Ground stations networks for Free-Space Optical communications: maximizing the data transfer

Mikaël Capelle, Marie-José Huguet, Nicolas Jozefowiez, Xavier Olive

Ground stations networks for Free-Space Optical communications: maximizing the data transfer

Mikaël Capelle [a,b,1] Marie-José Huguet [b] Nicolas Jozefowiez [b]
Xavier Olive [a]

[a] *Thales Alenia Space, Toulouse, France*

[b] *LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France*

Abstract

Free-space optical communications are becoming a mature technology, but unlike current radio-frequency technologies, they are strongly impacted by clouds. In this paper, we aim to find a network of optical ground stations maximizing the amount of data that can be sent from a low-earth orbiting satellite to the Earth during its missions, taking into consideration cloud information. We present a Mixed Integer Linear Program (MILP) and a hierarchical method based on an exhaustive enumeration of the sets of ground stations and on a dynamic programming algorithm and compare them on real scenarios based on archived years of cloud data. Even if the MILP can solve scenarios over small time horizons in less than one hour, experiments show that the hierarchical approach outperforms it in term of CPU time while achieving optimality.

Keywords: Free-Space Optical Communications, Optical Ground Station Network Optimisation, Dynamic Programming, Mixed-Integer Linear Programming.

[1] Email: `mikael.capelle@thalesaleniaspace.com`

1 Introduction

Free-Space-Optical (FSO) communications are seen as a key technology [6] [8] [11] to cope with the needs of higher data-rate payloads for future low-earth orbiting (LEO) observation satellites in replacement to the current radio-frequency (RF) technologies. Current RF technologies use X-Band for download which can currently provide up to a few Gbps [2]. These are not impacted by weather or atmospheric turbulences. Their main drawbacks are limited data-rates and a need for frequency licensing which will become a major issue in the upcoming years due to increases in the number of operational satellites and constellations. *FSO* communications offer data-rate order(s) of magnitude higher than current RF technologies: targeted data-rates go from some tens of Gbps to some Tbps. They do not require frequency licensing and are hard to intercept by malicious observers. However, *FSO* communications are strongly impacted by weather, cloud, and atmospheric turbulences.To evaluate the impact of FSO communications for spatial imagery systems, we aim to find a subset of optical ground stations in order to maximize the percentage of data downloaded from satellites taking cloud information into account using archived data from previous decades.

The optimization of an *Optical Ground Stations Network* taking into account the influence of clouds was first studied by [7] and [13]. Their objective was to find a network for a deep-space probe in order to reach a given temporal availability using an approximation algorithm with a high-resolution database as input. In [12], a probabilistic approach was used to analyze the availability of ground station networks: one in Japan for a geostationary satellite and one worldwide for a low-earth orbiting satellite (LEO) satellite. In [10] and [9], a greedy algorithm was used to find a network in Europe for a geostationary satellite based on data from the *SAF-NWC* cloud database and using an hypothetical substrate network.

In 2012, the *Optical Link Study Group (OLSG)* published a report in which the impact of FSO communications on various space systems (LEO observation satellite, geostationary relay) was evaluated using the *Lazercom Network Optimization Tool (LNOT)*[7].

In Section 2, we will characterize more formally the problem under study and propose a Mixed Integer Linear Program. In Section 3, we will present a hierarchical method based on a dynamic programming algorithm to solve this problem. We will then present our experimental context and the results of our experimentations in Section 4.

2 Problem statement and MILP modelisation

2.1 Problem definition

Considering a LEO satellite and given a set $\mathcal{L} = \{1, \ldots, N\}$ for possible locations of ground stations with associated costs w_i $(i \in \mathcal{L})$, we aim to find a subset $\mathcal{L}' \subseteq \mathcal{L}$ having a cost lower than K that **maximize** the *Percent Data Transferred* [4] (PDT), i.e. the percentage of data acquired by the satellite that can be successfully send to the ground. This problem is denoted `MaxPDT`.

