homogeneous Earth observation satellite constellation for disaster response

Nicolas Holvoet*, Wasanchai Vongsantivanich†, Supatcha Chaimatanan‡, Daniel Delahaye§

This article explores the mission planning problem for a constellation of non-homogeneous Earth observation satellites for disaster response to promote cooperation in the Asia-Pacific region between different space agencies or satellite operators. Given a set of weighted requests consisting of a polygonal area, a list of collaborating satellites and a time horizon, the goal is first to compute the observation opportunities and then to plan a sequence of acquisitions for each sensor to optimize the total coverage in the first days to help the disaster management agencies. The non-homogeneous satellites (non-agile or agile) and the different orbits result in a huge combinatorial search space. A simplified computational method is proposed to generate the observation opportunities and a metaheuristic based on the Simulated Annealing algorithm is used to solve the mission planning problem. Real requests from disaster management agencies are used to evaluate the proposed methodology.

I. Introduction

Space-based Earth observation has become a common practice to get invaluable information about Earth and human activities. For disaster mitigation and disaster management, prompt remote sensing data from space covering the affected area is indispensable for planning, analyzing the situation for disaster relief, post-disaster management or even future disaster prevention. Having more data, both spatially and temporally, will undeniably lead to clearer situation assessment and more effective decision-making.

Considering using a single remote sensing satellite, there are significant limitations in spatial coverage, temporal resolution and varieties of data to assimilate to perform further analysis. With the current global trends in space utilization, it can be seen that the number of satellites launched into orbit has increased progressively every year. To maximize the benefits from these space resources, constellations of satellites are designed. However, most of the Earth observation satellites (EOS) that are currently working in space are individually operated by different countries or agencies. Therefore, acquisition plan for each satellite is also individually performed based on each acquisition request that is submitted to each particular operator. For this reason, there are many initiatives to collaborate and coordinate satellite utilization among the satellite operators, for example: CEOS's Virtual Constellation, International Charter, Disaster Monitoring Constellation, etc. "Sentinel Asia" is one of these initiatives, established in 2005 to promote international cooperation to prevent, mitigate and respond to natural disaster damage in the Asia-Pacific region by using space technology. In case of a major event, it aims to allow collaboration between space agencies to form a constellation of different types of Earth observation satellites which will perform a mission together.

The Geo-Informatics and Space Technology Development Agency (GISTDA) has developed a platform called OPTEMIS (Operation Planning Tool for Earth observation MISsion) to support the Sentinel Asia initiative. The goal of this tool is to facilitate and improve the operational workflow of the space Data Provider Nodes (DPN). This platform will be used as an intermediary to collect imaging requests from different users, perform feasibility assessment, manage the mission plan and track the mission progress, which will help in synergizing closer collaboration between the stakeholders (rescue and disaster response teams, Data Provider Nodes, Data Analysis Nodes, program coordinator, etc.).

We present below some general considerations concerning the problem, including the goals and the main constraints. Section II gives an overview of the previous work done on similar problems. The model and the acquisition feasibility assessment module are introduced in section III. We present in section IV the mission planning and task allocation algorithm that is used to compute acquisition requests for each collaborating satellite. Finally, section V demonstrates the benefit of the proposed system on real-world scenarios.

*Master's degree student, nicolas.holvoet@alumni.enac.fr, École Nationale de l'Aviation Civile, 7 avenue Édouard Belin, 31055 Toulouse, France

†Satellite engineer, wasanchaiv@gistda.or.th, Geo-Informatics and Space Technology Development Agency, 120 The Government Complex, Chaeng Wattana Road, Lak Si Bangkok 10210, Thailand

‡Researcher, supatcha@gistda.or.th, Geo-Informatics and Space Technology Development Agency, 120 The Government Complex, Chaeng Wattana Road, Lak Si Bangkok 10210, Thailand

§Researcher, delahaye@recherche.enac.fr, École Nationale de l'Aviation Civile, 7 avenue Édouard Belin, 31055 Toulouse, France

A. Agile satellites

In this paper, we consider a constellation of non-homogeneous satellites.

Non-agile satellites have no maneuvering ability and the area they can acquire is determined by their instantaneous position on the orbital trajectory. Semi-agile satellites have one degree of freedom on the roll axis: the instrument can rotate to extend its *field of regard* (FOR) on the left and/or on the right side of the satellite's trajectory. This oblique viewing ability increases the imaging opportunities of a given area during an orbital cycle by reducing its revisit delay. Agile satellites enable off-nadir access by moving along their roll axis like semi-agile satellites but also along their pitch axis to point ahead or backward. Thus, the starting date of the acquisition can be freely decided in a time interval which depends on the minimum and maximum pitch angles allowed. This extended *time window* allows to schedule more observations which would not always have been feasible during a given orbit by a non-agile or semi-agile satellite as shown in Figure 1.

For both non-agile and (semi-)agile satellites, a minimum transition duration must be computed between two candidate observations to check if they can be successively acquired with enough time for the processing (compression, storage, downlink. . .) and maneuvering actions (changes in roll and pitch angles). Naturally, image quality strongly depends on the pitch and roll angles and could be a hard or soft constraint from the user.

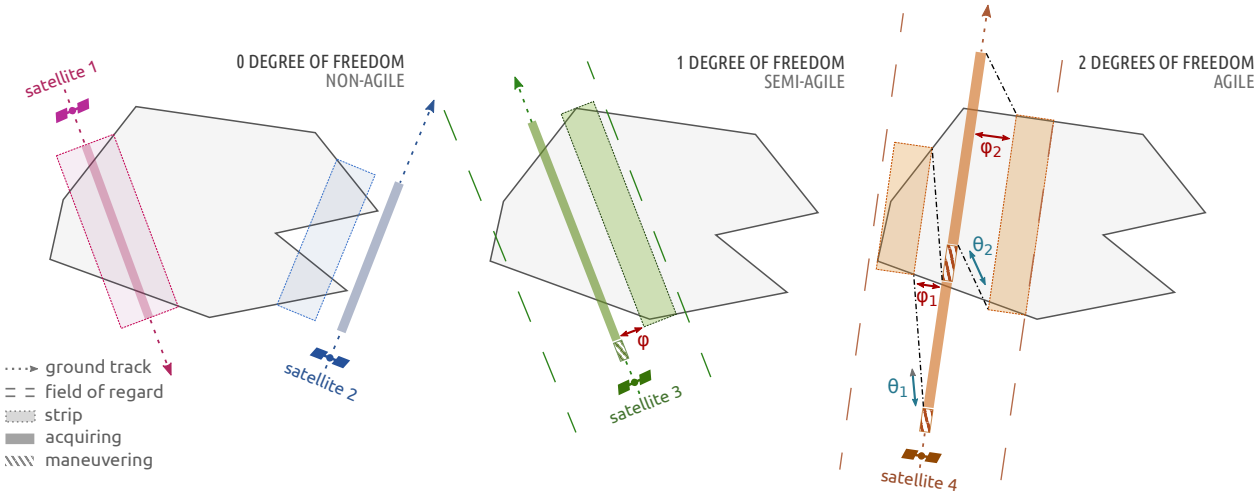


Fig. 1 Illustration of the different types of agility. Non-agile satellites can have a fixed roll offset, meaning that their observations are not centered on their ground track: *satellite 2* has a left-side viewing ability. The pitch agility of *satellite 4* allows two observations in one orbit pass.

