

Robust Earth observation satellite scheduling with uncertainty of cloud coverage

Xinwei Wang^{a,b}, Guopeng Song^b, Roel Leus^b, Chao Han^{a,*}

^a*School of Astronautics, Beihang University, Beijing, China*

^b*ORSTAT, KU Leuven, Leuven, Belgium*

Abstract

Earth observation satellites (EOSs) are specially equipped with remote sensing instruments to acquire images. In practical EOS scheduling, the uncertainty of cloud coverage is inevitable. We are the first to address robust EOS scheduling under uncertainty due to cloud coverage where the objective function aims to maximize the entire observation profit. We provide a robust formulation of the scheduling problem on the basis of a budgeted uncertainty set, while preserving the formulation linearity. A column-generation-based heuristic is also developed, in which the scheduling decisions in each satellite orbit are represented as columns. Eventually, a high-quality feasible solution is obtained using the generated columns. Extensive simulations are conducted on the basis of one of China's EOS constellations. The results indicate that the average optimality gap is less than 5%, which validates the performance of the proposed heuristic.

Keywords: Earth observation satellites, robust scheduling, cloud coverage, budgeted

*Corresponding author

Email address: hanchao@buaa.edu.cn (Chao Han)

¹This work was supported by the China Scholarship Council and the Academic Excellence Foundation of BUAA for PhD students.

1. Introduction

Earth observation satellites (EOSs) utilize remote sensing instruments to acquire images of the Earth surface, performing observations with large field of view according to users requirements. The broad range of applications of EOSs includes the fields of Earth resource exploration, global climate monitoring and disaster surveillance. With the number of orbiting EOSs and observation demands continuously growing [1], scheduling the observation targets for a set of EOSs has also become of the utmost importance. In this process of EOS scheduling, each satellite is designated to execute specific observation missions; in this work, an observation mission refers to the observation of one specific ground target by one given satellite within a specific observation time window (*OTW*).

As seen in Figure 1, a typical EOS has limited attitude maneuvering capacity along the roll axis, leading to a fixed *OTW* for targets in a given satellite orbit. The EOS can only start its observation at a pre-determined time t_a and end the observation at time t_b .

EOS scheduling has been well studied recently. Lin *et al.* [2] have developed an integer programming model to solve the imaging order scheduling problem for the EOS ROCSAT-II. Gabrel and Vanderpooten [3] formulate an EOS scheduling problem as the selection of a multiple-criteria path in a graph without cycles (a cycle is a path in which the first and last vertex are the same). Two stages are designed to generate efficient paths and select satisfactory paths using a multiple-criteria interactive procedure. Vasquez and Hao [4]

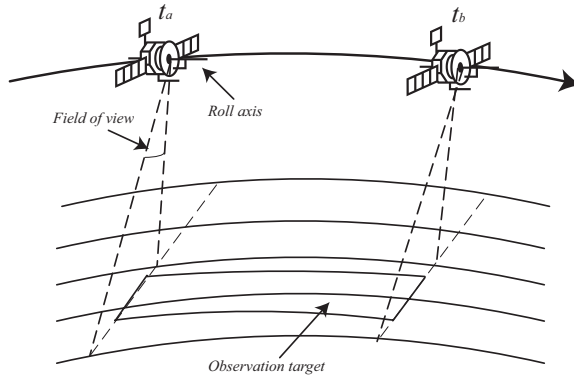


Figure 1: Illustration of the fixed observation interval of an EOS.

consider a large number of logical constraints and present a knapsack model [5]. Perea *et al.* [6] propose an EOS swath acquisition scheduling problem, which is modeled with a set covering formulation. A greedy randomized adaptive search procedure (GRASP) [7] is then introduced to find heuristic solutions. Considering the drift-angle constraint, Du *et al.* [8] provide an area target observation scheduling formulation and propose an ant colony algorithm to solve the problem. Ntagiou *et al.* [9] design an automated mission scheduling model for an Earth imaging mission and a data relay mission. Ant colony optimization is then applied to address these problems. Several constraint-satisfaction-based models [10, 11, 12] have also been proposed. In order to address the scheduling problem described with the use of these models, genetic algorithms are widely utilized [12, 13, 14, 15, 16], and other heuristic techniques have been developed [4, 12, 17]. Similarly, other satellite scheduling problems that aim to maximize the data transition throughput (downlink or uplink) have also been addressed [18, 19].

In practical EOS scheduling, the uncertainty of cloud coverage is inevitable. Ju and Roy [20] report that the pictures obtained by the Landsat-7 sensor were covered by clouds for around 35% on average, demonstrating that cloud coverage seems to be inevitable. Algra [21] proposes cloud-cover avoidance methods to improve system efficiency, while maintaining effective revisit time performance. Liao and Yang [22] consider current and future weather conditions to provide qualified pictures satisfying customer requirements, and establish a stochastic integer programming model. Lin *et al.* [2], Lin and Liao [23], and Lin and Chang [24] use cloud-coverage prediction data sets from weather forecasts and acquire substantially cloud-free images. Wang *et al.* [25] formulate cloud coverage as a stochastic event and propose an integer linear programming model based on a sample approximation method. Pang *et al.* [26] design an energy-efficient scheduling method considering stochastic failures where scenario optimization is utilized. Zhai *et al.* [27] consider that the observation missions may not be completed due to changing weather conditions, and propose a mission rearrangement method.

It is worth pointing out that all existing models considering cloud-coverage uncertainty have assumed that an observation mission is not awarded any profit once there are clouds in the picture. Nevertheless, pictures with partial cloud coverage can still be useful and entail a certain profit via cloud removal and image analysis in practice [28, 29, 30]. Consequently, the actual profit obtained from each observation mission will typically be a value strictly between zero and the profit without cloud coverage. To the best of our

knowledge, none of the existing models in this domain can readily be applied to optimize the actual profit retained in practice. In order to accurately describe the actual profit from each candidate observation mission, we will define a bounded profit range with a nominal and a deviation value, where the nominal value is the expected mission profit according to the mission's importance and the predicted cloud coverage, and the deviation value depends on the accuracy of the weather forecast. The actual observation profit is known when the actual weather condition is revealed.

There are two major paradigms for incorporating uncertainty into an optimization problem, namely stochastic programming and robust optimization. Stochastic programming involves a probability distribution for the uncertain parameters [31], while the accurate distributions are hard to obtain or even estimate in practice. Robust optimization, on the other hand, can still work with incomplete probabilistic information, and is typically more tractable from a practical point of view [32, 33]. Robust optimization has received significant attention ever since the work of Soyster [34], who applies a robust reformulation to find a solution that is feasible for all scenarios and has the best worst-case performance. Kouvelis and Yu [35] further extend and develop the robustness approach to decision making, assuming inadequate knowledge of the decision maker about the random state of nature. The aforementioned worst-case methods, however, tend to produce overly conservative solutions, and to accommodate this over-conservatism Bertsimas and Sim [36, 37] define a budgeted uncertainty set, where the level of conservatism can be

controlled while the robust formulation remains linear. With the advantages of flexibility and tractability, the concept of a budgeted uncertainty set has been widely applied in many application areas, such as power system operations [38], inventory theory [39], vehicle routing [40] and bin packing [41].

In this paper, we address a robust EOS scheduling problem with cloud-coverage uncertainty, subsequently referred to as “RSS.” We aim to maximize the entire observation profit subject to various practical constraints when the cloud coverage, which influences the actual observation profit, is uncertain. The robust formulation is described on the basis of a budgeted uncertainty set.

