Annals of Operations Research
Manuscript Draft

Manuscript Number: ANOR-1815

Title: Solving the Earth observing satellite constellation scheduling problem by column generation

Article Type: Original Research

Keywords: Column generation; dynamic programming; satellite constellation; Earth observing

Annals of Operations Research manuscript No.
(will be inserted by the editor)

Solving the Earth observing satellite constellation scheduling problem by column generation

Pei Wang · Renjie He · Yuejin Tan

Received: date / Accepted: date

Abstract The scheduling problem of the Earth observing satellite constellation has to assign the unary resource of satellite observing and image-downloading time to different weighted spot targets on Earth, respecting special constraints, such as observing and image-downloading time windows, consecutive observing and data-downloading transition time, and on-board memory capacity. Since the problem is NP-hard, previous research mainly focused on heuristics and meta-heuristics. We first propose a mixed-integer program formulation, and then decompose the original problem into a set packing master problem and a longest path sub-problem with resource constraints and time windows. The sub-problem corresponds to determining the best single-satellite feasible schedule in order to improve the optimal solution value of the linear-relaxed restricted master problem. The linear-relaxed master problem is recursively solved by CPLEX, and the sub-problem is solved by a labeling-based dynamic programming in a graph. CPLEX is also used as an integer program solver taking the final linear optimal solution of the restricted master problem as the starting point. Realistic instances of Chinese environmental and disaster small satellite constellation are used to test the effectiveness of the proposed model and algorithm, and the experimental results show their advantage against a constraint-programming-based column generation method coded in ILOG CP.

Keywords Column generation · dynamic programming · satellite constellation · Earth observing

Mathematics Subject Classification (2000) 90B35 · 90B99

This work was supported by the National Natural Science Foundation of China (70601035, 70801062)

Pei Wang · Renjie He · Yuejin Tan
School of Information System and Management, National University of Defense Technology, Yanzheng Street 47, Changsha, 410073, Hunan, China
Tel.: +86-84575857
Fax: +86-84576490
E-mail: peigongliu@hotmail.com

1 Introduction

An important economical issue for a space agency like China Center for Resource Satellite Data and Applications (CRESDA) is the optimization of the schedule of its Earth observing satellites. Some satellites are formed as a constellation to fulfill observation requests of targets on Earth. A mission control center receives observation requests from various customers and is responsible for the control of the satellites. Since the requests usually exceed the observing capacity of the constellation, the management of the constellation consists of selecting a feasible subset of requests and scheduling the observation activities of the satellites to acquire images as well as scheduling the download activities to transmit the images back to a set of ground stations, with the objective to maximize the summed rewards of the targets observed and image-downloaded. These selected and scheduled sequences of observations and downloads of the satellites must comply with visibility time windows, observing and downloading durations, satellite memory capacity, and minimum transition times between observations and downloads.

Most of the literature in the field of satellite scheduling considered single-satellite single-orbit problems. Moreover, few researchers took into account both observation scheduling and download scheduling, instead studied these two parts separately. As to the observation scheduling problem, for non-agile satellites, i.e., they have no observation freedom along and across their trajectories (therefore the observation start time for each target is fixed), the scheduling problem is actually a selection problem (the selection here may have two folded meanings, i.e., selection of requests and selection of the observing window for each target); whereas, for agile satellites, i.e., they have the observation freedom (therefore the observation start time is variable), the problem is a selecting and scheduling problem. [Bensana et al (1996)] considered the non-agile satellite observation scheduling problem, used integer linear programming and valued constraint satisfaction formulations, and applied depth first branch and bound, Russian dolls search, greedy search and tabu search to solve the problem. [Vasquez and Hao (2001)] studied both the capacitated and uncapacitated (corresponding to whether the satellite memory capacity is considered or not) versions of SPOT5 satellite (a series of French Earth observing satellites) scheduling problem, and devised a tabu search method. [Cordeau and Laporte (2005)] took into account an agile satellite observation scheduling problem, and also used tabu search to solve instances whose schedule periods are only one-orbit long. [Gabrel et al (1997)] formulated an agile satellite scheduling problem in a graph, and proposed a branch and bound, and an approximate algorithm. [Lemaitre et al (2002)] compared four methods, i.e., greedy search, local search, constraint programming and dynamic programming for an agile satellite scheduling problem.

On the other hand, for the download scheduling problem, representative researchers are [Burrowbridge (1999)], [Barbulescu et al (2006)] and [Zufferey et al (2008)], and they used greedy algorithm, local search and tabu search respectively.

So far as we know, the integrated scheduling problem of observations and downloads of a satellite constellation received rare attention. Only [Frank et al (2001)] used a constraint-based interval planning framework to model the problem and proposed a stochastic heuristic search method, but they did not give any experimental result. [Florio (2006)] and [Bianchessi and Righini (2008)] addressed the problem with a look-ahead insert heuristic and a First-In-First-Out (FIFO) heuristic respectively. Yet they did not provide the mathematical model of the problem and neither of them considered the station contention between two different satellites when downloading consecutively.

In this paper we propose a mixed-integer program formulation for the satellite constellation scheduling problem considering the download contentions. The original model is decomposed into a set packing master problem and a longest path sub-problem with resource constraints and time windows. The constraint matrix of the restricted master problem is augmented through solving the sub-problem. The linear-relaxed master problem is solved by CPLEX and the sub-problem is solved by a labeling-based dynamic programming. Given a fractional master problem solution, CPLEX is also employed as an integer program solver to find the final integer solution to the original problem. We take the Chinese environmental and disaster small satellite constellation as the scenario constellation to test our proposed model and solution method, the computation results on a series of randomly-generated instances show a better performance than a constraint-programming based column generation method coded in ILOG CP.