B. Problem statement

In case of a major disaster, authorized users will be able to send requests, via a Web interface of OPTEMIS system, by selecting the suitable satellites according to the type of the disaster in terms of sensor type and resolution (e.g. optical, radar, low, medium or high resolution). Then, our goal will be to make the most of these sensors to meet the needs of decision makers and search and rescue teams in the first days following the event. Eventually, an optimized acquisition tasking order will be particularly distributed to every collaborating satellite operators and/or space agencies. This work can be addressed as a multi-acquisitions by non-homogeneous satellites mission planning problem, where the inputs are requests defined by a polygonal region of interest (ROI), a set of satellites which will collaborate, a time horizon and a weight. The weights allow us, for instance, to give priority to urban areas over uninhabited ones. The model seeks to maximize the weighted-covered surface over the time horizons requested by the users while insuring the feasibility of the mission and a sufficient image quality. The non-homogeneous satellites and their different constraints (sensor type, field of view, satellite agility and maneuverability, and other particular satellite constraints) result in a complex combinatorial problem.

In this work, we suppose that each satellite operator considers the disaster requests as highest priority, and that the ROIs are relatively small. Therefore, it can be assumed that the constraints of on-board memory, energy, and ground station availability are satisfied. Other considerations concerning the fairness of resource sharing and the task distribution equity may be more appropriate for post-disaster monitoring of wide areas (non-urgent tasks).

For each satellite, we will need to predict its orbit passes over the requested areas and compute its observation opportunities based on its viewing characteristics. Then, we will try to plan acquisitions for each satellite by optimizing the whole mission with a metaheuristic optimization method. This process is suitable for the problem of co-constellation mission planning for disaster response, according to the general assumption for urgent disaster acquisitions that these tasks shall be put at the highest priority and shall override the existing plan in the acquisition queue.

II. Literature review

A lot of research has already been done on mission planning and scheduling for a single (agile) satellite and some other research papers deal with a cooperative satellite constellation. [1] proposed a multi-objective evolutionary algorithm to address four criteria: total number of acquisitions, total profit, image quality (minimizing the observation angles) and load balancing of satellites. In [2], a constructive planning and scheduling algorithm with look-ahead and back-tracking abilities was developed for the COSMO-SkyMed constellation. The goal was to maximize the number of images taken and transmitted regarding downlink operational constraints. [3] split the request into a set of elementary areas that a satellite can acquire in a single shoot and solved the problem with a self-adaptive multi-agent system. In [4], the problem was decomposed into a task assignment main problem and a single-satellite scheduling sub-problem. A Simulated Annealing heuristic was used for the single-satellite scheduling problem and the results were passed on an ant colony optimization algorithm to guide the assignment search process. A more comprehensive and comparative literature review can be found in [5]. However, some of these works rely on a predefined set of strips without a clear indication on how they were computed, and so address only one part of the problem. In some papers, the authors split the area to cover in regular and contiguous strips just once, without considering the dynamic nature of the overlapping optimization problem over the days, which leaves no place for further optimization and can't be applied for a constellation of non-homogeneous satellites.

To the best of our knowledge, relatively few studies have investigated a computational model for a problem similar to ours. In [6], the authors used the circle intersection point set (CIPS) to derive a set of key representative geographic locations, similar to the facility location problem, based on the characteristics of the most restrictive sensor. This approach hugely reduces the combinatorial complexity; however, they rely on a preexisting module which gives the best acquisition opportunities for each interpolated point. Their optimization model also addressed weighted spectral and spatial resolution and off-nadir angles. They solve the problem as a set covering problem (SCP) with a branch-and-bound algorithm. In [7], they extended the model to handle multiple polygonal requests with different weights. [8] proposed a dynamic segmenting algorithm for the multi-request problem for semi-agile satellites: for each orbit pass which can intersect with a requested area, they built a list of possible acquisitions by discretizing the roll angle depending on the sensor's characteristics. Then, they developed a greedy heuristic which also handles the dynamic arrival of new emergency tasks for a robust rescheduling. [9] proposed a model for the multi-request mission planning problem including download windows, which was evaluated for the environmental and disaster monitoring in China as part of the CRESDA program. Their model can be used for both spot and polynomial weighted requests. For agile satellites, they discretized the pitch angle and retain three time windows ($-\theta_{max}$, 0, $+\theta_{max}$) whereas how they handled the continuous roll angle to generate the possible acquisitions for each orbit pass is not clearly explained. Also, their model didn't allow more than one strip's acquisition for a given request in each orbit pass, which can lead to a sub-optimal use of agility but reduces the complexity of the problem. The authors then developed a heuristic with good computational performances whose main idea "is to assign the most conflict-free observation time window to the most desired request, supplemented by limited backtracking". In [10], the authors addressed the geometrical problem of modeling the ground observation opportunities and demonstrated how a "temporal series of re-centered map projections" can be used to compute the strips. Other interesting papers are [11] and [12] which addressed the target discretization with a grid-based approach which reduces the calculation of overlapping areas by computing intersections of grid point sets. Finally, [13] gave some interesting insights on the neighborhood structure for agile satellite scheduling and defined six removal operators and three insertion operators.

III. Problem modeling

The mission planning problem we aim to address is complex due to the non-homogeneous orbits, the mix between non-agile and (semi-)agile satellites and their different kinematics. Therefore, we develop a simplified model which responds to the basic requirements of our problem.

A. Assumptions

To reduce the complexity of the problem, the following assumptions are made:

1. Requests and strips

For Earth observation missions, there are two main types of requests: spot or polygon. Usually, a spot can be acquired in a single shot whereas a polygon needs to be split into smaller observations (strips) which will be merged. In this paper, we consider polygon-shaped requests and we also handle non-convex and overlapping requests.