Large RSS instances typically require models with a great number of variables. We therefore resort to column generation (CG), which is a standard tool for solving a mathematical formulation with a high number of columns, by iteratively adding variables (columns) to a restricted model with a subset of the columns [42]. We decompose an initial compact formulation of RSS into different orbits, and represent the schedule in each satellite orbit as a column. Then a CG procedure is executed where the optimal solution of the restricted master problem (RMP) provides a linear programming (LP) bound for RSS. The RMP is iteratively solved by a commercial solver and the pricing problems are tackled with a dynamic programming (DP) method, until the optimal solution of RMP is found. Subsequently, a high-quality integer solution to RSS is obtained using all columns generated in the CG phase.

The main contributions of this paper are threefold. (1) We study an EOS scheduling problem under cloud-coverage uncertainty, with a bounded profit range to describe the actual profit in practice. (2) For the first time, a budgeted certainty set [37] is introduced to describe the uncertainty of cloud coverage in EOS scheduling, providing a trade-off between cost and robustness while preserving formulation linearity. (3) A compact robust formulation for RSS cannot be effectively handled by a generic commercial solver, and we therefore propose a CG-based heuristic, with a DP procedure for the pricing problems.

The remainder of this paper is structured as follows. A deterministic and a robust variant of RSS are defined in Section 2. The CG-based heuristic is sketched in Section 3. Section 4 contains a series of computational results which demonstrate the performance of the proposed model and method. We conclude our work in Section 5.

2. Problem statement

In this section, the deterministic EOS scheduling problem is first introduced, followed by a robust reformulation.

2.1. Deterministic EOS scheduling

The following assumptions will be maintained throughout this text regarding EOS scheduling:

1. A set of multiple EOSs executes observation missions for given targets within a given horizon.

2. Each satellite processes at most one observation mission at a given time, and observation preemption is not allowed.
3. Each target can be observed more than once to generate a higher profit, while there is an profit upper bound for each target.
4. The mission transformation time, energy consumption and memory capacity are considered as hard constraints.
5. The process of data transmission is not considered since we assume there are enough ground transmission stations and relay satellites to satisfy the data downlink requirements.

An overview of the most important symbols is as follows.

- T : the set of observation targets
- S : the set of satellites
- B_j : the set of orbits for satellite $j \in S$
- M_{jk} : the set of candidate observation missions for each orbit $k \in B_j$
- $(OTW_{jkp}, i, \rho_{jkp})$: a tuple denoting each observation mission p in orbit $k \in B_j$, where i represents the associated target and ρ_{jkp} is the received observation profit
- U_i : profit upper bound for each target $i \in T$
- x_{jkpq} : binary decision variable for observation missions p and q in orbit $k \in B_j$

- y_i : real variable representing the actual observation profit for each target $i \in T$

The value of ρ_{jkp} is typically determined by the observation target, the specific observation time window and the corresponding cloud coverage conditions. In practice, the range of ρ_{jkp} is given as input. Typically, multiple missions are associated with each target. Since we cannot receive unlimited profit for multiple observations for one target (see assumption 3 above), a profit upper bound U_i is defined for each target $i \in T$. For a first intuitive formulation of the deterministic variant of RSS, we define decision variable x_{jkpq} , which equals 1 when observation mission q is the immediate successor of mission p in orbit $k \in B_j$, and $x_{jkpq} = 0$ otherwise. Dummy missions s_{jk} and e_{jk} are added as source and sink nodes respectively in each satellite orbit. Moreover, variable y_i is introduced, with $0 \leq y_i \leq U_i$, to represent the actual observation profit for each target $i \in T$.

Transformation constraints between two missions are checked in a preprocessing step: the decision variable x_{jkpq} is defined only on condition that the sum of the completion time of mission p and the transformation time Δ_{jkpq}^T is not greater than the starting time of mission q in orbit $k \in B_j$. In fact, we only check transitions from one observation mission to other missions in the same orbit, since missions in different orbits always have enough transformation time. In the worst case when all targets are accessible by the satellite in one orbit, the computational complexity of the preprocessing is still $\mathcal{O}(n^2)$, with n the number of observation targets. The quantity Δ_{jkpq}^T consists of attitude maneuvering

time Δ_{jkpq}^V and attitude stabilization time Δ_{jkpq}^S , which are known values for the decision makers.

For modelling the memory constraints, we define M_j^C and M_j^I as, respectively, the satellite memory capacity in each orbit and the unit-time imaging memory occupation of satellite j . The total memory occupation of observation missions in each orbit cannot exceed the corresponding memory capacity. The maximal energy capacity of satellite j can be assumed to be constant [25] and is denoted as E_j^C in each orbit. Finally, we represent the unit-time imaging energy consumption and maneuvering energy consumption as E_j^I and E_j^M for satellite j , respectively. Our model for deterministic EOS scheduling is then:

$$\max \sum_{i \in T} y_i \quad (1)$$

subject to

$$\sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \rho_{jkp} \cdot x_{jkpq} \geq y_i \quad \forall i \in T \quad (2)$$

$$0 \leq y_i \leq U_i \quad \forall i \in T \quad (3)$$

$$\sum_{q \in M_{jk \cup e}} x_{jkpq} - \sum_{q \in M_{jk \cup s}} x_{jkqp} = \begin{cases} 1, & p = s_{jk} \\ 0, & \forall p \in M_{jk} \\ -1, & p = e_{jk} \end{cases} \quad \forall k \in B_j, j \in S \quad (4)$$

$$\sum_{p \in M_{jk}} \sum_{q \in M_{jk \cup e}} x_{jkpq} d_{jkp} M_j^I \leq M_j^C \quad \forall k \in B_j, j \in S \quad (5)$$

$$\sum_{p \in M_{jk}} \sum_{q \in M_{jk \cup e}} x_{jkpq} d_{jkp} E_j^I + \sum_{p \in M_{jk}} \sum_{q \in M_{jk}} x_{jkpq} \Delta_{jkpq}^V E_j^M \leq E_j^C \quad \forall k \in B_j, j \in S \quad (6)$$

$$x_{jkpq} \in \{0, 1\} \quad \forall q \in M_{jk \cup e}, p \in M_{jk \cup s}, k \in B_j, j \in S \quad (7)$$

where M_{jk}^i stands for the observation mission set associated with target i in orbit $k \in B_j$, $M_{jk \cup s} = M_{jk} \cup \{s_{jk}\}$, $M_{jk \cup e} = M_{jk} \cup \{e_{jk}\}$ and d_{jkp} is the observation duration of OTW_{jkp} . In this model, constraints (2) indicate that observation profit is obtained according to the scheduled observation missions for each target when the target profit is bounded by constraints (3). Constraint set (4) ensures flow generation and conservation. The memory and energy constraints are satisfied through (5) and (6), respectively. A variant of the foregoing model, which does not consider cloud coverage and with a profit function that is non-linear in the number of observations, was studied earlier in [43].

2.2. Robust reformulation

Uncertainty is inevitable in EOS scheduling, and cloud coverage is one of the main causes. Dependent on the mission importance and the weather conditions, the actual

profit of mission p in orbit $k \in B_j$ is bounded to an interval $[\bar{\rho}_{jkp} - \hat{\rho}_{jkp}, \bar{\rho}_{jkp} + \hat{\rho}_{jkp}]$, where $\bar{\rho}_{jkp}$ is the nominal profit and $\hat{\rho}_{jkp}$ is the deviation; $\bar{\rho}_{jkp}$ and $\hat{\rho}_{jkp}$ are non-negative, with $\bar{\rho}_{jkp} \geq \hat{\rho}_{jkp}$. The deviation value depends on the accuracy of the weather forecast. In line with the ideas in [37], an integer Γ_i is input, serving as a budget for the number of scheduled missions associated with target i that can deviate from their nominal profit, which controls the so-called ‘‘price’’ of robustness. The value of Γ_i should be set by the decision maker according to the operational cost of the satellites and the desired level of conservativeness.