The remainder of the paper is organized as follows: Section 2 describes the problem in details and presents a mixed-integer program formulation. Section 3 decomposes the model into a set packing master problem and a longest path sub-problem with resource constraints and time windows. Considering the download contention, some modifications to the master problem and the sub-problem are applied. Section 4 provides a column generation procedure for the whole problem and proposes a labeling-based dynamic programming to solve the sub-problem. Section 5 introduces the problem instances and presents the computational results. Section 6 draws the conclusion and points out the future directions.

2 Problem formulation

The Earth observing satellite constellation scheduling problem (EOSCSP) can be described as follows:

2.1 Problem description

- Given
  - a set of heterogeneous satellites with a given on-board memory capacity, each satellite is circling in different orbits

  - a set of spot targets with known rewards and image sizes (one target's image may occupy different amounts of space on different satellites) on Earth
  - a set of heterogeneous ground stations for receiving images from satellites
  - a set of observation time windows for each pair of satellite and target
  - a set of download time windows for each pair of satellite and ground station
- We are required to find a satellite constellation schedule such that
  - each target is observed by at most one satellite in at most one of their observation time windows
  - each observed target's image is downloaded to at most one ground station in at most one of the download time windows between the ground station and the satellite observing the target
  - the schedule of each satellite satisfies satellite capacity, time windows and activity (observation or download) transition time
  - total rewards of observed-and-image-downloaded targets are maximized

Among the problem elements, we feel obliged to emphasize on the time windows either between the satellites and the targets or between the satellites and the ground stations. Note that the time windows are hard constraints, which means that

- the problem solution is infeasible if any activity is scheduled beyond the upper bound provided by any time window
- there exists transition time, namely observation or download instrument adjustment time, from the former activity (observation or download) to the next immediate activity. And the transition time is sequence-dependent and is a prior. Actually, the transition time depends on the two consecutive activities' time windows, not on the two activities themselves
- the satellite must wait (keep idle) until the lower bound provided by any time windows if it adjusts its observation or download instrument too early

2.2 Notations

In order to formulate EOSCSP, we introduce the following notations:

- Satellite set K, each satellite $k \in K$ has a limited memory capacity Q_k. For simplicity, we transform the capacity to be the amount of time that takes the satellite's memory from completely empty to completely full
- N^P, set of observation requirements associated with targets
- N^{Pk}, set of observation requirements which satellite k can fulfill
- N^D, set of download requirements associated with targets. Because observation requirements and download requirements are in pair, $|N^P| = |N^D| = n$. For any observation requirement $i \in N^P$, $(n+i) \in N^D$ denotes the coupled download requirement.
- N^{Dk}, set of download requirements which satellite k can fulfill

- ALT_i^k, set of alternative opportunities of satellite k to fulfill requirement $i \in N^{Pk} \cup N^{Dk}$. Rewards, time windows, activity durations and memory demands are associated with such opportunities.
  - For opportunity $w \in ALT_i^k, i \in N^{Pk} \cup N^{Dk}$, its reward is v_w. We assume the reward of any observation opportunity is the same as the reward of the coupled download opportunity, for $w \in ALT_i^k, i \in N^{Pk}$, its reward $v_w = v_u$, where $u \in ALT_{n+i}^k$
  - For opportunity $w \in ALT_i^k, i \in N^{Pk} \cup N^{Dk}$, its time window is $[a_w^k, b_w^k]$, where a_w^k and b_w^k are the earliest start time and latest start time of w respectively
  - For opportunity $w \in ALT_i^k, i \in N^{Pk} \cup N^{Dk}$, its duration is d_w^k, and the duration is normally shorter than the associated time window span. We assume the duration of any observation opportunity is equal to the duration of the coupled download opportunity, for $w \in ALT_i^k, i \in N^{Pk}$, its duration $d_w^k = d_u^k$, where $u \in ALT_{n+i}^k$
  - For observation opportunity $w \in ALT_i^k, i \in N^{Pk}$, its memory demand is $l_w^k > 0$; whereas for the coupled download opportunity $u \in Alt_{n+i}^k$, its memory demand is $l_u^k = -l_w^k < 0$. For simplicity, we transform the memory demand of each observation opportunity to be the amount of time that takes the associated satellite to fulfill the observation, i.e., $l_w^k = d_w^k, l_u^k = -d_u^k$
- $o(k), d(k)$, start and end position of satellite k. The rewards, durations and memory demands of them are all 0, and time windows for $o(k)$ and $d(k)$ are $[0, 0]$ and $[H, H]$ respectively, where H is the end time of the schedule horizon
- $G^k = (V^k, A^k)$, directed graph associated with satellite k, where node set is $V^k = (\cup_{i \in N^{Pk}} ALT_i^k) \cup (\cup_{i \in N^{Dk}} ALT_i^k) \cup \{o(k), d(k)\}$, and the elements of arc set A^k satisfy the following condition: if $a_i^k + t_{ij}^k \le b_j^k$, then $(i, j) \in A^k$, where t_{ij}^k is the transition time between i and j
- decision variables
  - X_{ij}^k, binary flow variable, indicating whether satellite k executes j immediately after executing i
  - T_i^k, integer variable, indicating the exact start time of i for satellite k
  - L_i^k, integer variable, indicating the exact accumulated memory level for satellite k after executing i