Strips are chosen parallel to the satellite's ground track to avoid costly maneuvering transition between successive shots which would result in fewer acquisitions ([16]). As we support overlapping requests, strips are not attached to one specific request and can cover several requests.

2. Image acquisitions

We only consider full-strip acquisitions (the satellite will continuously acquire the strip from one edge of the polygon to the other one). It could mean that the pitch agility will be underexploited in some scenarios. One reason to avoid partial acquisitions of strips and the overuse of agility is that having too many different observations could result in a non-homogeneous quality and a lot of small, scattered acquisitions which are less valuable than continuous ones.

B. Input data: requests

We note the set of available satellites $K = \{k_1, ..., k_{N_K}\}$. The input data is a set of requests $R = \{r_1, ..., r_{N_R}\}$, each request r_i consisting of:

- a polygonal region of interest (ROI) with lat/long coordinates: $ROI_{LatLng}(r_i)$;
- a time interval for the mission: $[T_{start}(r_i); T_{end}(r_i)]$;
- a set of sensors which will collaborate to perform the mission together: $K(r_i) \subseteq K$;
- a weight: $w(r_i)$.

The problem will ultimately rely on geometric computations such as the intersection and the union of polygons, which are computationally expensive tasks. For efficiency, we decide to discretize each ROI with a grid-based approach ([11]). We first project the polygonal ROI in the *Lambert azimuthal equal-area* space centered at the centroid of the ROI. Then, we compute a discrete set of points spaced by `gridSpacing` km covering the ROI. This parameter must be chosen carefully to reach a good balance between accuracy and performance, considering the size of the requests and the average swath width of the sensors (e.g. 10 km for very large requests with an average swath width of 25 km should be enough to compensate the side effects of discretization, whereas we would require 3 km for smallest ROIs).

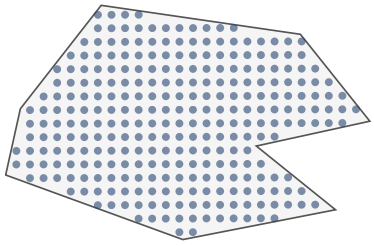


Fig. 2 Discretization of the ROI

We note $Points(r_i)$ the set of grid points for the request r_i.

N.B.: Even if two requests overlap, each request will still have its own set of points.

C. Strips modeling and computation

From these requests, we want to generate all the acquisition opportunities for the constellation.

We note $S(k) = \{s_{k1}, ..., s_{kN_{S(k)}}\}$ the set of strips for the satellite k. Each strip s_{ki} will be modeled by:

- A polygon with lat/long coordinates corresponding to the ground projection of the Earth surface it covers: $Strip_{LatLng}(s_{ki})$;
- A roll angle: $\phi(s_{ki})$;
- A duration: $d(s_{ki})$;

- The discretized coverage of the ROIs, i.e. for each request r_i, the set of the ROI's points it covers: $\forall r_i \in R, Points_{r_i}(s_{ki}) \subseteq Points(r_i)$;
- A list of time windows $TW(s_{ki})$, each time window $s_{ki}[t]$ is defined by:
  - A starting date: $T_{start}(s_{ki}[t])$;
  - A pitch angle: $\theta(s_{ki}[t])$;
  - A final pitch angle: $\theta_f(s_{ki}[t])$;
  - A quality score: $quality(s_{ki}[t])$.

The angles are discretized with a tunable step for each satellite (typically, 1°). Non-agile or semi-agile satellites will have only one time window (0° pitch angle).

We define an *acquisition* as a strip s_{ki} and one of its time window $s_{ki}[t]$. For simplicity purposes, $s_{ki}[t]$ will be used to denote the acquisition, assuming it is also possible to retrieve the attached strip from this notation.

1. Orbit propagation and strip building

We describe below the algorithm used to build the strips for a given sensor. It is based on a fast propagation mode used to determine when a satellite intersects an ROI and a slow propagation mode used to build precise strips when an intersection is detected.

We first determine the set of requests the sensor is attached to and we merge their date intervals so that we know the useful timespan for propagation. We initialize the time `increment` of the propagation to 30 seconds.

1) We compute the lat/long ground position of the satellite by using an SPG4 library and its TLE. If the sensor is optical and is not in sunlight, go to the next date.
2) For each request attached to this sensor whose time horizon contains the current date:
   - We compute the FOR of the sensor between the previous and the current position (at nadir, based on its maximum roll angle and its FOV angle) in the *Lambert azimuthal equal-area* euclidean space centered at the ROI's centroid.
   - We detect intersection between the FOR and the ROI in this space.
3) If the FOR intersects at least one request: if `increment` was 30 seconds, we decrease it to 0.5 seconds (or less if the ROIs are small) and go back to 1) at the previous date, else, for each discretized roll angle:
   - We compute the "corridor" coordinates (boundaries of the strip at this position) based on the roll and the FOV angles in the *Azimuthal equidistant* space centered at this ground position ([10]).
   - We project these coordinates back in lat/long.
   - For each intersecting request detected in 2):
     - We project the lat/long corridor coordinates in the *Lambert azimuthal equal-area* space centered at the ROI's centroid.
     - We detect intersection in this space, if there is one:
       * If the strip was updated more than `minGap` seconds ago, we decide to build a new strip. First, we finalize the last strip (compute the discretized coverage of the ROIs it intersects and the list of its time windows) and then we create a new strip.
       * We extend the strip by adding the two corridor coordinates.
4) If the FOR intersects no request: if we were in slow mode, set `increment` back to 30 seconds.
5) Increment the date by `increment` and go back to 1).

Figures 3, 4 and 5 describe the propagation and strip building process.

We use the `minGap` parameter to detect the end of a strip so as to decide whether we extend the strip or create a new one in case there was no intersection for some time. Indeed, for a given roll angle, if the last intersection comes from the previous orbit pass, from another non-overlapping request or from a previous position in a large non-convex ROI, we may decide it is better to switch off the sensor between two acquisitions.