A robust solution protects against the extreme situation where multiple missions with worst-case profit deviation are associated with the same target (namely, up to Γ_i for target i). The constraints (2) are then replaced by

$$\sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk} \cup e} \bar{\rho}_{jkp} x_{jkpq} - \max_{\bar{M} \subset M^i: |\bar{M}| \leq \Gamma_i} \sum_{p \in \bar{M}} \hat{\rho}_{jkp} x_{jkpq} \geq y_i, \quad \forall i \in T \quad (8)$$

where $M^i = \bigcup_{k \in B_j, j \in S} M_{jk}^i$, the set of candidate missions for target i across all orbits and satellites. For each target i , the second term in the left-hand side of equation (8), which represents the protection for a mission schedule \mathbf{x}^* , can be denoted as $\beta_i(\mathbf{x}^*, \Gamma)$. This value can be computed via the following auxiliary program:

$$\beta_i(\mathbf{x}^*, \Gamma) = \max \sum_{p \in M^i} \hat{\rho}_{jkp} x_{jkpq}^* z_{jkpq} \quad (9)$$

$$\text{s.t. } \sum_{p \in M^i} z_{jkpq} \leq \Gamma_i \quad (10)$$

$$0 \leq z_{jkpq} \leq 1 \quad \forall q \in M_{jk \cup e}, p \in M_{jk}^i, k \in B_j, j \in S \quad (11)$$

For a solution to $\beta_i(\mathbf{x}^*, \Gamma)$, we can simply sort all missions associated with target i in non-ascending order of profit deviation and pick at most Γ_i observation missions from the beginning of the list.

We associate the dual variables θ_i^1 with constraints (10) and dual variables θ_{jkpq}^2 with constraints (11). Then the dual formulation of model (9)–(11) is (for a given $i \in T$):

$$\min \quad \Gamma_i \theta_i^1 + \sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \theta_{jkpq}^2 \quad (12)$$

$$\text{s.t. } \theta_i^1 \geq 0 \quad (13)$$

$$\theta_i^1 + \theta_{jkpq}^2 \geq \hat{\rho}_{jkp} x_{jkpq}^* \quad \forall q \in M_{jk \cup e}, p \in M_{jk}^i, k \in B_j, j \in S \quad (14)$$

$$\theta_{jkpq}^2 \geq 0 \quad \forall q \in M_{jk \cup e}, p \in M_{jk}^i, k \in B_j, j \in S \quad (15)$$

By strong duality, the optimal value of (12)–(15) equals $\beta_i(\mathbf{x}^*, \Gamma)$. Substituting this formulation into (8), the problem RSS can be captured as the maximization of (1) subject to constraints (3)–(7), (13), (15) and

$$\sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \bar{\rho}_{jkp} x_{jkpq} \geq y_i + \Gamma_i \theta_i^1 + \sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \theta_{jkpq}^2 \quad \forall i \in T \quad (16)$$

$$\theta_i^1 + \theta_{jkpq}^2 \geq \hat{\rho}_{jkpq} x_{jkpq} \quad \forall q \in M_{jkUe}, p \in M_{jk}^i, k \in B_j, j \in S, i \in T \quad (17)$$

Although a large number of dual constraints are introduced into the robust model, the robust reformulation remains linear.

2.3. Illustration

To further distinguish the robust model from the deterministic one, an illustrative instance is presented in this section. We consider one satellite only, for which the parameters for five potential non-dummy observation missions in three different satellite orbits are presented in Table 1, where the first column is the mission index, and columns $\bar{\rho}$ and $\hat{\rho}$ are the nominal profit and the profit deviation, respectively. The remaining columns contain the observation time window, the target and the orbit per mission. The profit upper bound U_i and uncertainty budget Γ_i are set as 12 and 1, respectively, for all targets i . Memory and energy constraints are not considered here for simplicity. Notice that observation missions 1 and 2 overlap and thus cannot be scheduled together; the same goes for missions 3 and 4.

Table 1: An illustrative instance.

mission	$\bar{\rho}$	$\hat{\rho}$	<i>OTW</i>	target	orbit
1	9	3	[20, 30]	1	1
2	6	1	[25, 35]	2	1
3	8	3	[200, 210]	2	2
4	7	1	[205, 215]	1	2
5	5	1	[300, 310]	3	3

The deterministic model, in which each $\rho_{jkp} = \bar{\rho}_{jkp}$, has an optimal solution that executes missions 1, 3 and 5, achieving a maximal profit of $9 + 8 + 5 = 22$.

In the robust formulation with uncertainty budget, the same schedule receives a profit of $(9 - 3) + (8 - 3) + (5 - 1) = 15$. An optimal robust schedule, however, executes missions 1, 4 and 5, with profit $(9 - 3) + (7 - 1) + (5 - 1) = 16$. This example clearly shows that the incorporation of uncertainty has a strong influence on the scheduling results.

3. Column-generation heuristic

CG, which was first proposed in [44, 45] for the cutting-stock problem, has been used for decades in various fields [42, 46, 47, 48]. The main advantage of CG is that not all possible variables need to be enumerated. Below we describe a decomposition of the compact formulation for RSS proposed in Section 2.2, leading to a new model that typically has a better LP bound but with a large number of variables, which will be the basis for the CG heuristic that is developed.

3.1. Formulation decomposition and pricing problems

The original model for RSS is decomposed by enumerating all possible solutions per satellite orbit. The schedules x_{jkpq} in each orbit k are regarded as columns. Denote the set of schedules for satellite orbit $k \in B_j$ as R_{jk} . Each schedule $m \in R_{jk}$ contains values x_{jkpq}^m , where $x_{jkpq}^m = 1$ if the mission p is the immediate predecessor of mission q , and $x_{jkpq}^m = 0$ otherwise. A binary variable z_{jkm} is introduced so that $z_{jkm} = 1$ when schedule

$m \in R_{jk}$ is chosen and 0 otherwise. The resulting master problem is then to maximize (1)

subject to (3), (13), (15) and

$$\sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \sum_{m \in R_{jk}} \bar{\rho}_{jkp} x_{jkpq}^m z_{jkm} \geq y_i + \Gamma_i \theta_i^1 + \sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \theta_{jkpq}^2 \quad \forall i \in T \quad (18)$$

$$\theta_i^1 + \theta_{jkpq}^2 \geq \sum_{m \in R_{jk}} \hat{\rho}_{jkp} x_{jkpq}^m z_{jkm} \quad \forall q \in M_{jk \cup e}, p \in M_{jk}^i, k \in B_j, j \in S, i \in T \quad (19)$$

$$\sum_{m \in R_{jk}} z_{jkm} = 1 \quad \forall k \in B_j, j \in S \quad (20)$$

$$z_{jkm} \in \{0, 1\} \quad \forall m \in R_{jk}, k \in B_j, j \in S \quad (21)$$

Constraints (18) and (19) correspond with (16) and (17). Constraints (20) state that only one scheduling solution is selected per orbit.