Regarding image acquisition, we assume that the time horizon is divided in a set $\mathcal{S}$ of successive acquisition slots and that a given amount of data a^s is acquired at the beginning of each slot s. The satellite has a buffer of size $B \geq 0$ filled with $B_0 \geq 0$ gigabits of data at the beginning of the time horizon.

Communications are possible when a station is reachable from the satellite (during visibility windows). We denote by $\mathcal{L}^s \subseteq \mathcal{L}$ the set of reachable stations during the slot s.

Characteristics of optical links during communications between satellites and ground stations are not well known and multiple parameters, mainly clouds, may influence the established link during a visibility windows, thus impacting the final data-rate. The computation of the real data-rate would be too complicated and beyond the scope of this paper, so we choose to simplify the problem as follows: for any visibility window (s, i) between the satellite and the station i during the slot s, we compute beforehand the amount of data q_s^s that can be downloaded using archived data from a cloud database, and we assume that this volume can be downloaded **instantaneously** at the beginning of the visibility window. Thus, each *visibility window* is reduced to a single time point associated with a download volume, which we call a ***download window***. We denote by $\mathcal{Q} = \{(s, i), s \in \mathcal{S}, i \in \mathcal{L}^s\}$ the set of all possible *download windows*.

We assume that the satellite cannot switch from one station to another in the middle of a visibility windows (two overlapping visibility windows cannot be both used). Thus, for two overlapping visibility windows, the two associated download windows are in mutual exclusion. We define $\mathcal{I} \subseteq 2^{\mathcal{Q}}$ the set of incompatible download windows.

2.2 Mixed Integer Linear Program

To model the `MaxPDT` problem, we first consider two types of binary variables for the selection of stations and download windows. For each station $i \in \mathcal{L}$, we define binary variables $y_i \in \{0,1\}$: $y_i = 1$ if and only if the station i is chosen ($\mathcal{L}' = \{i, y_i = 1\}$), and for each reachable station $i \in \mathcal{L}^s$, we define binary variables $x_i^s \in \{0,1\}$: $x_i^s = 1$ if and only if there is a download to station i during the acquisition slot s. Moreover, we introduce two real variables to model the amount of sent and lost data. For each slot $s \in \mathcal{S}$ we define $b^s \in \mathbb{R}^+$ the amount of data in the buffer *at the end of* s ($b^0 = B_0$ is the initial amount of data in the buffer) and $l^s \in \mathbb{R}^+$ the amount of data lost *during* s.

The objective is to *maximize* the *Percent of Data Transferred* or *minimize* the amount of data lost:

$$\max_{\mathcal{L}' \subseteq \mathcal{L}} PDT(\mathcal{L}') \quad \Leftrightarrow \quad \min_{\mathcal{L}' \subseteq \mathcal{L}} losses(\mathcal{L}') \quad \Leftrightarrow \quad \min_{\mathcal{L}' \subseteq \mathcal{L}} \sum_{s \in \mathcal{S}} l^s$$

$$\begin{aligned}
s.t. \quad & x_i^s \leq y_i, \quad s \in \mathcal{S}, \quad i \in \mathcal{L}^s && \text{(1a)}\\
& \sum_{(s,i) \in \mathcal{X}} x_i^s \leq 1, \quad \mathcal{X} \in \mathcal{I} && \text{(1b)}\\
& b^s + l^s = \max\left(0, b^{s-1} + a^s - \sum_{i \in \mathcal{L}^s} x_i^s q_i^s\right), \quad s \in \mathcal{S} && \text{(1c)}\\
& 0 \leq b^s \leq B - a^{s+1}, \quad s \in \mathcal{S} && \text{(1d)}\\
& b^S = B_0 && \text{(1e)}\\
& \sum_{i \in \mathcal{L}} w_i y_i \leq K && \text{(1f)}
\end{aligned}$$