2.3 Mixed-integer program model

EOSCSP could be formulated as follows:

$$\max \sum_{k \in K} \sum_{(i,j) \in A^k} v_i X_{ij}^k / 2 \tag{1}$$

$$\text{s.t.} \sum_{k \in K} \sum_{i \in ALT_w^k} \sum_{j:(i,j) \in A^k} X_{ij}^k \le 1, \quad \forall\, w \in N^P \tag{2}$$

$$\sum_{j:(o(k),j)\in A^k} X^k_{o(k),j} = \sum_{i:(i,d(k)\in A^k)} X^k_{i,d(k)} = 1, \quad \forall\, k \in K \tag{3}$$

$$\sum_{j:(i,j)\in A^k} X^k_{ij} - \sum_{j:(j,i)\in A^k} X^k_{ji} = 0, \quad \forall\, k \in K, \forall\, i \in V^k \setminus \{o(k), d(k)\} \tag{4}$$

$$X^k_{ij}(T^k_i + d^k_i + t^k_{ij} - T^k_j) \leq 0, \quad \forall\, k \in K, \forall\, (i,j) \in A^k \tag{5}$$

$$a^k_i \leq T^k_i \leq b^k_i, , \quad \forall\, k \in K, \forall\, i \in V^k \tag{6}$$

$$X^k_{ij}(L^k_i + l^k_j - L^k_j) = 0, \quad \forall\, k \in K, \forall\, (i,j) \in A^k \tag{7}$$

$$l^k_i \leq L^k_i \leq Q_k, \quad \forall\, k \in K, \forall\, w \in N^{Pk}, \forall\, i \in ALT^k_w \tag{8}$$

$$\sum_{j:(i_1,j)\in A^k} X^k_{i_1 j} - \sum_{j:(j,i_2)\in A^k} X^k_{j i_2} = 0, \quad \forall\, k \in K, \forall\, w \in N^{Pk}, \forall\, i_1 \in ALT^k_w, \forall\, i_2 \in ALT^k_{n+w} \tag{9}$$

$$T^k_i + t^k_{ij} - T^k_j \leq 0, \quad \forall\, k \in K, \forall\, w \in N^{Pk}, \forall\, i \in ALT^k_w, , \forall\, j \in ALT^k_{n+w} \tag{10}$$

$$L_{o(k)} = L_{d(k)} = 0, \quad \forall\, k \in K \tag{11}$$

$$X^k_{ij} \in \{0,1\}, \quad \forall\, k \in K, \forall\, (i,j) \in A^k \tag{12}$$

where (1) states that the objective is to maximize the overall rewards of the fulfilled requirements. Since in the final solution, observation requirements and download requirements are fulfilled in pairs, there is a multiplier of 1/2 in (1). (2) regulates that each observation requirement is at most fulfilled once in one of its alternative opportunities. Because observation requirements and download requirements are fulfilled in pairs, this constraint also implies that each download requirement is at most fulfilled once in one of its alternative opportunities. (3) implies that for every satellite, it has exactly one start activity and one end activity. (4) constrains that every executed activity has exactly one predecessor and one successor. (5) states that for every pair of consecutively executed activities, there must be enough transition time between the end of the former activity and the start of the latter activity. (6) constrains that for every executed activity, its start time must be within the selected time window. (7) regulated that for every pair of consecutively executed activities, the memory load difference between them is just the memory demand of the first activity. (8) constrains that for every observation requirement, after fulfilling it, the memory load must be greater than the memory demand of the observation, and be smaller than the capacity of the associated satellite. (9) is the coupled constraint which restricts that the observation requirement and the download requirement belonging to the same target must be fulfilled by the same satellite. (10) restricts that the start time of the observation must be ahead of the start time of the coupled download. (11) regulates that the start and end condition of the memory load level of every satellite are both empty. (12) is the binary requirement of the arc-flow variables.

3 Danzig-Wolfe decomposition for the original model

The above formulation has the block diagonal structure, and constraints (3)-(12) are separable by satellite; what is more, they define the feasible region of an elementary longest path problem with constraints on two resources: time (time windows can be considered as a special resource) and memory load. Since every extreme point of the feasible region defined by constraints (3)-(12) corresponds to a feasible $o(k) - d(k)$ path in G^k, we reformulate the model from an arc-flow formulation into a path-flow formulation, and at the same time we decompose the original model by satellite. And we need the following additional notations:

3.1 Notations

- Ω^k: set of feasible schedule (a subset of all the feasible paths in G^k including the empty $o(k)-d(k)$ path, note that due to the resource constraints, every feasible schedule for satellite k corresponds to a feasible path in G^k, and the vice versa is not necessarily true) for satellite k
- c_p^k: reward of feasible path $p \in \Omega^k$
- ρ_{pi}^k: binary parameter equal to 1 if path $p \in \Omega^k$ visits node i
- θ_p^k: binary variable equal to 1 if path $p \in \Omega^k$ is adopted