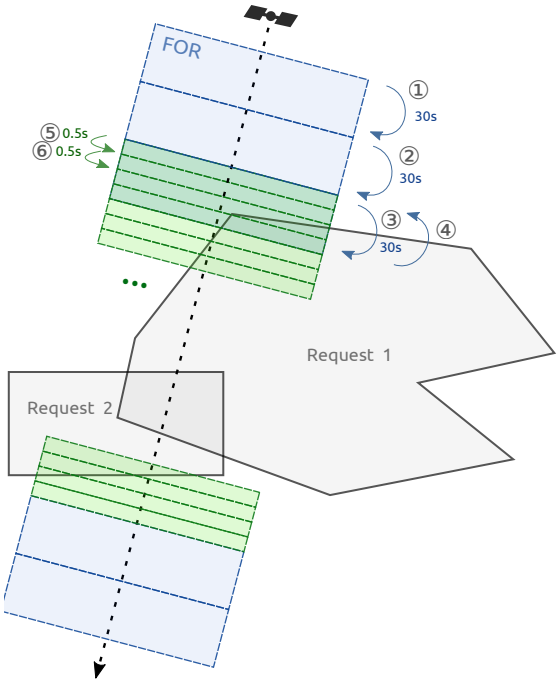


Fig. 3 Orbit propagation: acquisition opportunities are identified by intersecting the FOR and the requests between two successive positions. As soon as there is one intersection, we go back to the previous position and start again in slow mode.

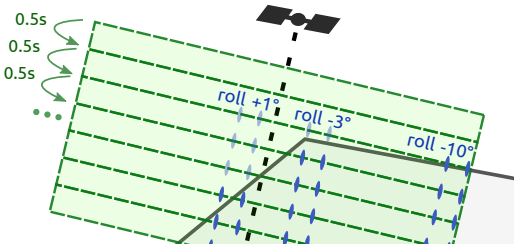


Fig. 4 In slow mode, when the FOR intersects at least one request, we discretize the roll angle and if there is one intersection for a roll angle, we extend the corresponding strip. Example is given for three arbitrary roll angles.

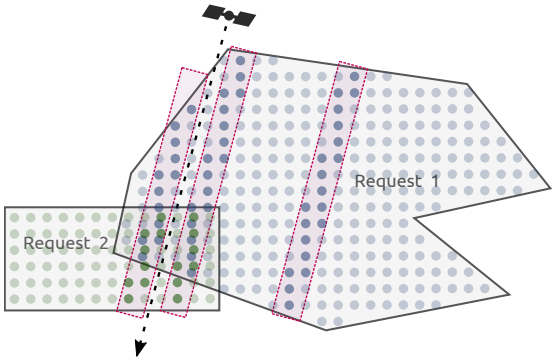


Fig. 5 Strips are finalized when there is no more intersection with the requests. The coverage of a strip is, for each request, the set of points it covers.

2. Time windows

First, we compute the duration of the strip's acquisition based on the characteristics of the sensor. Then, we can compute the time windows by discretizing the pitch angle. For each possible pitch angle, we can approximate the starting date of the acquisition with basic geometry and compute the final pitch angle knowing the kinematics of the sensor.

Finally, we define the quality of an acquisition as a decreasing function of the roll and the initial pitch angles:

$$quality(s_{ki}[t]) = 1 - \frac{|\phi(s_{ki})| + |\theta(s_{ki}[t])|}{(|\phi| + |\theta|)_{max,K}} \tag{1}$$

where $(|\phi| + |\theta|)_{max,K}$ is the worst distortion angle that could be produced by a sensor of the constellation. Thus, the quality lies between 0 (worst) and 1 (best).

N.B.: We assume that requests are covered by sensors with similar resolution so that the value of the acquisition can be evaluated with its quality score. Otherwise, we should extend this quality function and consider the resolution.

3. Successive feasibility and strips clustering

For each satellite k, we assume that we have a function to get the minimum maneuvering duration between two successive acquisitions {strip s_{ki} with time window $s_{ki}[t]$} and {strip s_{kj} with time window $s_{kj}[u]$}. Then the physical feasibility constraint can be written:

$$feasibility(s_{ki}[t], s_{kj}[u]) \quad := \quad T_{start}(s_{ki}[t]) \; + \; d(s_{ki}) \; + \; maneuvering_k(s_{ki}[t], s_{kj}[u]) \quad \leq \quad T_{start}(s_{kj}[u]) \tag{2}$$

Two strips are said to be in conflict if the feasibility constraint is violated for at least a pair of time windows (typically, the last time window of one strip and the earliest time window of the other strip). For each sensor, we cluster its strips so that we know for sure that there won't be any feasibility conflict between a strip from one cluster and a strip from another cluster. In most single-request scenarios with a straightforward geometry, a cluster would correspond to one orbit pass.

We note $C(k) = \{C_{k1}, ..., C_{kN_{C(k)}}\}$ the set of clusters for the sensor k, with $C_{kc} \subseteq S(k)$.

D. Decision variables

For each satellite k, for each strip $s \in S(k)$, we can choose to plan zero (denoted by $\emptyset$) or one of its time windows, denoted by its index in $TW(s)$. Therefore, we introduce the decision variables:

$$\forall\, k \in K,\ \forall\, 1 \leq i \leq N_{S(k)},$$

$$x_{ki} = \begin{cases} \emptyset & \text{if strip } s_{ki} \text{ is not planned} \\ t & \text{if strip } s_{ki} \text{ is selected with time window } s_{ki}[t] \end{cases}$$

E. Objective

For each point $p \in Points(r)$ belonging to the request r, we note $Covering(p)$ the set of the planned acquisitions – coming from any sensor – that cover this point:

$$Covering(p) = \bigcup_{k \in K} \bigcup_{\substack{s_{ki} \in S(k), \\ p \in Points_r(s_{ki}), \\ x_{ki} \neq \emptyset}} s_{ki}[x_{ki}] \tag{3}$$

We introduce the $qualityCoverage$ score of a point p as a function of the best quality that can be achieved by one of the planned acquisitions covering this point:

$$qualityCoverage(p) = \begin{cases} 0 & \text{if } Covering(p) = \emptyset \\ f\left(\max\limits_{acq \in Covering(p)} quality(acq)\right) & \text{otherwise} \end{cases} \tag{4}$$

where f is a tunable increasing function to balance the quality and the coverage, for instance a linear function with positive slope and y-intercept. Special cases are $f = 1$ where every covered point will have the same score regardless of the image quality and $\forall q, f(q) = \alpha q + \beta$ with $\beta < 0$ where worst quality leads to negative scores which will act as a soft constraint on the quality.

In this research, we are interested in short-term disaster response. The main consideration is maximizing the covered surface while optimizing the quality of the acquisitions, taking into account the weights of the requests. Therefore, we propose to maximize the following objective function:

$$\max \sum_{r \in R} \left(w(r) \times \sum_{p \in Points(r)} qualityCoverage(p) \right) \tag{5}$$

This objective function is designed to guarantee a *first* maximized coverage of the area in the first days following a major event. It is not intended to be used for monitoring as the $qualityCoverage$ score doesn't carry sufficient information for such purposes and cannot handle scoring for lower-quality and full-overlapping strips which won't improve the objective.