The integrality of the master problem is removed for the LP relaxation. With dual variables π_i^1 with constraints (3), variables π_i^2 with constraints (18), variables π_{jkpq}^3 with constraints (19), and variables π_{jk}^4 with constraints (20), we obtain the following dual:

$$\min \sum_{i \in T} \pi_i^1 U_i + \sum_{j \in S} \sum_{k \in B_j} \pi_{jk}^4 \quad (22)$$

subject to

$$\pi_i^1 - \pi_i^2 \geq 1 \quad \forall i \in T \quad (23)$$

$$-\Gamma_i \pi_i^2 + \sum_{j \in S} \sum_{k \in B_j} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} \pi_{jkpq}^3 \geq 0 \quad \forall i \in T \quad (24)$$

$$-\pi_i^2 + \pi_{jkpq}^3 \geq 0 \quad \forall q \in M_{jk \cup e}, p \in M_{jk}^i, k \in B_j, j \in S, i \in T \quad (25)$$

$$\sum_{i \in T} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} (\bar{\rho}_{jkp} \pi_i^2 - \hat{\rho}_{jkp} \pi_{jkpq}^3) x_{jkpq}^m + \pi_{jk}^4 \geq 0 \quad \forall m \in R_{jk}, k \in B_j, j \in S \quad (26)$$

$$\pi_i^1 \geq 0, \pi_i^2 \leq 0 \quad \forall i \in T \quad (27)$$

$$\pi_{jkpq}^3 \leq 0 \quad \forall q \in M_{jk \cup e}, p \in M_{jk}^i, k \in B_j, j \in S, i \in T \quad (28)$$

$$\pi_{jk}^4 \in \mathbb{R} \quad \forall k \in B_j, j \in S \quad (29)$$

In the RMP, not all columns in R_{jk} are included, but for LP optimality we should check all the constraints (26) at each CG iteration. If a non-included column with a violated constraint exists, it should be added to the model and otherwise the solution to the RMP is optimal also for the full master with all columns. The LP relaxation is typically solved faster by adding strongly violated constraints, and this equates with the search for columns in the primal master with most negative reduced cost. This search is referred to as a pricing problem, separately for each satellite orbit $k \in B_j$, as follows:

$$\min \sum_{i \in T} \sum_{p \in M_{jk}^i} \sum_{q \in M_{jk \cup e}} (\bar{\rho}_{jkp} \pi_i^2 - \hat{\rho}_{jkp} \pi_{jkpq}^3) x_{jkpq} + \pi_{jk}^4 \quad (30)$$

$$\text{s.t.} \quad \sum_{q \in M_{jk \cup e}} x_{jkpq} - \sum_{q \in M_{jk \cup s}} x_{jkqp} = \begin{cases} 1, & p = s_{jk} \\ 0, & \forall p \in M_{jk} \\ -1, & p = e_{jk} \end{cases} \quad (31)$$

$$\sum_{p \in M_{jk}} \sum_{q \in M_{jk \cup e}} x_{jkpq} d_{jkp} M_j^I \leq M_j^C \quad (32)$$

$$\sum_{p \in M_{jk}} \sum_{q \in M_{jk \cup e}} x_{jkpq} d_{jkp} E_j^I + \sum_{p \in M_{jk}} \sum_{q \in M_{jk}} x_{jkpq} \Delta_{jkpq}^V E_j^M \leq E_j^C \quad (33)$$

For each pricing problem (each satellite orbit), if the optimal value of the pricing problem is less than 0, then a new column is generated with lowest reduced cost in the corresponding orbit; otherwise no column is appended. The iterative CG procedure halts when the pricing problem for each satellite orbit obtains an optimal value not less than 0, indicating that constraints (26) are satisfied. In our computational experiments described in Section 4, the number of CG iterations typically depends on the size of instance, mainly on the number of observation targets. The corresponding LP objective value constitutes a tight upper bound for problem RSS.

3.2. Column initialization

We introduce a dominance relation among the columns to improve the initialization process. For each satellite orbit, if the worst-case profit of the scheduled observations for any target in column m_1 is no less than the nominal profit of the corresponding target in column m_2 , and for at least one target, the worst-case profit in column m_1 is strictly

greater than that of column m_2 , then column m_2 can be discarded since column m_1 offers more contribution to the objective function even in the worst situation.

The random column initialization process proceeds as follows. The number of attempts generating initial columns in each satellite orbit is given as input. For each generation iteration in orbit $k \in B_j$, we randomly re-order the observation missions and check whether they can be greedily added into the current schedule $CurR$. After the generation of the $CurR$, we then check whether it is dominated by the existing columns in R_{jk} . $CurR$ is discarded if it is dominated, otherwise $CurR$ is added into the column pool R_{jk} . Similarly, if a column currently in the column set is dominated by $CurR$, then we remove the dominated column.

3.3. Dynamic programming for pricing

The pricing problems in each orbit need to be solved iteratively in the CG process. Following Gabrel and Vanderpooten [3], a DP method with partial path dominance criteria is designed for solving the pricing problems efficiently.

For each pricing problem in satellite orbit $k \in B_j$, a directed acyclic graph $G_{jk} = (N_{jk}, E_{jk})$ is defined initially, where the vertex set N_{jk} contains all candidate observation missions and dummy missions s_{jk} and e_{jk} , and an edge is present in the edge set E_{jk} if the mission corresponding with the destination node can be executed immediately after the origin node. We also include edges from s_{jk} to each other vertex, and e_{jk} has edges incoming from all other vertices. Then we define partial paths in graph G_{jk} as follows.

Definition 1. *Partial Path.* For the pricing problem in satellite orbit $k \in B_j$, a partial path P_{jkpt} is a tuple $(PCost_{jkpt}, PM_{jkpt}, PE_{jkpt})$, representing a path that connects the dummy source mission to an actual observation mission or dummy sink mission p , with t an index of the different paths ending at mission p , $PCost_{jkpt}$ representing the sum of the mission cost along the path, and PM_{jkpt} and PE_{jkpt} denoting the consumption of mission memory occupation and energy along the path, respectively.

Note that the partial path can contain several missions, as long as all constraints are satisfied. The mission cost $MCost_{jkp}$ of mission p in orbit $k \in B_j$ equals $\bar{\rho}_{jkp}\pi_i^2 - \hat{\rho}_{jkp}\pi_{jkp}^3$, where i is the associated mission target and q is the immediate predecessor mission (see objective function (30)). For the dummy source and sink missions, the mission cost is chosen as $\pi_{jk}^4/2$, and the mission memory and energy consumption are set to 0. The search for a column with the most negative reduced cost now transforms to the search of a partial path with minimal mission cost from source to sink.

We denote the set of partial paths ending at mission p as \mathcal{P}_{jkp} , and further introduce the concept of partial path dominance to enhance the algorithmic efficiency of the DP method.

Definition 2. *Path dominance.* We say that in the set of partial paths \mathcal{P}_{jkp} , partial path P_{jkpt_1} dominates P_{jkpt_2} when the following conditions hold, and at least one of the inequalities is strict: 1) $PCost_{jkpt_1} \leq PCost_{jkpt_2}$; 2) $PM_{jkpt_1} \leq PM_{jkpt_2}$; 3) $PE_{jkpt_1} \leq PE_{jkpt_2}$.

For the pricing problem in a given satellite orbit, a dominated partial path can be discarded without losing all optimal solutions. The DP method with partial path dominance is described in Algorithm 1. The observation missions in orbit $k \in B_j$ are ordered in increasing observation starting time so that each edge in G_{jk} leads to a mission with a strictly larger starting time than its origin node. For each mission p in orbit $k \in B_j$, we denote the set of candidate predecessor missions as M_{jk}^{-p} , and we invoke function *GetCurPreMisIndex()* to obtain the mission index of the predecessor mission. We then extend the path in \mathcal{P}_{jkr} ending at mission r by appending the current cost $MCost_{jkp}$, memory consumption MM_{jkp} and energy ME_{jkp} . Function *CheckConstraints()* returns *true* when the extended path satisfies all constraints and returns *false* otherwise. Function *CheckPathDominance()* returns *true* if *NewP* is not dominated by paths in \mathcal{P}_{jkp} and *false* otherwise. The extended path can be added into \mathcal{P}_{jkp} after the dominance check. We also check whether *NewP* dominates the paths in \mathcal{P}_{jkp} . Eventually, we obtain a path with minimal cost in \mathcal{P}_{jke} , which coincides with a minimum for objective function (30).