Constraints (1a) and (1b) prevent downloads on stations that are not chosen ($y_i = 0$) and on forbidden set of locations. Constraints (1c) and (1d) force the amount of data at the end of a slot s to be consistent with the amount at the beginning of s and $s+1$ and to be less than the buffer size B minus the acquisition of slot $s+1$ (i.e. at the end of slot s, there must be at least a^{s+1} free space in the buffer). Constraint (1e) forces the final amount of data in the buffer to be the same as the initial buffer B_0. Constraint (1f) forces the total cost of the network to be less than the maximum cost allowed K.
Constraints (1c) can easily be linearized by introducing binary variables δ^s and constraints (1a) can be enhanced by forcing the model to use all non-overlapping download windows on chosen stations ($x_i^s = y_i$ if $\forall \mathcal{X} \in \mathcal{I}, (s,i) \notin \mathcal{X}$).

3 Hierarchical approach

3.1 Overview

In real instances, the number N of possible locations for the stations is often very small (some tens), but the temporal horizon is large (some years), and thus the number of x_i^s variables is orders of magnitude larger than the number of y_i variables. We propose to separate the decision process of these variables into two cooperative algorithms: A *Master* algorithm will exhaustively enumerate all possible networks of ground stations (decide y_i variables) and a *Slave* dynamic programming algorithm will evaluate the *PDT* of each of these networks by affecting values to the x_i^s variables.

Once a network has been found by the *Master* algorithm (y_i are decided), the remaining problem reduces to constraints (1b) to (1e) of the `MaxPDT` problem. In the following, we will details the *Slave* dynamic programming algorithm that we use to solve this reduced problem.

3.2 Label definition

The algorithm proceeds by extending a tree of labels which can be *active* or *visited*. A *label* $h = (b_h, l_h, \Omega_h, \mathcal{W}_h)$ is associated to each node in this tree. b_h is the current amount of data in the buffer, l_h the accumulated losses since the beginning, Ω_h the set of conflicting windows and $\mathcal{W}_h$ the list of download windows used.

The initial tree is made of a single *active* root node with a *label* $h_0 = (B_0, 0, \emptyset, \emptyset)$.

3.3 Label extension

The tree is extended in a breadth-first search manner: at each step, each *active* node having a non-dominated label is extended to a set of new *active* nodes and then marked *visited*, while any node with a dominated label is simply marked *visited* (no extension is made from dominated nodes).

Extension steps are made in a chronological order at the beginning of each acquisition slot and for each download window as follows:

When an acquisition slot is processed, each *active* node with a label h not dominated by another *active* node is extended to a new node having label h' with updated amount of data lost and in the buffer, $b_{h'} = min(B, b_h + a^s)$ (acquisition data are added to the buffer) and $l_{h'} = l_h + max(0, b_h + a^s - B)$ (if the buffer is not big enough, old data are lost). Conflicts and used download windows are not changed ($\Omega_{h'} = \Omega_h$ and $\mathcal{W}_{h'} = \mathcal{W}_h$).

When a download window $w = (s, i)$ is processed, each *active* node with label h is extended with one or two nodes with associated labels h_1 and h_2:

- If $w \in \Omega_h$ (the window is in conflict with the solution associated with h), the node is simply duplicated (one new child node is created with $h_1 = h$).
- Otherwise, two new nodes are created:

(i) One is a duplication of the current *active* node, i.e. it is node with label $h_1 = h$ which corresponds to a solutions where the window w is not chosen ($x_i^s = 0$).

(ii) The second one is an extension with label h_2 and corresponds to a solution where the window w is used ($x_i^s = 1$), thus:

· $\mathcal{W}_{h_2} = \mathcal{W}_h \cup \{w\}$ - The window is added to the set of used download windows;

· $b_{h_2} = max(0, b_h - q_i^s)$ - Data sent during the window are removed from the buffer;

· $\Omega_{h_2} = \Omega_h \cup \ \omega_w$ - Future windows in conflict with the window w are added to the set of conflicting windows of the solution.

After any extension step, the set of conflicting windows Ω associated with any *active* nodes is reduced by removing all windows in the past, i.e. windows that cannot be used in future solutions.

3.4 Dominance rules

From the above extension rules, we see that the number of nodes can double at each extension step. This exponential growth of the tree must be controlled by the use of dominance rules while ensuring optimality.