3.2 Master problem

The path-flow formulation is as follows:

$$\max \sum_{k\in K} \sum_{p\in \Omega^k} c_p^k \theta_p^k \tag{13}$$

$$\text{s.t.} \sum_{k\in K} \sum_{i\in ALT_w^k} \sum_{p\in \Omega^k} \rho_{pi}^k \theta_p^k \leq 1, \quad \forall\, w \in N^P \tag{14}$$

$$\sum_{p\in \Omega^k} \theta_p^k = 1, \quad \forall\, k \in K \tag{15}$$

$$\theta_p^k \in \{0,1\}, \quad \forall\, k \in K, p \in \Omega^k \tag{16}$$

The linear relaxation of this formulation is our master problem (MP), which is solved by column generation. The following two dual variables are introduced:

- $\pi_w, w \in N^P$: dual variables associated with constraint (14)
- $\eta_k, k \in K$: dual variables associated with constraint (15)

Therefore, the reduced cost $\bar{c}_p^k$ of variable θ_p^k is as follows:

$$\bar{c}_p^k = c_p^k - \sum_{w\in N^P} \sum_{i\in ALT_w^k} \rho_{pi}^k \pi_w - \eta_k = \sum_{w\in N^P} \sum_{i\in ALT_w^k} \rho_{pi}^k (v_i - \pi_w) - \eta_k \tag{17}$$

Since the rewards of all the possible opportunities for a given observation requirement w are the same, we introduce the reduced cost of any opportunity $i \in ALT_w^k$ of the observation requirement w, and assume $\pi_i = \pi_w$. Thus $(v_i - \pi_i)$ is the node reduced cost.

3.3 Sub-problem

The following is the sub-problem (SP) formulation:

$$\max \sum_{(i,j)\in A^k} (v_i - \pi_i) X_{ij}^k / 2 \tag{18}$$

$$\text{s.t.} \quad (3) - (12) \tag{19}$$

The domain complications involved in EOSCSP (i.e. multiple time windows, coupled observation and download, sequence between observation and download, transition time, memory capacity) are not explicitly shown in MP. Instead, all these complexities are embedded in the columns of MP, in other word, in SP. The advantage of this is that all the complications can be handled locally at the level of the individual satellite, and EOSCSP' s complexity is distributed to different satellites, leaving the space for some efficient algorithms to solve SP.

3.4 Modification to the model

Up to now, either the original formulation or the reformulation does not consider the download contention for one certain ground station between different satellites when downloading images consecutively. As a disjunctive resource, every ground station can receive images from at most one satellite at any time. In the original formulation, we only take into account the alternative download opportunities (these opportunities are provided by different stations with different satellites) of every download requirement. And the consecutive download transition time in the original model is from the satellite point of view, and the satellite is a disjunctive resource. The consecutive transition time for every station (image-downloading for two satellites to the station or image-receiving for the station from the two satellites) is left unsettled. If the image-receiving transition time of every station is respected, the model will be further complicated. In order to strike a balance between the content-richness and the treatablilty of the model, we make such a simplification: assume that all the download opportunities come from one ground station, which means that, for some download opportunities provided by one certain satellite with different stations, if their time windows are overlapping, they will be combined, and the time lower bound of the combined one is the smallest among the lower bounds of all the involved original time windows, and the upper bound of the combined one is the biggest among the upper bounds of all the involved original time windows.

After the above combination, in order to enforce that at any time, the station can receive images from at most one satellite, we make the following preprocessing of the combined download time windows from different satellites. For instance, if satellite k_1 has one download time window $[150, 240]$ and satellite k_2 has another download time window $[210, 260]$, the preprocessing will count the non-overlapped parts of the two time windows as two new download time windows, and will also count the overlapped part as one single new download time windows:

$[150, 240], [210, 260] \Rightarrow [150, 210], [210, 240], [240, 260]$

Note that between the non-overlapped part and the overlapped part, such as between $[150, 210]$ and $[210, 240]$, there is no gap, i.e., the transition time is 0. The schedule practice in the ground station justifies this handling method. We can achieve this seamless transition through increasing the number of re-

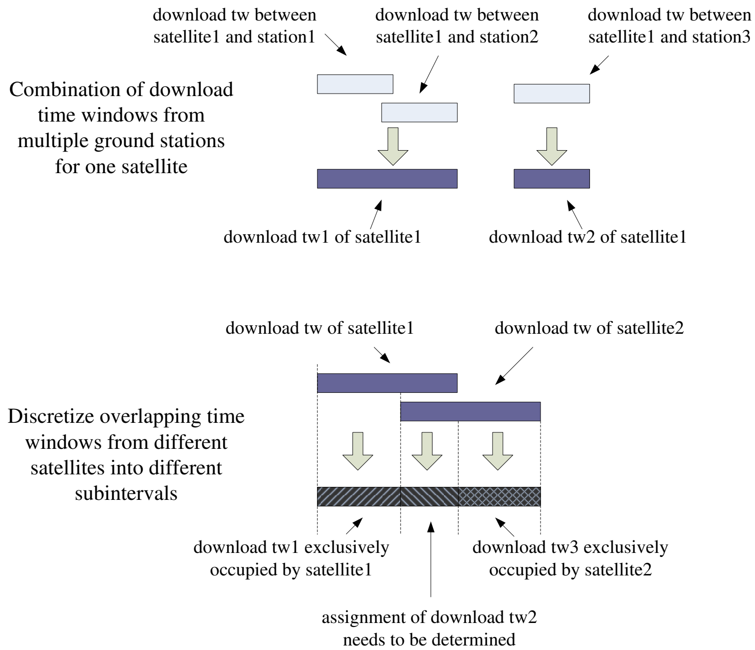

Fig. 1 Combination and preprocessing of download time windows

ceiving antennas in the station or making preparation ahead of time, because the condition is not so strict on ground as on-board, and the activity transition time for every satellite is normally greater than 0. Through the above combination and preprocessing, we get a new set DTW of download time windows for all the satellites, and there is no overlapping between any two from the set. Figure 1 illustrates the combination and preprocessing of the download time windows.