3.4. Framework of the CG heuristic

The framework of the CG-based heuristic is sketched in Algorithm 2. After initialization of the column pool, the RMP is iteratively solved by a commercial solver while the pricing problems are tackled with the DP method. Function *CheckOptFound()* returns *true* when there is no new column generated for any of the satellite orbits; otherwise it

Algorithm 1 DP for pricing

```
1: Order missions  $M_{jk}$  in ascending observation starting time;
2: for each mission  $p \in M_{jk} \cup \{e_{jk}\}$  do
3:    $P_{jkp} = \emptyset$ ;
4:   for each predecessor mission in  $M_{jk}^{-p}$  do
5:      $r = \text{GetCurPreMisIndex}()$ ;
6:     for each  $P_{jkrt'} \in \mathcal{P}_{jkr}$  do
7:        $\text{NewP} = (\text{PCost}_{jkrt'} + \text{MCost}_{jkp}, \text{PM}_{jkrt'} + \text{MM}_{jkp}, \text{PE}_{jkrt'} + \text{ME}_{jkp})$ ;
8:       if  $\text{CheckConstraints}(\text{NewP})$  then
9:         if  $\text{CheckPathDominance}(\mathcal{P}_{jkq}, \text{NewP})$  then
10:           $\mathcal{P}_{jkp} = \mathcal{P}_{jkp} \cup \{\text{NewP}\}$ ;
11:        end if
12:      end if
13:    end for
14:  end for
15: end for
16: Output a minimal-cost path in  $\mathcal{P}_{jke}$ 
```

returns *false*. Function $\text{CheckViolationExist}()$ returns *true* when the optimal solution of the pricing problem violates the dual constraint, and *false* otherwise. At the end, a feasible schedule is generated for RSS by solving the RMP with all the generated columns as an integer program.

Algorithm 2 CG heuristic

```
1: Column initialization;
2: while  $\text{CheckOptFound}()$  do
3:   Solve RMP and update dual parameters;
4:   for each  $j \in S$  do
5:     for each  $k \in B_j$  do
6:       Solve the pricing problem via DP;
7:       if  $\text{CheckViolationExist}()$  then
8:         Generate new column for current satellite orbit;
9:       end if
10:    end for
11:  end for
12: end while
13: Solve the integer master problem of RSS with all the existing columns
```

Table 2: Orbital parameters of the satellite constellation *Gaojing-1*.

<i>ID</i>	<i>a</i> (km)	<i>i</i> (°)	Ω (°)	<i>e</i>	ω (°)	<i>M</i> (°)
<i>Sat1</i>	6903.673	97.5839	97.8446	0.0016546	50.5083	2.0288
<i>Sat2</i>	6903.730	97.5310	95.1761	0.0015583	52.2620	31.4501
<i>Sat3</i>	6909.065	97.5840	93.1999	0.0009966	254.4613	155.2256
<i>Sat4</i>	6898.602	97.5825	92.3563	0.0014595	276.7332	140.1878

4. Computational experiments

4.1. Data generation

One of China’s high resolution EOS constellations, namely *Gaojing-1* (*SuperView-1*) [49], is used as the basis for the data generation. *Gaojing-1* is a Chinese commercial constellation of four remote sensing satellites. In Table 2, the first column *ID* is the name of the satellite, and the other columns contain the parameter values for the satellite semi-major axis *a*, inclination *i*, right ascension of the ascending node Ω , eccentricity *e*, argument of perigee ω and mean anomaly *M*. In addition, the satellite platform of the constellation allows up to 30° maneuvers along the roll axis.

Since there is no common benchmark for EOS scheduling with cloud coverage uncertainty, the experiments are conducted on a diverse set of instances. Following [50, 51], we generate instances in which the observation targets are randomly distributed worldwide, augmented with concentrations in specific interest areas. The total number of observation targets is 150, 200, 250 or 300, with a worldwide distribution of 150 targets, and additionally zero, one, two or three interest areas with 50 targets each are generated in each instance. We assume that each target is relatively small and that the satellite can

entirely observe the target with a large-scale field of view for each observation mission. The mission horizon is set as 72 hours (three days).

For each target $i \in T$, we define the target profit ρ_i , which is uniformly distributed from 1 to 10, as the maximal observation profit that each mission associated with target i can obtain. The upper profit U_i of target i is uniformly distributed from 1 to 2 times ρ_i . The nominal profit of target i varies from 0 to ρ_i , indicating that the mission profit depends on mission target's importance as well as on predicted cloud-coverage conditions. The profit deviation is uniformly distributed from 0.1 to 0.5 times the corresponding nominal mission profit. In order to test the influence of various degrees of cloud-coverage uncertainty, the budget value Γ_i is set as 1, 2 or 3, and takes the same value for all targets $i \in T$ in the instance.

Since the satellites in the *Gaojing-1* constellation have very similar properties, we take their parameters as identical. The unit-time imaging memory occupation M^I is 10 MB/s, while the unit-time imaging consumption E^I and maneuvering energy consumption E^M are 500 and 1000 W , respectively. The memory capacity M^C is set as 400, 500 or 600 MB. The energy capacity E^C is either 40, 50 or 60 kJ. For each combination of values for the parameters, 10 instances are randomly generated.

4.2. Computational results

4.2.1. Setup

The computational experiments are conducted on a laptop with Intel Core i5-7200 CPU at 2.5 GHz and 8 GB of RAM on a Windows 10 64-bit OS. Our algorithms are implemented in Visual C++. All LPs and integer programs are solved by CPLEX 12.6.3 using Concert Technology with four threads on two cores. The optimality gap tolerance of the integer solver and the time limit for each run are set as 1% and 1200 seconds, respectively.

In the following tables, the columns labelled *opt* report the number of instances whose integer master problem is solved within the time limit, out of 10 instances per setting. The entries *gap* represent the relative gap between the integer solution obtained from the CG heuristic and the LP bound. Columns *time* and *mas* report the total average runtime, and the runtime spent on the integer master problem, respectively (both in seconds).

4.2.2. Results and discussion

We first report the CPU time of the original compact robust formulation of Section 2.2 for instances with 150 and 200 targets in Table 3, where the columns gap_c contain the relative solution gap (expressed as a percentage) reported by CPLEX. The solver efficiently obtains optimal solutions for instances with 150 targets, although there is a slightly increasing trend in the runtime as the budgeted value increases. For all instances with 200

targets, however, the original formulation struggles due to its large size and does not run to optimality within the time limit; the solution gaps are significant, and are increasing with Γ_i . Due to this poor computational performance, we will not include the results of the original formulation for the larger instances.

Table 3: Results of the compact robust formulation for instances with 150 and 200 targets.

		150 targets			200 targets					
		$\Gamma_i = 1$	$\Gamma_i = 2$	$\Gamma_i = 3$	$\Gamma_i = 1$		$\Gamma_i = 2$		$\Gamma_i = 3$	
M^C	E^C	<i>time</i>	<i>time</i>	<i>time</i>	<i>time</i>	<i>gap_c</i>	<i>time</i>	<i>gap_c</i>	<i>time</i>	<i>gap_c</i>
400	40	1.91	5.73	6.30	1200.00	11.4	1200.00	18.9	1200.00	22.3
	50	1.23	3.38	4.88	1200.00	8.8	1200.00	13.4	1200.00	16.1
	60	0.99	2.84	3.89	1200.00	6.1	1200.00	10.1	1200.00	12.8
500	40	2.42	6.32	6.40	1200.00	12.6	1200.00	24.2	1200.00	25.4
	50	1.22	3.84	10.09	1200.00	7.3	1200.00	19.7	1200.00	28.2
	60	0.77	2.48	8.56	1200.00	6.8	1200.00	12.6	1200.00	20.9
600	40	2.05	4.17	11.67	1200.00	10.2	1200.00	16.6	1200.00	26.6
	50	1.27	3.85	12.99	1200.00	7.7	1200.00	13.3	1200.00	31.7
	60	0.72	1.72	6.55	1200.00	5.6	1200.00	9.5	1200.00	17.8
Overall		1.40	3.81	8.10	1200.00	8.9	1200.00	15.4	1200.00	22.4

The CG-based heuristic, referred as CGH below, is tested on instances with 150, 200, 250 and 300 targets, and the computational results are summarized in Tables 4 to 7.