F. Constraints

First, we must ensure that the decision vector contains only valid time windows:

$$\forall\ k \in K,\ \forall\ 1 \leq i \leq N_{S(k)},\ x_{ki} \in \{\emptyset\} \cup [\![1, \#TW(s_{ki})]\!] \tag{6}$$

The only constraint we consider is the intra-cluster feasibility in terms of maneuvering as described in 2:

$$\begin{aligned} &\forall\ k \in K,\ \forall\ C \in C(k),\ \forall (s_{ki}, s_{kj}) \in C^2,\ i \neq j, \\ &x_{ki} \neq \emptyset,\ x_{kj} \neq \emptyset,\ T_{start}(s_{ki}[x_{ki}]) \leq T_{start}(s_{kj}[x_{kj}]), \\ &\text{ensure } feasilibity(s_{ki}[x_{ki}], s_{kj}[x_{kj}]) \end{aligned} \tag{7}$$

Other physical and operational constraints fall outside the scope of this work.

IV. Computational method

The mission planning problem involves a high combinatorial search space. To solve this problem, we propose below a heuristic to reach an approximate solution in polynomial time based on the Simulated Annealing algorithm.

A. Fast computation of the objective function

First, we explain how to make most of the model to compute quickly the objective function. For our problem, we can attach a list of quality scores to each point of each request and maintain the $qualityCoverage$ score of the point. Each time we decide to plan an acquisition, we loop over the points covered by the strip, we add the quality score of the acquisition to their list and update their $qualityCoverage$ score if needed. The reverse process happens when we decide to remove an acquisition. (Algorithm 1)

The objective function can now be computed efficiently for a state X by looping over the states of the points as described in Algorithm 2, with a constant-time getQualityCoverage function.

B. Simulated Annealing

Simulated Annealing is a probabilistic metaheuristic motivated by the physical process of annealing in metallurgy. It involves heating a material to bring energy to the system and then cooling it down to a lower energy state. During the process, crystals are formed and their quality will depend on the thermodynamic properties of the material and the cooling rate. The Simulated Annealing heuristic (Algorithm 3) starts from a random state with high *temperature* and as the temperature is decreasing, it will probabilistically move to random neighboring states (little and localized changes in the decision vector) based on an *acceptance criterion*. The main idea is to allow deteriorating moves in the search space to escape from local optima, as long as the temperature (energy) is still high enough. While the energy is decreasing, such moves are less likely to be accepted and the heuristic will eventually favor improving states. The algorithm is stopped when the temperature has reached a lower bound or when the solution seems to be stabilized around a local optimum with no improvements over some iterations.

In this work, we use the Metropolis algorithm, inspired by the Boltzmann distribution, as the acceptanceCriterion: an improving move is always accepted and a deteriorating move is accepted with probability $\exp(-\Delta obj/T)$ where Δobj is the magnitude of the deterioration between the current state and the proposed state and T is the current temperature.

Algorithm 1 Time window insertion and removal

procedure INSERTTIMEWINDOW($s_{ki}[t]$)
  $x_{ki} \leftarrow t$
  for all request $r, Points_r(s_{ki}) \neq \emptyset$ **do**
    for all point $p \in Points_r(s_{ki})$ **do**
      ADDQUALITY(p, $quality(s_{ki}[t])$)
      UPDATEQUALITYCOVERAGE(p)
    end for
  end for
end procedure
procedure REMOVETIMEWINDOW($s_{ki}[t]$)
  $x_{ki} \leftarrow \emptyset$
  for all request $r, Points_r(s_{ki}) \neq \emptyset$ **do**
    for all point $p \in Points_r(s_{ki})$ **do**
      REMOVEQUALITY(p, $quality(s_{ki}[t])$)
      UPDATEQUALITYCOVERAGE(p)
    end for
  end for
end procedure

Algorithm 2 Computation of the objective function

procedure COMPUTEOBJECTIVE(X)
  $objective \leftarrow 0$
  for all request r **do**
    $score \leftarrow 0$
    for all point $p \in Points(r)$ **do**
      $score \leftarrow score$ + GETQUALITYCOVERAGE(p)
    end for
    $objective \leftarrow objective + w(r) \times score$
  end for
  return $objective$
end procedure

The temperature control will determine how much of the search space is explored and the quality of the solution. For the cooling scheme, we choose a geometric decrement $\alpha < 1$ such that $T \leftarrow \alpha T$ at each iteration. In order to choose the initial temperature, we compute an estimate for Δobj over successive moves by starting from random states. Then we reverse the acceptance criterion to get an initial probability of 0.70 to accept deteriorating moves.

1. Initialization

As our constraints are limited, we can start by selecting all the candidate strips of non-agile satellites with their single time window as there is no possible conflict between them. Then we randomly loop over the strips of agile sensors and try to plan each strip with a random time window while ensuring the feasibility (Algorithm 4). The CLUSTERFEASIBLE(C, $s_{ki}[t]$) function checks that the constraint is not violated if we plan $s_{ki}[t]$ belonging to cluster C.

2. Neighborhood structure

Metaheuristics such as Simulated Annealing rely on the generation of neighboring states with little and localized changes which are easily reversible. For instance, it can be inserting a new acquisition in the planning or a limited sequence of deletions of planned acquisitions followed by an insertion. We use different random operators and give a simple overview of such operations in Algorithm 5.

Finally, the COMEBACK function is just responsible to cancel the changes made during the neighbor's generation.

Algorithm 3 In-place Simulated Annealing

```
procedure SIMULATEDANNEALING
    T ← INITTEMPERATURE
    X ← INITRANDOMSTATE
    obj ← COMPUTEOBJECTIVE(X)
    while T ≥ T_final and not system stabilized do
        for i ← 1 to nbIter do
            GENERATENEIGHBORSTATE(X)
            newObj ← COMPUTEOBJECTIVE(X)
            if ACCEPTCRITERION(obj, newObj, T) then
                obj ← newObj
            else
                COMEBACK(X)
            end if
        end for
        T ← DECREASETEMPERATURE
    end while
    return X
end procedure
```

Algorithm 4 Random state initialization

```
procedure INITRANDOMSTATE
    for all sensor k ∈ K_non−agile do
        for all strip s_ki ∈ S(k) do
            INSERTTIMEWINDOW(s_ki[1])
        end for
    end for
    for all sensor k ∈ K_agile do
        for all cluster C ∈ C(k) do
            for all strip s_ki ∈ shuffle(C) do
                t ← index of random time window in TW(s_ki)
                if CLUSTERFEASIBLE(C, s_ki[t]) then
                    INSERTTIMEWINDOW(s_ki[t])
                end if
            end for
        end for
    end for
end procedure
```

C. Post-processing

Once the optimization is done, the solution may include some acquisitions which fully overlap with acquisitions of better quality. These acquisitions have no effect on the objective function (as the maximum quality is already reached for the points they cover) and were chosen randomly, the reason why we decide to remove such acquisitions.