We introduce a new constraint into the original model regulating that for every download time window in DTW, there must be at most one satellite using it.

$$\sum_{k \in K} Y_i^k \leq 1, \quad \forall\, i \in DTW \tag{20}$$

where Y_i^k is a binary variable equal to 1 if satellite k adopts time window i in DTW for image downloading. Consequently, MP is added a new constraint as follows:

$$\sum_{k \in K} \sum_{p \in \Omega^k} \delta_{pi}^k \theta_p^k \leq 1, \quad \forall\, i \in DTW \tag{21}$$

where δ_{pi}^k is a binary parameter equal to 1 if path $p \in \Omega^k$ adopts time window i for image downloading. Therefore, the reduced cost $\bar{c}_p^k$ of variable θ_p^k is modified as follows:

$$\sum_{w \in N^P} \sum_{i \in ALT_w^k} \rho_{pi}^k (v_i - \pi_w) - \sum_{i \in DTW} \lambda_i \delta_{pi}^k - \eta_k \tag{22}$$

where $\lambda_i, i \in DTW$ is the dual variables associated with constraint (21). The impact of the modification on SP, actually on the graph G^k, is that the nodes in the set $(\cup_{i \in N^{Dk}} ALT_i^k)$ are replaced by the new generated download time windows through the combination and the preprocessing. Thus, it is possible that several observation requirements will share the same download time window in the revised G^k of satellite k. On one hand, the number of the download time windows of every satellite may be decreased, therefore the associated graph G^k is reduced (We make the assumption that the time windows after preprocessing are large enough to accommodate the download requirements associated with them respectively). On the other hand, this will incur new decision-making with regard to the shared download time window. To be precisely, before the modification, we should only determine whether a download time window is adopted when its coupled observation is already fulfilled; whereas, after the modification, when it comes to a download time window, we should not only determine whether the download time window is adopted, but also determine how many images should be downloaded at the time window if the time window is adopted. Because in that case, all the adopted observations whose finishing time are before this download time window could share the time window. For example, if a download time window is $[30, 40]$ and three executed observations are $[5, 7], [13, 17], [24, 27]$, because the required download quantity is smaller than the capacity the download time window could supply, i.e., $(7-5)+(17-13)+(27-24) < (40-30)$, we should decide how many images should be downloaded at this time window, and we have $n+1$ choices when the image access mechanism of the satellite memory is 'FIFO (First-in-First-out)', where n is the number of observations executed before the download time window. When the image access mechanism of the satellite memory is 'RAM (just like the normal computer file system)', we have 2^n choices.

The reason why the download time window is not fully employed to satisfied the download requirements even when the download time window capacity is larger than the required download amount is as follows: it is possible that the download time window overlaps with some observation time windows. The improper amount of image-downloading may therefore result in that the overlapped observation time windows cannot be used as observation. So the amount of image-download in the download time window is also a decision variable in such case. Figure 2 shows the case. If such overlapping does not exist, then the download time window should be fully used without doubt.

We did not reflect the download amount variable in our MP, instead, we handle this decision making in our SP solution methodology. From the selection of every adopted observation time window and the download amount in the download time window, we can deduce that which target's image is downloaded at which download time window.

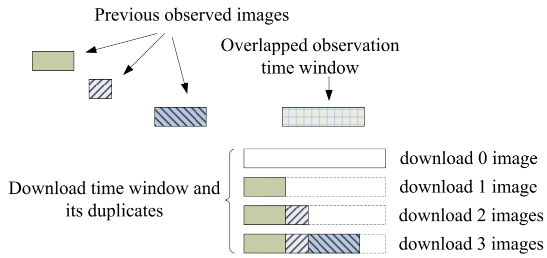

Fig. 2 Download amount should be decided when overlapping with observations exists

4 Solution methodology

4.1 Column generation for EOSCSP

The column generation procedure for EOSCSP is as follows:

1. Solve the linear relaxation of MP using 0 column.
2. For each satellite k, solve SP: if its objective value $> \eta_k$, then include the optimal schedule in the linear relaxation of MP; otherwise the linear relaxation of MP has been solved, stop.
3. Re-solve the linear relaxation of MP and go to Step 2.

4.2 Dynamic programming for SP

The SP associated with satellite $k \in K$ is actually an elementary longest path problem with resource constraints (ELPPRC), where elementary means that revisits to any node are not allowed, that is to say, cycles in the solution are forbidden. A similar problem, i.e., elementary shortest path problem with resource constraints (ESPPRC) is proved to be NP-hard [Desaulniers et al (1994)], and is usually solved by dynamic programming, or more precisely, by a labeling algorithm. Considering the differences between ELPPRC and ESPPRC, we devise a labeling algorithm to solve ELRRRC. In the algorithm, partial paths start at the source node $o(k)$ and are represented by multi-dimensional resource vectors, called labels. Starting from an initial label associated with node $o(k)$, labels are propagated forwardly using resource extension functions (REFs) through graph G^k. Because a resource is actually a quantity that varies along a path and its value must fall within a pre-specified range, called resource window at each node, in order to enforce the path feasibility, resource windows are checked at each node to discard infeasible partial paths. Since multiple labels are associated with each node, the number of feasible paths may be exponential. In order to avoid enumerating all feasible paths, a dominance rule is applied to discard unpromising labels. Some details of the algorithm are provided in the following parts.