We say that a label h_2 is **dominated** by a label h_1 ($h_2 \prec h_1$) if and only if $\Omega_{h_1} = \Omega_{h_2}$, two labels can only be compared if they have the same set of conflicting windows, and:

$$b_{h_1} < b_{h_2} \wedge l_{h_1} \leq l_{h_2} \quad \mathit{or} \quad b_{h_1} = b_{h_2} \wedge l_{h_1} < l_{h_2} \tag{2a}$$

$$\mathit{or} \quad b_{h_1} = b_{h_2} \wedge l_{h_1} = l_{h_2} \wedge \mathcal{W}_{h_1} \prec \mathcal{W}_{h_2} \tag{2b}$$

Conditions (2a) compares labels according to the amount of data lost (current objective value) and the amount of data in the buffer. If there are less losses and less data remaining in the buffer in h_1 than in h_2, then the solution corresponding to h_1 is better than the one for h_2. Since $\Omega_{h_1} = \Omega_{h_2}$, any choice possible for extending h_2 is also possible for h_1 (the only constraints for extension come from conflicting windows), so if h_1 is better than h_2, there will be at least one solution made from extending h_1 that will be better than any solution created by extending h_2, thus h_1 dominates h_2.

Condition (2b) is only used to avoid having solution with the same objective value: two solutions may have the same amount of data lost and in the buffer, keeping both of them would be inefficient, so we remove the one with the *worst* set of used download windows ($\prec$ must be a **strict total order**).

This algorithm has a worst-case exponential complexity, but we will see that on real instances with few overlaps, the computation time is near linear. Moreover, the dominance rules guarantee that the dynamic programming algorithm provides optimal solutions, and combined with the exhaustive enumeration of the *Master* algorithm, we have the guarantee to find optimal solutions to the original `MaxPDT` problem.

4 Experiments and Results

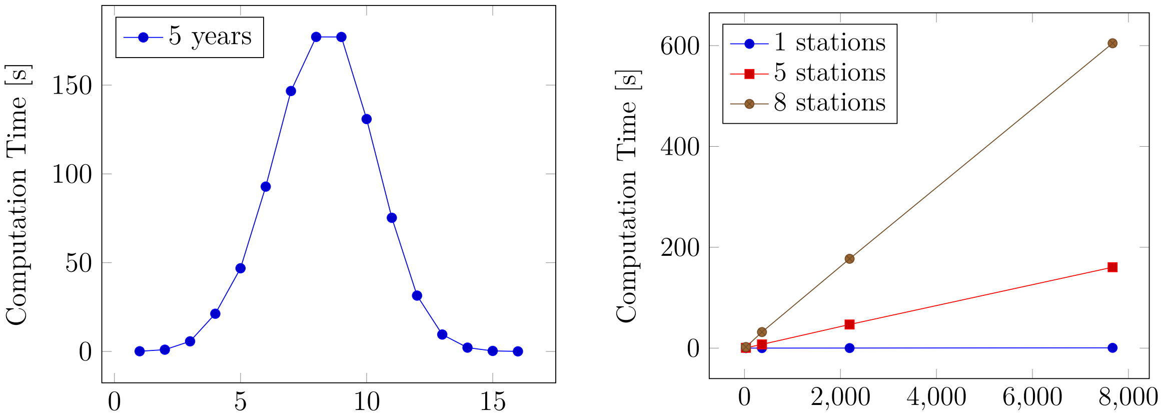
The algorithm was implemented in C++. All experiments where run on a 8-cores machine with 32GB of RAM running Linux (Ubuntu v14.04.4 LTS) and the proposed hierarchical algorithm was made parallel and was allowed to use the 8 cores of the machine. The MILP solver used was *CPLEX 12.6.3.*