The solution may also be a local optimum which can still be improved. Especially in terms of time windows where some residual pitch angles could be reduced or even set to zero. Therefore, we decide to run a second optimization pass, initialized with the solution given by the first pass. We consider that the strips selection problem has been solved and our only concern is now to affect better time windows for the selected strips to improve the total average quality, with the same constraint of intra-cluster feasibility. We solve this problem with a similar heuristic and as the combinatorial search space is reduced, the complexity decreased and this post-processing process is not expensive.

Algorithm 5 Random neighbor generation

```
procedure generateNeighborState(X)
    for all sensor k ∈ shuffle(K_agile) do
        iter ← 0
        while iter ≤ 15 do
            for all cluster C ∈ shuffle(C(k)) do
                for all strip s_ki ∈ shuffle(C) do
                    if x_ki = ∅ then
                        t ← index of random time window in TW(s_ki)
                        if clusterFeasible(C, s_ki[t]) then
                            insertTimeWindow(s_ki[t])
                            exit procedure
                        end if
                    end if
                end for
            end for
            deleted ← false
            while not deleted do
                C ← random cluster ∈ C(k)
                if there is planned strips in C then
                    s_ki ← random planned strip in C
                    removeTimeWindow(s_ki[x_ki])
                    deleted ← true
                end if
            end while
            iter = iter + 1
        end while
    end for
end procedure
```

V. Experimental results

The proposed model is evaluated on real requests issued by official agencies. The first scenario comes from the Department of Disaster Management of Bangladesh after flooding affected a rectangular 16000 km^2 area and 6 million people. The second request, defined by a 2800 km^2 circular area, was sent after flooding in several Vietnamese provinces. In both cases, quick disaster response is needed to find and give assistance to casualties and to assess damage to housing, infrastructures (roads, dams, bridges...) and agriculture.

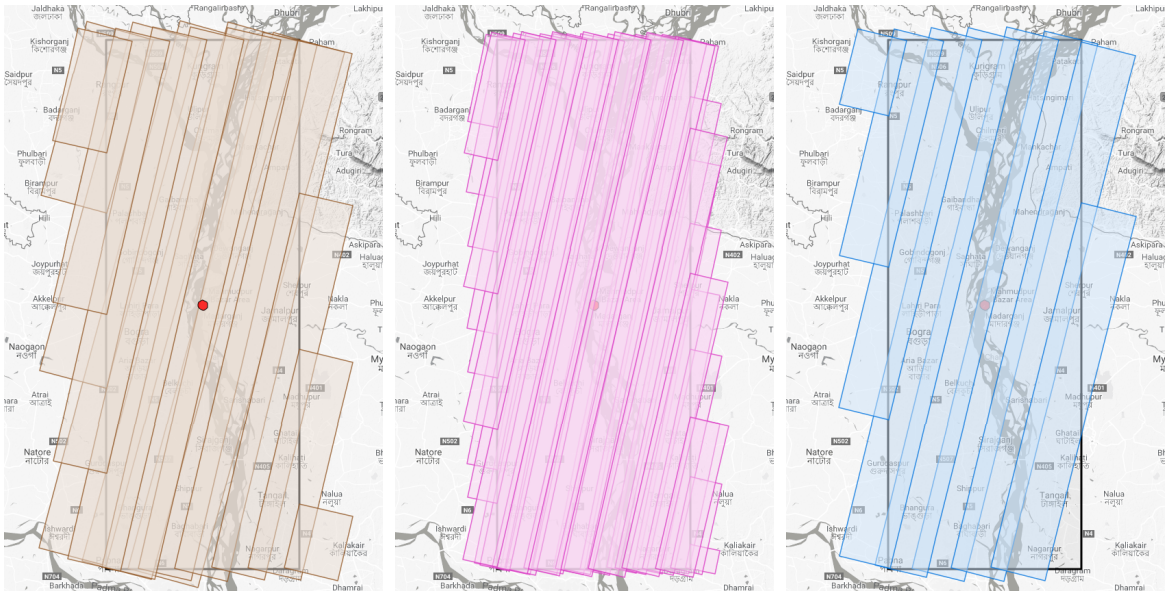
For the simulation, three high-resolution optical sensors are collaborating over a maximum timespan of 3 days. THEOS and DUBAISAT-2 are agile satellites and maximum acceptable roll and pitch angles are set to ±30° in the experiment. VNREDSAT-1 is a semi-agile satellite with a field of regard of ±30° roll angle.

Figure 6 shows the strips computed over the 3-day timespan for Bangladesh, with a ±1° step for the discretized roll angles. In this simple scenario, the number of clusters happens to be the number of orbit passes over the ROI. At that point, we know for sure that we can plan at least 2 non-conflicting acquisitions for THEOS, 3 for DUBAISAT-2 and 1 for VNREDSAT-1. Pitch agility and time windows, which are not depicted, can allow several observations belonging to the same cluster and thus greatly increase the combinatorial complexity of the mission planning problem.

Figure 7 illustrates the results for three different f function used to balance the coverage and the quality and Figure 8 gives more information about the scheduled acquisitions. When quality is not taken into account and every observation is worth the same with $f(q) = 1$, the constellation can cover 99.7% of the ROI in 2 days thanks to the pitch agility of THEOS (7 a and 8 a). When $f(q) = q$, the scores are proportional to the quality and it can be seen than the pitch agility is no more beneficial and the coverage is reduced to 94.8% (7 b and 8 b). Finally, we set $f(q) = 3q - 2$ so as to increase the weight of image quality and to introduce a soft constraint: indeed, only acquisitions with quality in [0.66, 1] will give a positive score and the bigger areas are now covered with a good quality (roll angles of 5°, -7°and 1°) to the cost of coverage, which drops to 62.3% (7 c and 8 c).