4.2.1 Labels for the nodes

A label B_i^p represents a partial path p from node $o(k)$ to a node i contains the five following components

- C_i^p: reduced cost of p
- L_i^p: memory load accumulated along p after servicing node i
- T_i^p: earliest end of service time at node i
- W_i^p: set of visited (either only observed or observed-and-image-downloaded) targets or download time windows along p after servicing node i
- Z_i^p: set of observed-but-not-yet-image-downloaded observations along p before servicing node i

And $B_i^p = (C_i^p, L_i^p, T_i^p, W_i^p, Z_i^p)$ is feasible if

$$L_i^p \in [0, Q_k] \quad and \quad T_i^p \in [a_i^k + d_i^k, b_i^k + d_i^k] \tag{23}$$

4.2.2 Resource extension functions (REFs)

When relaxing the elementary requirement, path p (associated label B_i) can be extended by appending arc $(i,j) \in A^k$ to it, then the resulting path is represented by a label $B_j = (C_j, L_j, T_j, W_j, Z_j)$ whose components are computed using the following resource extension functions

- $C_j = f_{ij}^{cost}(B_i) = \begin{cases} C_i + v_j & j \in (\cup_{u \in N^{Pk}} ALT_u^k) \\ C_i & j \in DTW^k \end{cases}$, when j is an observation time window, the reduced cost is increased by the observation's reward; otherwise, the downloading will not result in reduced cost improvement (the rewards of the coupled observation and download will only be counted once in the objective, as well as in the reduced cost).
- $L_j = f_{ij}^{load}(B_i) = \begin{cases} L_i + l_j^k & j \in (\cup_{u \in N^{Pk}} ALT_u^k) \\ L_i - \sum\limits_{j' \in \tilde{Z}_j} l_{j'}^k & j \in DTW^k \end{cases}$, where $\tilde{Z}_j$ is a subset of Z_j, denoting the selected observed-but-not-yet-image-downloaded observations to be downloaded in download time window j. And the elements in $\tilde{Z}_j$ are determined by the downloading schedule policy and the capacity of the associated download time window j. Currently, we only consider the 'FIFO' image access mechanism discussed before. Note that $\tilde{Z}_j$ could be an empty set. When j is an observation time window, the memory level is increased by the observation's memory occupation; otherwise, the memory level is decreased by the total memory occupations of all the elements in $\tilde{Z}_j$
- $T_j = f_{ij}^{time}(B_i) = \begin{cases} max\{a_j^k + d_j^k, T_i + t_{ij}^k + d_j^k\} & j \in (\cup_{u \in N^{Pk}} ALT_u^k) \\ max\{a_j^k + \sum\limits_{j' \in \tilde{Z}_j} d_{j'}^k, T_i + t_{ij}^k + \sum\limits_{j' \in \tilde{Z}_j} d_{j'}^k\} & j \in DTW^k \end{cases}$

– $W_j = f_{ij}^{target}(B_i) = \begin{cases} W_i \cup \{j\} & j \in (\cup_{u \in N^{P_k}} ALT_u^k) \\ W_i \cup \{j\} & j \in DTW^k \text{ and } \tilde{Z}_j \neq \phi \\ W_i & j \in DTW^k \text{ and } \tilde{Z}_j = \phi \end{cases}$

– $Z_j = f_{ij}^{observe}(B_i) = \begin{cases} Z_i \cup \{i\} & i \in (\cup_{u \in N^{P_k}} ALT_u^k) \\ Z_i \setminus \tilde{Z}_i & i \in DTW^k \end{cases}$

Note that REFs $f_{ij}^{time}()$ ensure that time window lower bounds are always respected. Therefore, B_j is feasible if $L_j \leq Q_k$ and $T_j \leq b_j^k$.

4.2.3 Dominance

The principle of the dominance rule is as follows: consider two labels (B^1 and B^2) representing two feasible partial paths ending at the same node, and from B^2 we cannot obtain a longest $o(k) - d(k)$ path if

– Every feasible extension of B^2 is also a feasible extension of B^1
– For every such extension, the reduced cost of the path resulting from the extension of B^2 is less than or equal to the reduced cost of the path resulting from the extension of B^1

Under such circumstances, we say that label $B^1 = (C^1, L^1, T^1, W^1, Z^1)$ dominates label $B^2 = (C^2, L^2, T^2, W^2, Z^2)$, and the latter can then be discarded. In practice, the dominance rule is applied when the following conditions are satisfied

– $C^1 \geq C^2$
– $L^1 \leq L^2$
– $T^1 \leq T^2$
– $W^1 \subseteq W^2$
– $Z^1 \subseteq Z^2$

Otherwise, keep both the two labels for further extension.

4.2.4 Elementary restriction

In order to ensure the elementary requirement, we have to forbid any element in W_i^p reemerging in the set. If we extend path p by appending $(i, j) \in A^k$, and find that j is already in W_i^p, then the extension is withdrawn. This trial-and-check, then extend-or-withdraw mechanism is added to the above REFs.