Overall, CGH performs consistently well, obtaining high-quality solutions with less than 4% average optimality gap. CGH produces integer solutions within limited time for all instances with 150, 200 and 250 targets. For 46 out of the 270 instances with 300 targets, CGH is unable to complete the integer master problem within the imposed limit of 1200 seconds, since CPLEX struggles with the branch-and-bound procedure for large instances (the CG phase for the LP relaxation is very fast). The average gap between

Table 4: Results for CGH on instances with 150 targets.

M^C	E^C	$\Gamma_i = 1$				$\Gamma_i = 2$				$\Gamma_i = 3$			
		<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>
400	40	10	2.18	4.24	0.46	10	2.60	4.05	1.01	10	2.47	4.76	1.27
	50	10	2.06	4.52	0.44	10	2.48	4.21	0.89	10	2.37	4.56	1.13
	60	10	2.02	4.51	0.37	10	2.44	4.42	0.87	10	2.38	4.62	1.12
500	40	10	2.13	3.73	0.48	10	2.58	4.09	0.94	10	2.44	4.75	1.06
	50	10	1.93	4.51	0.33	10	2.35	4.44	0.83	10	2.34	4.49	0.99
	60	10	1.87	3.97	0.24	10	2.29	3.99	0.59	10	2.22	4.15	0.91
600	40	10	2.11	3.46	0.44	10	2.58	3.89	0.97	10	2.43	4.07	0.99
	50	10	1.98	4.18	0.32	10	2.31	4.49	0.71	10	2.35	3.85	0.93
	60	10	1.85	4.03	0.22	10	2.17	3.93	0.49	10	2.16	3.94	0.80
Overall		90	2.01	4.13	0.37	90	2.42	4.17	0.81	90	2.35	4.35	1.02

Table 5: Results for CGH on instances with 200 targets.

M^C	E^C	$\Gamma_i = 1$				$\Gamma_i = 2$				$\Gamma_i = 3$			
		<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>
400	40	10	2.87	5.16	1.44	10	3.62	7.08	3.36	10	3.62	8.51	4.56
	50	10	2.83	4.85	1.08	10	3.50	6.20	2.63	10	3.38	7.68	3.70
	60	10	2.76	5.24	1.12	10	3.58	6.75	2.88	10	3.38	7.47	3.55
500	40	10	2.85	5.02	1.34	10	3.70	6.64	2.97	10	3.46	7.82	3.89
	50	10	2.67	4.43	0.89	10	3.34	5.68	2.19	10	3.25	6.68	2.83
	60	10	2.60	5.03	0.89	10	3.29	6.08	2.18	10	3.17	6.65	2.72
600	40	10	2.70	4.92	1.06	10	3.57	6.68	2.94	10	3.47	7.34	3.64
	50	10	2.59	4.60	0.75	10	3.34	5.82	2.14	10	3.23	6.64	2.96
	60	10	2.53	4.89	0.62	10	3.22	6.25	1.83	10	3.14	6.67	2.26
Overall		90	2.71	4.90	1.02	90	3.46	6.35	2.56	90	3.34	7.27	3.35

Table 6: Results for CGH on instances with 250 targets.

M^C	E^C	$\Gamma_i = 1$				$\Gamma_i = 2$				$\Gamma_i = 3$			
		<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>
400	40	10	3.21	9.72	4.53	10	4.10	16.20	10.82	10	3.79	23.24	18.10
	50	10	3.27	10.84	5.35	10	4.03	22.29	16.46	10	3.70	26.45	20.76
	60	10	3.22	10.90	4.83	10	4.00	18.71	12.91	10	3.69	24.92	18.98
500	40	10	3.63	17.26	12.08	10	3.43	14.91	9.63	10	4.08	47.09	41.68
	50	10	3.28	11.80	5.82	10	3.84	21.14	15.21	10	3.72	21.32	27.28
	60	10	3.39	24.76	18.56	10	3.26	14.75	7.93	10	3.84	20.71	14.37
600	40	10	3.24	10.50	5.37	10	4.01	20.99	15.81	10	3.85	22.56	17.27
	50	10	3.14	12.25	6.21	10	3.99	21.76	15.48	10	3.71	24.14	18.13
	60	10	3.02	11.84	5.05	10	3.79	19.42	12.00	10	3.61	25.39	18.53
Overall		90	3.27	13.32	7.53	90	3.83	18.91	12.92	90	3.78	26.20	21.68

Table 7: Results for CGH on instances with 300 targets.

M^C	E^C	$\Gamma_i = 1$				$\Gamma_i = 2$				$\Gamma_i = 3$			
		<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>	<i>opt</i>	<i>gap</i>	<i>time</i>	<i>mas</i>
400	40	10	3.64	24.80	17.74	8	4.54	313.74	306.67	8	4.16	292.43	284.76
	50	10	3.51	29.47	22.08	7	4.41	380.99	373.17	8	4.16	292.37	283.65
	60	10	3.58	20.33	12.91	7	4.45	423.70	414.96	8	4.22	336.69	327.05
500	40	10	3.47	24.20	16.37	7	4.42	321.18	313.26	8	4.14	308.05	299.60
	50	10	3.45	29.09	19.90	8	4.34	397.10	386.77	8	4.07	339.14	329.61
	60	10	3.47	22.61	13.18	7	4.34	387.22	377.65	7	4.12	398.47	386.03
600	40	10	3.54	29.68	21.56	7	4.39	370.90	362.89	7	4.14	396.48	387.57
	50	10	3.36	20.98	13.28	7	4.30	384.96	375.72	8	4.00	354.12	344.90
	60	10	3.36	25.66	15.02	7	4.28	389.85	379.89	7	4.07	399.22	387.90
Overall		90	3.49	25.20	16.89	65	4.38	374.40	365.66	69	4.12	346.33	336.79

the best obtained solution after 1200 seconds and the LP bound for these instances is still less than 5%, however, which indicates that the integer master problem is probably nearly solved to optimality.

We also observe that as the number of targets increases, the average LP gap (between the best solution found and the LP bound) rises slightly but never exceeds 5% for any setting. It is worth pointing out that the real optimality gap between the solution output by CGH and the optimal objective-function value is typically far smaller than the value *gap*, since the LP bound is only an upper bound. Different values of Γ_i also lead to different LP gaps; this influence is analyzed more explicitly in Section 4.2.3.

The values of the memory capacity M^C and energy capacity E^C do not have much impact on the performance of CGH. This is because the number of candidate missions in the instances is never more than 2000, which ensures that the pricing problems can be efficiently solved by the DP algorithm. Even for the largest instances, the resolution

Table 8: The impact of the budget on the solutions and the runtime.

Γ_i	<i>obj</i>	<i>gap</i>	<i>time</i>
0	14570	1.08	3.73
1	13524	3.18	11.05
2	12989	3.67	18.50
3	12751	3.17	21.40
4	12625	3.20	25.04
5	12578	2.61	17.13
6	12554	2.11	12.43
7	12550	1.73	9.73
8	12545	1.44	7.28
9	12542	1.24	6.83
Overall	12923	2.34	13.31

of the LP relaxation (RMP) never exceeds 10 seconds, and most of the CGH runtime is spent on the integer master problem. The proportion of time spent on the integer program thus goes up as the number of targets increases. Especially for 300 targets and Γ_i equal to 2 or 3, the integer master problem occupies more than 95% of the total runtime.