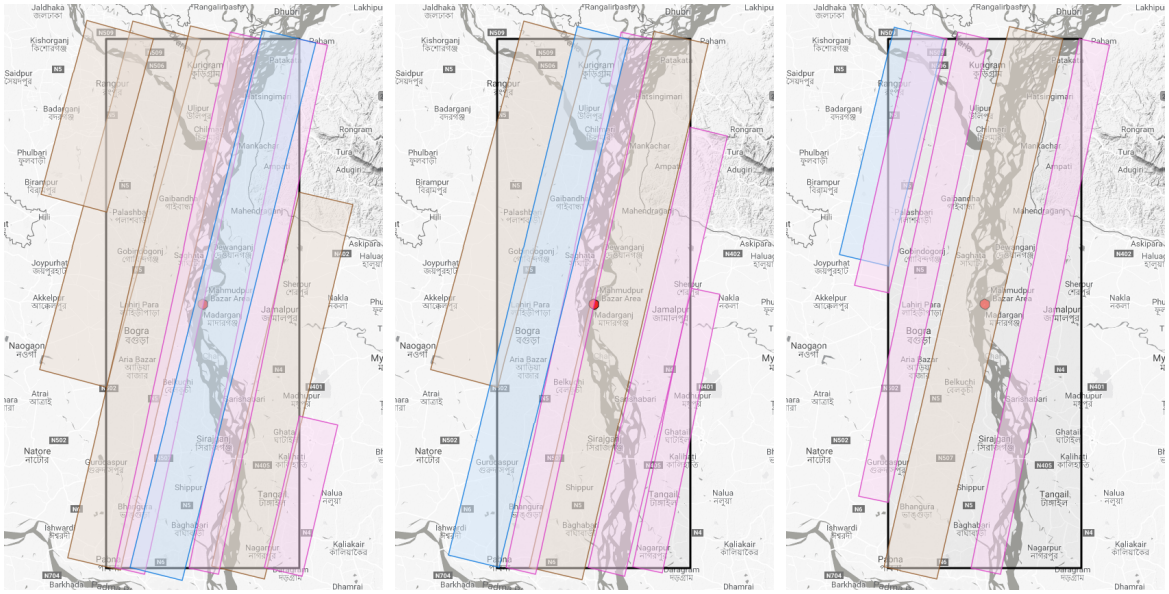
Figures 9 and 10 show similar results for the Vietnamese request.

The algorithms were implemented in Java and run on a 2 GHz CPU. From pre-processing to post-processing, the above instances were individually solved in about 30 seconds with strong convergence requirements on the metaheuristic.



(a) THEOS (15 strips, 2 clusters) (b) DUBAISAT-2 (34 strips, 3 clusters) (c) VNREDSAT-1 (8 strips, 1 cluster)

Fig. 6 Strips for 3 satellites, over 3 days (Bangladesh)



(a) $f(q) = 1$, 99.7% coverage (b) $f(q) = q$, 94.8% coverage (c) $f(q) = 3q - 2$, 62.3% coverage

Fig. 7 Selected acquisitions for 3 different f functions balancing coverage and quality (Bangladesh)

Sensor	Date Start ▲	Date End	Roll (°)	Pitch (°)	Area (km²)
DUBAISAT-2	Sat Sep 16 06:31:56 CEST 2017	Sat Sep 16 06:32:27 CEST 2017	5.0	0.0	2348
THEOS	Sun Sep 17 06:35:36 CEST 2017	Sun Sep 17 06:35:47 CEST 2017	-26.0	30.0	656
THEOS	Sun Sep 17 06:37:15 CEST 2017	Sun Sep 17 06:37:37 CEST 2017	-27.0	-12.0	2505
DUBAISAT-2	Sun Sep 17 06:38:54 CEST 2017	Sun Sep 17 06:39:26 CEST 2017	-9.0	0.0	2730
VNREDSAT 1	Sun Sep 17 07:03:00 CEST 2017	Sun Sep 17 07:03:33 CEST 2017	-28.0	0.0	4724
THEOS	Mon Sep 18 06:16:39 CEST 2017	Mon Sep 18 06:17:12 CEST 2017	2.0	26.0	4253
THEOS	Mon Sep 18 06:19:02 CEST 2017	Mon Sep 18 06:19:25 CEST 2017	-2.0	-30.0	2371
DUBAISAT-2	Mon Sep 18 06:46:13 CEST 2017	Mon Sep 18 06:46:22 CEST 2017	-29.0	0.0	443

(a) $f(q) = 1$, 99.7% coverage

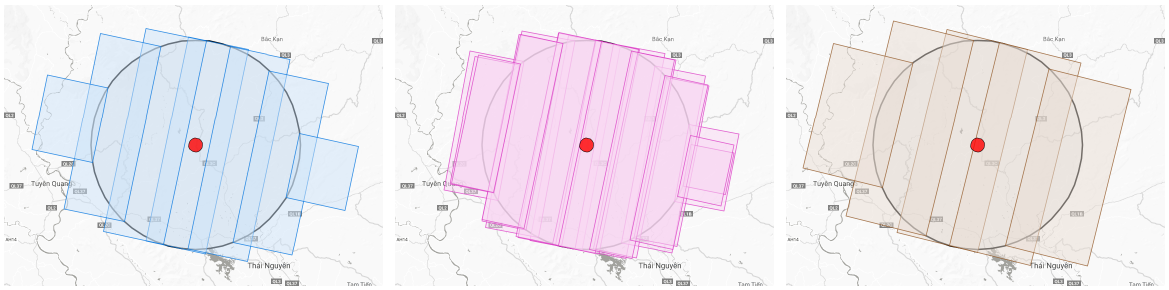
Sensor	Date Start ▲	Date End	Roll (°)	Pitch (°)	Area (km²)
DUBAISAT-2	Sat Sep 16 06:32:10 CEST 2017	Sat Sep 16 06:32:26 CEST 2017	3.0	0.0	1038
THEOS	Sun Sep 17 06:36:49 CEST 2017	Sun Sep 17 06:37:10 CEST 2017	-27.0	0.0	2505
DUBAISAT-2	Sun Sep 17 06:38:54 CEST 2017	Sun Sep 17 06:39:26 CEST 2017	-9.0	0.0	2730
VNREDSAT 1	Sun Sep 17 07:03:01 CEST 2017	Sun Sep 17 07:03:33 CEST 2017	-26.0	0.0	3737
THEOS	Mon Sep 18 06:17:40 CEST 2017	Mon Sep 18 06:18:13 CEST 2017	0.0	0.0	4741
DUBAISAT-2	Mon Sep 18 06:45:57 CEST 2017	Mon Sep 18 06:46:23 CEST 2017	-27.0	0.0	2274