4.2.5 Algorithm structure

The notations used by the algorithm are as follows

– UN_i: set of unprocessed labels at node i
– PR_i: set of processed labels at node i
– $i(B)$: last node of the path associated with label B

– $DOM(UN_i, PR_i)$: dominance rule applied to labels in UN_i and PR_i that returns a possibly reduced set UN_i, and the rule is the multiple arithmetic operations and set-include judgment operation mentioned in the previous Dominance subsection.

Following is the algorithm structure

Algorithm 1 ELPPRC labeling algorithm

```
Set UN_i = PR_i = φ for all i ∈ V^k except UN_o(k) = {(0, 0, 0, o(k), φ)}
while ∪_{i∈V^k} UN_i ≠ φ do
    Choose the label B ∈ ∪_{i∈V^k} UN_i with smallest T and remove B from UN_i(B)
    for all arcs (i(B), j) ∈ A^k do
        Extend B along (i(B), j) using the REFs to create a new label B̄
        if B̄ is feasible then
            Add B̄ to UN_j
            UN_j = DOM(UN_j, PR_j)
    Add B to PR_i(B)
Decode PR_d(k) to find a longest o(k) − d(k) path
```

4.3 Obtain integer solution through CPLEX

Our computational experiments show that the optimal solution of the linear relaxation of MP is integral in some cases, and in most other cases, few variables in the solution have fractional values. Therefore, we directly apply CPLEX as an IP solver to solve the last restricted MP. This usually takes not so much time to get an integer solution when there are large number of columns in MP.

5 Computational results

The dynamic programming algorithm was coded in C# and integrated with ILOG CPLEX 12.1, and CPLEX was responsible for solving MP and obtaining the dual variable values. The column generation framework was also implemented in CPLEX. The whole algorithm was run on a PC (Intel Pentium3, 2.4 GHz, 1GB memory). We compared our column generation algorithm (denoted as DP in Table 1) with another column generation algorithm (denoted as CP in Table 1) which uses constraint programming as SP solution method. The two column generation algorithms only differ in SP solution methods. The constraint programming algorithm was coded in ILOG CP. ILOG CP is a constraint programming modeler and solver, and the solver is based on self-adaptive large neighborhood search, and it encapsulates the handling of the renewable resource ([Laborie and Godard (2007)]), such as satellite memory. After formulating SP in a constraint optimization model, we use CP as a black-box to solve the model.

We generated 17 EOSCSP instances. In Table 1, columns k, n and H represent the number of satellites, the number of targets and the number of days of scheduling respectively. Columns $\#Iters$ and $\#Cols$ show the counts of iterations and the number of generated columns respectively. The unit of all the computational times are in seconds. Satellites are the two environmental monitoring satellites HJ-1A and HJ-1B in the Chinese environmental and disaster small satellite constellation. Targets are randomly generated on Earth, and their rewards are random integer number in [1,10].

In DP, in every iteration, SP is solved once to optimal, so the number of iterations is equal to the number of generated columns, and the column $\#Cols$ is neglected in DP part in the table. In CP, in every iteration, SP is solved several times, and CP only tries to find a good feasible solution to SP instead of an optimal solution. DP is able to optimally solve the linear relaxation of MP for all the instances in acceptable time. From Table 1 we can see that DP outperforms CP in both optimal solution value and computational time on average. For large instances, such as instances $15, 16, 17$, the pricing time of DP is quite long because of the search for the optimal of SP. However, for small and medium sized instances, the pricing time of DP is mostly much shorter than CP. Note that CP sometimes terminates in finding good feasible solutions to the linear relaxation of MP. The MP solving and the integer solution searching are very fast for both column generation methods. However, we cannot guarantee that directly using CPLEX as an integer solver and using the optimal to the linear relaxation to MP can obtain integer optimal, although we conjecture this is true.

6 Conclusion

In this paper we describe and formulate the Earth observing satellite constellation scheduling problem with download contentions. Danzig-Wolfe reformulation is used to decompose the original mixed-integer model into a set packing master problem and a longest path sub-problem with resource constraints and time windows. CPLEX and a special labeling-based dynamic programming method are used to solve the master problem and the sub-problem. CPLEX is also used to obtain the integer solution to the master problem. Computational results show that the proposed model and algorithm generally outperforms another constraint-programming-based column generation method in both optimal solution value and computational time.

In the proposed model, the download contention is preliminarily considered. We simplify that the ground stations are replaced by a monolithic ground station, and all the download time windows are provided by this special station. Future research should take into account more general cases with more ground stations, and these stations should be interchangeable. Another prospective direction should be further polishing and improving the sub-problem solution method, such as introducing special data structures ([Righini (2006)]) and novel ideas in time constrained shortest path problems, for example, bi-directional dynamic programming ([Righini and Salani (2009)]), to shorten computational time, especially when solving large instances.

Acknowledgements We would like to thank Alberto Ceselli and Giovanni Righini from University of Milan as well as Gerhard Reinelt from University of Heidelberg, without the inspiring discussions with them, this work will not be done. Some of the research was undertaken when the authors were visiting scholars in the above two universities.