4.2.3. Sensitivity analysis

The budgeted value Γ_i is the key parameter of the robust model. In this section, we examine its influence on the solutions and the runtime. We fix the energy and memory capacity of the satellites as 50 kJ and 500 MB, and 10 instances with 250 targets are used to conduct this small sensitivity analysis of the performance of CGH.

The simulation results are reported in Table 8, where column *obj* represents the average objective-function value over the 10 instances. Clearly, the highest objective value is obtained when $\Gamma_i = 0$, since then we simply do not consider cloud-coverage

uncertainty at all. The objective function decreases with larger Γ_i , which indicates that more conservative scheduling results are produced when we account for cloud-coverage uncertainty; the deterioration in the objective can be seen as the “price of robustness.” The decrease gradually diminishes with higher budget, and is very limited upon going from $\Gamma_i = 8$ to 9. This is logical: for each target, when the number of candidate missions is less than Γ_i , the achieved mission profit for target i remains unchanged.

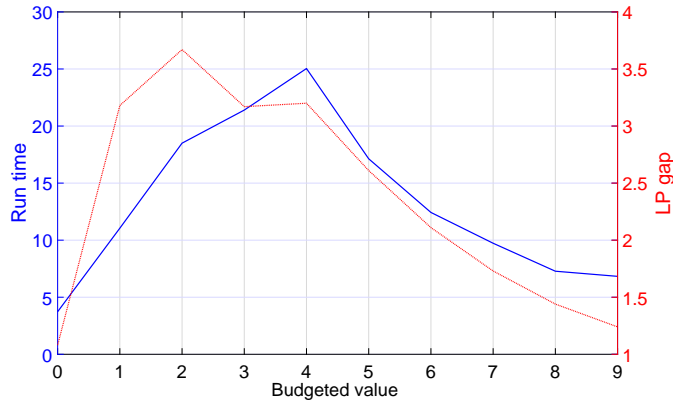


Figure 2: The influence of budgeted value on the runtime and LP gap.

As seen in the table and in Figure 2, the budget also has a significant impact on the runtime. The figure clearly shows an easy-hard-easy transition, where the average runtime first climbs with increasing Γ_i , then reaches a peak around 25 seconds for $\Gamma_i = 4$, and then goes down again when Γ_i increases beyond 4. For small Γ_i (≤ 4), the increase in runtime occurs because of the increasing difficulty of protection against uncertainty, with more non-dominated columns generated. For larger budget, protection against uncertainty becomes easier for more targets, and when Γ_i is greater than or equal to the number of

candidate missions of target i , all candidate missions with target i are forced to contribute their worst-case profit. The figure shows a similar pattern for the LP gap.

The length of the mission horizon also influences the performance of the proposed CGH. We select one instance with 250 targets and set the budgeted value as 2. The memory and energy capacities of the satellites remain as 50 kJ and 500 MB.

Table 9 contains the results of a sensitivity analysis with respect to the horizon length, where the columns Horizon and $|M|$ denote the mission horizon (in days) and the number of candidate observation missions, respectively. We observe that a longer horizon leads to more orbits, which provides a higher chance of generating more candidate observation missions and thus obtaining higher profits. A detailed computational procedure can be found in [52], where a rapid satellite-to-target access determination method is designed for generating candidate missions; the number of such missions is approximately linear in the horizon length. The profit growth declines for longer horizons, however. This is logical: each target's profit is upper bounded, and thus also the maximum attainable overall profit, such that the gap becomes smaller with longer horizons. The runtime also follows an easy-hard-easy transition with the horizon length. When the horizon is relatively short, CGH can struggle with large instances due to an increasing number of candidate observation missions; for larger horizons, the objective-function value gradually approaches the profit upper bound, making the model easy to solve.

Table 9: The impact of the horizon length.

Horizon	$ M $	<i>obj</i>	<i>gap</i>	<i>time</i>
1	386	7601	5.97	8.55
3	1119	11381	3.19	20.51
5	1758	13720	2.82	492.18
7	2348	15051	1.18	280.56
10	3400	15155	0.99	26.27
Overall	1802	12582	2.83	181.91

5. Conclusions

In this article, we have studied the robust scheduling of Earth observation satellites with cloud-coverage uncertainty. We aim to maximize the total observation profit, which is influenced by the conditions of cloud coverage. A robust model with budgeted uncertainty set is developed, where we control the price of robustness via a budget value. A compact robust formulation with rather poor computational performance is decomposed according to the different satellite orbits, and we propose a column-generation heuristic based on the reformulation. Computational results show that the original robust formulation can only solve small instances, while the proposed heuristic efficiently obtains near-optimal solutions also for large instances.

Future research on robust EOS scheduling with protection from cloud coverage may be oriented in several directions. Agile EOSs with strong attitude maneuvering ability can generate more candidates missions, but the scheduling complexity is obviously significantly increased by this agile character [53, 54]. Preliminary experiments demonstrate that the method proposed in this work cannot be readily applied for agile EOS scheduling

under uncertainty, which is still a largely unexplored topic. Another valuable avenue for future research is a more accurate modelling of the data transmission constraints (uplink and downlink). Finally, autonomous satellite scheduling systems have received significant attention recently [55, 56]; combining planning methods under uncertainty with autonomous planning platforms is also a promising opportunity for future engineering practice.

References

- [1] S. Nag, J. LeMoigne, and O. de Weck, “Cost and risk analysis of small satellite constellations for earth observation,” in *2014 IEEE Aerospace Conference*, 2014, pp. 1–16.
- [2] W. C. Lin, D. Y. Liao, C. Y. Liu, and Y. Y. Lee, “Daily imaging scheduling of an earth observation satellite,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 2, pp. 213–223, 2005.
- [3] V. Gabrel and D. Vanderpooten, “Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite,” *European Journal of Operational Research*, vol. 139, no. 3, pp. 533–542, 2002.
- [4] M. Vasquez and J.-K. Hao, “A logic-constrained knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite,” *Computational Optimization and Applications*, vol. 20, no. 2, pp. 137–157, 2001.

- [5] H. Kellerer, U. Pferschy, and D. Pisinger, “Multidimensional knapsack problems,” in *Knapsack problems*. Springer, 2004, pp. 235–283.
- [6] F. Perea, R. Vázquez, and J. Galán-Viogue, “Swath-acquisition planning in multiple-satellite missions: an exact and heuristic approach,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1717–1725, 2015.
- [7] T. A. Feo and M. G. Resende, “A probabilistic heuristic for a computationally difficult set covering problem,” *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, 1989.
- [8] B. Du, S. Li, Y. She, W. Li, H. Liao, and H. Wang, “Area targets observation mission planning of agile satellite considering the drift angle constraint,” *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 4, no. 4, pp. 047 002–047 020, 2018.
- [9] E. V. Ntagiou, C. Iacopino, N. Policella, R. Armellin, and A. Donati, “Ant-based mission planning: Two examples,” in *2018 SpaceOps Conference*, 2018, pp. 2498–2508.
- [10] E. Bensana, G. Verfaillie, J. Agnese, N. Bataille, and D. Blumstein, “Exact and inexact methods for the daily management of an earth observation satellite,” in *Proceeding of the International Symposium on Space Mission Operations and Ground Data Systems*, vol. 4, 1996, pp. 507–514.