(b) $f(q) = q$, 94.8% coverage

Sensor	Date Start ▲	Date End	Roll (°)	Pitch (°)	Area (km²)
DUBAISAT-2	Sat Sep 16 06:31:56 CEST 2017	Sat Sep 16 06:32:27 CEST 2017	5.0	0.0	2348
DUBAISAT-2	Sun Sep 17 06:38:55 CEST 2017	Sun Sep 17 06:39:22 CEST 2017	-7.0	0.0	1994
VNREDSAT 1	Sun Sep 17 07:03:02 CEST 2017	Sun Sep 17 07:03:16 CEST 2017	-24.0	0.0	1019
THEOS	Mon Sep 18 06:17:40 CEST 2017	Mon Sep 18 06:18:14 CEST 2017	1.0	0.0	4853
DUBAISAT-2	Mon Sep 18 06:45:54 CEST 2017	Mon Sep 18 06:46:09 CEST 2017	-21.0	0.0	1021

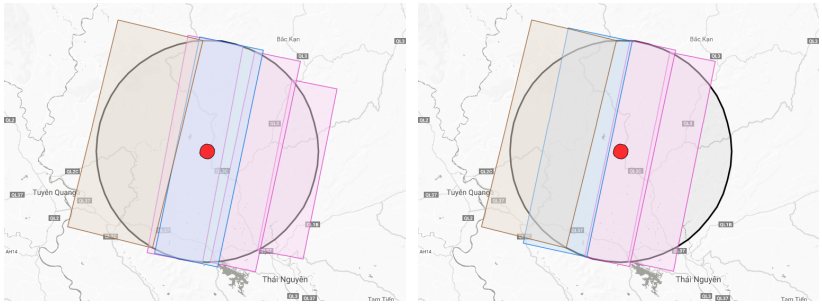
(c) $f(q) = 3q - 2$, 62.3% coverage

Fig. 8 Selected acquisitions for 3 different f functions balancing coverage and quality (Bangladesh)



(a) THEOS (5 strips, 1 cluster) (b) DUBAISAT-2 (20 strips, 3 clusters) (c) VNREDSAT-1 (7 strips, 1 cluster)

Fig. 9 Strips for 3 satellites, over 3 days (Vietnam)



(a) $f(q) = q$, 100% coverage (b) $f(q) = 4q - 3$, 89.6% coverage

Fig. 10 Selected acquisitions for 2 different f functions balancing coverage and quality (Vietnam)

VI. Conclusion

In this article, we proposed a mission planning methodology that allows the user to maximize the use of satellite resources for disaster response. The proposed approach optimizes the mission planning for a constellation of non-homogeneous satellites. First, a mathematical model was introduced, supporting multiple requests with different weights and both convex or non-convex ROIs. Then, a simplified strip-building algorithm was proposed. This pre-processing part must rely on a good knowledge of the physical characteristics of each collaborating satellite (e.g. orbital elements, camera characteristic and maneuvering capability for feasibility assessment). Finally, the mission planning problem was solved efficiently with the Simulated Annealing metaheuristic. Results from numerical experiments provide precious insight into possible cooperation in the very first days following a major event.

Future work should address common operational constraints (storage capacity, energy, downlink availability...) to ensure the feasibility of the mission and data delivering on time. The model could also be extended to other types of missions like post-disaster monitoring by seeking fairness in resource sharing.

VII. Acknowledgement

The authors would like to thank the OPTEMIS team for their contributions and GISTDA for sponsoring this project.

References

[1] Sun, K., Li, J., Chen, Y., and He, R., "Multi-objective Mission Planning Problem of Agile Earth Observing Satellites," *Proceedings of the 12th International Conference on Space Operations*, Vol. 4, 2012, pp. 2802–2810.

[2] Bianchessi, N., and Righini, G., "Planning and scheduling algorithms for the COSMO-SkyMed constellation," *Aerospace Science and Technology*, Vol. 12, No. 7, 2008, pp. 535–544.

[3] Bonnet, J., Gleizes, M.-P., Kaddoum, E., Rainjonneau, S., and Flandin, G., "Multi-satellite Mission Planning Using a Self-Adaptive Multi-agent System," *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*, 2015, pp. 11–20.

[4] Li, J., Yao, F., Bai, B., and He, R., "A Decomposition-Based Algorithm for Imaging Satellites Scheduling Problem," *2009 International Conference on Information Engineering and Computer Science*, 2009, pp. 1–6.

[5] Mitrovic-Minic, S., "Collection Planning Management: Multi-Satellite Collection Scheduling. Task 1 Report: Problem Statement and Literature Review," , 2016.

[6] Liu, S., and Hodgson, M. E., "Optimizing large area coverage from multiple satellite-sensors," *GIScience & Remote Sensing*, Vol. 50, No. 6, 2013, pp. 652–666.

[7] Liu, S., and Hodgson, M. E., "Satellite image collection modeling for large area hazard emergency response," *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 118, 2016, pp. 13–21.

[8] Niu, X., Zhai, X., Tang, H., and Wu, L., "Multi-satellite scheduling approach for dynamic areal tasks triggered by emergent disasters," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLI-B1, 2016, pp. 475–481.

[9] 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," *Computers & Industrial Engineering*, Vol. 61, No. 2, 2011, pp. 322–335.

[10] Hodgson, M. E., and Kar, B., "Modeling the Potential Swath Coverage of Nadir and Off-Nadir Pointable Remote Sensing Satellite-Sensor Systems," *Cartography and Geographic Information Science*, Vol. 35, No. 3, 2008, pp. 147–156.

[11] Li, X.-M., "Two-Archive2 Algorithm for Large-Scale Polygon Targets Observation Scheduling Problem," *2017 2nd International Conference on Information Technology and Management Engineering*, 2017, pp. 1–6.

[12] Rabideau, G., Chien, S., Mclaren, D., Knight, R., Anwar, S., Mehall, G., and Christensen, P., "Scheduling Results for the THEMIS Observation Scheduling Tool," *Proceedings International Workshop on Planning and Scheduling for Space*, 2011.

[13] Liu, X., Laporte, G., Chen, Y., and He, R., "An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time," *Computers & Operations Research*, Vol. 86, 2017, pp. 41–53.

[14] Chaimatanan, S., and Vongsantivanich, W., "The Development of Mission Planning Tool for Thailand's Earth Observation Mission," *SpaceOps 2016 Conference*, 2016.

[15] Kaku, K., and Held, A., "Sentinel Asia: A space-based disaster management support system in the Asia-Pacific region," *International Journal of Disaster Risk Reduction*, Vol. 6, 2013, pp. 1 – 17.

[16] Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., and Bataille, N., "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, Vol. 6, No. 5, 2002, pp. 367–381.