References

[Barbulescu et al (2006)] Barbulescu L, Howe A, Whitley D, (2006) AFSCN scheduling: How the problem and solution have evolved. Mathematical and Computer Modelling 43(9-10):1023 – 1037

[Bensana et al (1996)] Bensana E, Verfaillie G, Agnese J, Bataille N, Blumstein D, (1996) Exact and approximate methods for the daily management of an earth observation satellite. In: Proc. of the 4th International Symposium on Space Mission Operations and Ground Data Systems

[Bianchessi and Righini (2008)] Bianchessi N, Righini G, (2008) Planning and scheduling algorithms for the COSMO-SkyMed constellation. Aerospace Science and Technology 12(7):535–544

[Burrowbridge (1999)] Burrowbridge S, (1999) Optimal allocation of satellite network resources. Master's thesis, Virginia Polytechnic Institute and State University, USA

[Cordeau and Laporte (2005)] Cordeau J, Laporte G, (2005) Maximizing the value of an earth observation satellite orbit. Journal of the Operational Research Society 56(8):962–968

[Desaulniers et al (1994)] Desaulniers G, Desrosiers J, Ioachim I, Solomon M, Soumis F, (1994) A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. Cahiers du GERAD

[Florio (2006)] Florio SD, (2006) Performances optimization of remote sensing satellite constellations: a heuristic method. In: Proc. of the 5th International Workshop on Planning and Scheduling for Space

[Frank et al (2001)] Frank J, Jonsson A, Morris R, Smith D, (2001) Planning and scheduling for fleets of earth observing satellites. In: Proc. of the 6th International Symposium on Artificial Intelligence, Robotics and Automation for Space

[Gabrel et al (1997)] Gabrel V, Moulet A, Murat C, Paschos V, (1997) A new single model and derived algorithms for the satellite shot planning problem using graph theory concepts. Annals of Operations Research 69:115–134

[Laborie and Godard (2007)] Laborie P, Godard D, (2007) Self-adapting large neighborhood search: Application to single-mode scheduling problems. Proceedings MISTA-07, Paris pp 276–284

[Lemaitre et al (2002)] Lemaitre M, Verfaillie G, Jouhaud F, Lachiver J, Bataille N, (2002) Selecting and scheduling observations of agile satellites. Aerospace Science and Technology 6(5):367–381

[Righini and Salani (2009)] Righini G, Salani M, (2009) Decremental state space relaxation strategies and initialization heuristics for solving the Orienteering Problem with Time Windows with dynamic programming. Computers & Operations Research 36(4):1191–1203

[Righini (2006)] Righini G, (2006) A mathematical programming algorithm for planning and scheduling an earth observing SAR constellation. In: Proc. of the 5th International Workshop on Planning and Scheduling for Space

[Vasquez and Hao (2001)] Vasquez M, Hao J, (2001) A logic-constrained knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. Computational Optimization and Applications 20(2):137–157

[Zufferey et al (2008)] Zufferey N, Amstutz P, Giaccari P, (2008) Graph colouring approaches for a satellite range scheduling problem. Journal of Scheduling 11(4):263–277

Table 1 Experiment results: dynamic programming-based column generation versus constraint programming-based column generation

Inst. Info			DP						CP						
k	n	H	#Iters	Master Time	Pricing Time	Linear Obj.	Integer Obj.	Integer Time	#Iters	#Cols	Master Time	Pricing Time	Linear Obj.	Integer Obj.	Integer Time
2	20	0.5	9	< 1	< 1	58.0	58	< 1	22	37	< 1	85	58.0	58	< 1
2	20	0.5	7	< 1	< 1	62.0	62	< 1	16	30	< 1	67	62.0	62	< 1
2	20	0.5	16	< 1	< 1	75.0	74	< 1	49	94	< 1	542	74.9	74	< 1
2	20	0.5	10	< 1	< 1	58.0	58	< 1	29	42	< 1	812	58.0	58	< 1
2	20	0.5	11	< 1	< 1	77.0	73	< 1	19	34	< 1	< 1	77.0	67	< 1
2	50	0.5	4	< 1	6	84.0	84	< 1	16	30	< 1	349	79.0	79	< 1
2	50	0.5	8	< 1	25	91.5	91	< 1	24	28	< 1	945	88.0	88	< 1
2	50	0.5	7	< 1	11	82.0	82	< 1	15	23	< 1	697	78.0	78	< 1
2	50	1	15	< 1	40	90.7	89	< 1	33	61	< 1	1000	86.4	86	< 1
2	50	1	19	< 1	35	95.3	95	< 1	35	65	< 1	561	86.0	86	< 1
2	50	1	16	< 1	47	118.5	118	< 1	61	96	< 1	2195	113.0	113	< 1
2	100	1	16	< 1	665	173.8	173	< 1	31	60	< 1	242	147.5	146	< 1
2	100	1	28	< 1	1803	160.0	156	< 1	49	83	< 1	1425	131.0	130	< 1
2	100	1	27	< 1	2176	167.5	167	< 1	27	187	< 1	2176	167.5	167	< 1
2	100	2	55	< 1	7274	322.5	195	< 1	117	217	< 1	2292	238.0	172	< 1
2	100	2	48	< 1	7478	331.9	180	< 1	76	147	< 1	922	271.1	163	< 1
2	100	2	49	< 1	7247	344.8	250	< 1	89	153	< 1	1941	277.4	244	< 1

Solving the Earth observing satellite constellation scheduling problem by column generation

Pei Wang[*], Renjie He, Yuejin Tan
College of Information Systems and Management
National University of Defense Technology
47 Yanzheng Street, Changsha 410073, Hunan China

December 14, 2010

*Corresponding author; Tel: +86-84575857, Fax: +86-84576490, Email: peigongliu@hotmail.com