- [11] B. Sun, W. Wang, and Q. Qi, “Satellites scheduling algorithm based on dynamic constraint satisfaction problem,” in *2008 International Conference on Computer Science and Software Engineering*, vol. 4, 2008, pp. 167–170.
- [12] W. J. Wolfe and S. E. Sorensen, “Three scheduling algorithms applied to the earth observing systems domain,” *Management Science*, vol. 46, no. 1, pp. 148 – 166, 2000.
- [13] S. Baek, S. Han, K. Cho, D. Lee, J. Yang, P. M. Bainum, and H. Kim, “Development of a scheduling algorithm and GUI for autonomous satellite missions,” *Acta Astronautica*, vol. 68, no. 7-8, pp. 1396–1402, 2011.
- [14] H. Kim and Y. K. Chang, “Mission scheduling optimization of SAR satellite constellation for minimizing system response time,” *Aerospace Science and Technology*, vol. 40, pp. 17–32, 2015.
- [15] V. Kolicic, X. Herrero, F. Xhafa, and L. Barolli, “Local search and genetic algorithms for satellite scheduling problems,” in *2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications*, 2013, pp. 328–335.
- [16] P. Tangpattanakul, N. Jozefowicz, and P. Lopez, “A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite,” *European Journal of Operational Research*, vol. 245, no. 2, pp. 542–554, 2015.
- [17] G. Wu, J. Liu, M. Ma, and D. Qiu, “A two-phase scheduling method with the con-

- sideration of task clustering for Earth observing satellites,” *Computers & Operations Research*, vol. 40, no. 7, pp. 1884–1894, 2013.
- [18] S. H. Durrani and K. Y. Jo, “Efficient scheduling algorithm for demand-assigned TDMA satellite systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 259–267, 1989.
- [19] M. Porretta, B. K. Schlarmann, A. Ballereau, and M. Crisci, “A novel uplink scheduling algorithm for the Galileo system,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 819–833, 2018.
- [20] J. Ju and D. P. Roy, “The availability of cloud-free Landsat ETM+ data over the conterminous United States and globally,” *Remote Sensing of Environment*, vol. 112, no. 3, pp. 1196–1211, 2008.
- [21] T. Algra, “On the effectiveness of cloud cover avoidance methods in support of the super-spectral mission for land applications,” in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, 2002, pp. 982–985.
- [22] D. Y. Liao and Y. T. Yang, “Imaging order scheduling of an earth observation satellite,” *IEEE Transactions on Systems, Man, and Cybernetics – Part C : Applications and Reviews*, vol. 37, no. 5, pp. 794–802, 2007.
- [23] W.-C. Lin and D.-Y. Liao, “A tabu search algorithm for satellite imaging schedul-

- ing,” in *2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, 2004, pp. 1601–1606.
- [24] W.-C. Lin and S.-C. Chang, “Hybrid algorithms for satellite imaging scheduling,” in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2005, pp. 2518–2523.
- [25] J. Wang, E. Demeulemeester, and D. Qiu, “A pure proactive scheduling algorithm for multiple earth observation satellites under uncertainties of clouds,” *Computers & Operations Research*, vol. 74, pp. 1–13, 2016.
- [26] C. K. Pang, A. Kumar, C. H. Goh, and C. V. Le, “Nano-satellite swarm for SAR applications: Design and robust scheduling,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 853–865, 2015.
- [27] X. Zhai, X. Niu, H. Tang, L. Wu, and Y. Shen, “Robust satellite scheduling approach for dynamic emergency tasks,” *Mathematical Problems in Engineering*, vol. 2015, pp. 1–20, 2015.
- [28] A. Gafurov and A. Bárdossy, “Cloud removal methodology from modis snow cover product,” *Hydrology and Earth System Sciences*, vol. 13, no. 7, pp. 1361–1373, 2009.
- [29] C.-H. Lin, P.-H. Tsai, K.-H. Lai, and J.-Y. Chen, “Cloud removal from multitemporal satellite images using information cloning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 232–241, 2013.

- [30] Z. Zou and Z. Shi, “Ship detection in spaceborne optical image with SVD networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5832–5845, 2016.
- [31] P. Kall and S. W. Wallace, *Stochastic Programming*. Springer, 1994.
- [32] D. Bertsimas and A. Thiele, “Robust and data-driven optimization: Modern decision making under uncertainty,” in *INFORMS TutORials in Operations Research*. INFORMS, 2006, pp. 95–122.
- [33] D. Bertsimas, D. Brown, and C. Caramanis, “Theory and applications of robust optimization,” *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.
- [34] A. L. Soyster, “Technical note-convex programming with set-inclusive constraints and applications to inexact linear programming,” *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973.
- [35] P. Kouvelis and G. Yu, *Robust Discrete Optimization and its Applications*. Springer Science & Business Media, 2013, vol. 14.
- [36] D. Bertsimas and M. Sim, “Robust discrete optimization and network flows,” *Mathematical Programming*, vol. 98, no. 1-3, pp. 49–71, 2003.
- [37] ———, “The price of robustness,” *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004.
- [38] R. Jiang, J. Wang, and Y. Guan, “Robust unit commitment with wind power and

- pumped storage hydro,” *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 800–810, 2012.
- [39] D. Bertsimas and A. Thiele, “A robust optimization approach to inventory theory,” *Operations Research*, vol. 54, no. 1, pp. 150–168, 2006.
- [40] I. Sungur, F. Ordóñez, and M. Dessouky, “A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty,” *IIE Transactions*, vol. 40, no. 5, pp. 509–523, 2008.
- [41] G. Song, D. Kowalczyk, and R. Leus, “The robust machine availability problem—bin packing under uncertainty,” *IIE Transactions*, vol. 50, no. 11, pp. 997–1012, 2018.
- [42] M. E. Lübbecke and J. Desrosiers, “Selected topics in column generation,” *Operations Research*, vol. 53, no. 6, pp. 1007–1023, 2005.
- [43] C. Han, X. Wang, G. Song, and R. Leus, “Scheduling multiple agile Earth observation satellites with multiple observations,” Department of Decision Sciences and Information Management, FEB, KU Leuven, Tech. Rep. KBI_1813, 2018.
- [44] P. C. Gilmore and R. E. Gomory, “A linear programming approach to the cutting-stock problem,” *Operations Research*, vol. 9, no. 6, pp. 849–859, 1961.
- [45] —, “A linear programming approach to the cutting stock problem - part II,” *Operations Research*, vol. 11, no. 6, pp. 863–888, 1963.

- [46] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Operations Research*, vol. 46, no. 3, pp. 316–329, 1998.
- [47] W. E. Wilhelm, “A technical review of column generation in integer programming,” *Optimization and Engineering*, vol. 2, no. 2, pp. 159–200, 2001.
- [48] T. Gschwind, S. Irnich, A.-K. Rothenbächer, and C. Tilk, “Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems,” *European Journal of Operational Research*, vol. 266, no. 2, pp. 521–530, 2018.
- [49] SpaceView, “Superview-1,” 2018, available at <http://www.spaceview.com/SuperView-1English/index.html>.
- [50] X. Liu, G. Laporte, Y. Chen, and R. He, “An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time,” *Computers & Operations Research*, vol. 86, pp. 41–53, 2017.
- [51] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen, “An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling,” *Computers & Operations Research*, vol. 100, pp. 12–25, 2018.
- [52] X. Wang, C. Han, P. Yang, and X. Sun, “Onboard satellite visibility prediction using metamodeling based framework,” *Aerospace Science and Technology*, vol. 94, p. 105377, 2019.

- [53] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, “Selecting and scheduling observations of agile satellites,” *Aerospace Science and Technology*, vol. 6, no. 5, pp. 367–381, 2002.
- [54] X. Wang, Z. Chen, and C. Han, “Scheduling for single agile satellite, redundant targets problem using complex networks theory,” *Chaos, Solitons & Fractals*, vol. 83, pp. 125–132, 2016.
- [55] G. Beaumet, G. Verfaillie, and M.-C. Charmeau, “Feasibility of autonomous decision making on board an agile earth-observing satellite,” *Computational Intelligence*, vol. 27, no. 1, pp. 123–139, 2011.
- [56] Z. Zheng, J. Guo, and E. Gill, “Onboard autonomous mission re-planning for multi-satellite system,” *Acta Astronautica*, vol. 145, pp. 28–43, 2018.