We generated custom instances for a *low-earth orbiting* satellite using concepts of operations from [4], [3] and [5] (data-rate $\mathcal{D}_{\mathcal{R}} = 10.5$ Gbps, buffer size $B = 2300$ Gbits, an acquisition (slot) every hour and a constant acquisition volume of $a^s = 500$ Gbits). We used the *ERA Interim* cloud database (freely available) [1] to approximate the cloud cover $c_i^s \in [0, 1]$ during any visibility window (s, i). Visibility windows of the satellite were computed using the *Systems Tool Kit (AGI).* We assumed that the download volume for a download window (s, i), q_i^s, was proportional to $(1 - c_i^s)$ and to the duration of the visibility window d_i^s, i.e. $q_i^s = \mathcal{D}_{\mathcal{R}} * d_i^s * (1 - c_i^s)$. We discarded visibility windows that were too cloudy ($c_i^s > \gamma$) or too small ($q_i^s < \beta = 1$ Gbits). We used two different networks composed of 11 ($\mathcal{N}_{11}$) and 16 ($\mathcal{N}_{16}$) possible locations and since we could not find realistic information on the costs, we choose to simply select fixed numbers K of stations (between 1 and 16). We generated instances of various time horizons of one year, five years (partial instances) and 21 years (global instances).

Table 1 compares the MILP and the hierarchical approach on yearly, partial and global instances (with $K = 7$ and $\gamma = 1.0$), results are averaged for yearly and partial instances and *Gap* is the difference between the two methods. Results show that while the MILP solver manages to provide optimal solutions for the yearly instances, it only manages to provide non-optimal (*) solutions for some partial instances (optimal only for $\mathcal{N}_{11}$), and it cannot solve the more complex instances (global), while the hierarchical approach provides optimal solutions for all instances in reasonable time.

	$\mathcal{N}_{11}$				$\mathcal{N}_{16}$			
	MILP		**HA**		**MILP**		**HA**	
	CPU	*Gap*	*CPU*	*Gap*	*CPU*	*Gap*	*CPU*	*Gap*
Yearly	$33.7s$	0%	$0.949s$	0%	$2696s$	0%	$25.7s$	0%
Partial	$757s$	0%	$3.76s$	0%	$7200s$	0.117%	$86.9s$	0%
Global	$7200s$	N/A^*	$13.42s$	0%	$7200s$	N/A^*	$380.6s$	0%

Table 1
Comparison of CPU time and GAP between *MILP* and *HA*.



(a) Number of selected stations

(b) Number of days

Fig. 1. Hierarchical Approach performance on partial instances

Figure 1 shows computation times for the hierarchical approach on $\mathcal{N}_{16}$ for the four partial instances (with a time horizon of 5 years, results are averaged for all values of γ). Figure 1a displays an exponential behavior regarding the number of selected stations due to the exhaustive enumeration of the *Master* algorithm, while Figure 1b shows a near-linear behavior regarding the size of the time horizon. This linear growth may be explain by the fact that their are few overlaps and a lots of slots without any download windows in real instances, thus allowing the dominance rules to prune branches efficiently.

These results show that although the dynamic programming algorithm has a worst-case exponential complexity, the dominance rules perform very well on real instances.

5 Conclusion and perspectives

In this paper, we consider the `MaxPDT` optimization problem in the context of optical communications between a satellite and a network of ground stations. We model this problem using an aggregation of visibility windows. We propose a Mixed Integer Linear Program and an optimal hierarchical method using a complete enumeration of the subsets of locations and a dynamic programming (sub-)algorithm from the remaining problem. We show that while this dynamic-programming algorithm is theoretically exponential, its running time on real instances is nearly linear regarding the size of the temporal horizon of the instances and allow the hierarchical approach to outperform the MILP solver we tested. Solving such instances rapidly make it possible to analyze the impact of system parameters (data-rate, buffer size, network size) on spatial imagery systems using free-space optical communications. For our hierarchical approach, we use a simple exhaustive enumeration to find networks, which may not work on large instances with tens of stations. In the near future, we hope to solve instances with bigger networks and with more realistic costs by using custom (possibly approximate) algorithms for the enumeration of the networks. Furthermore, in the `MaxPDT` problem, we average the *Percent Data Transferred* over the whole horizon, which may result in solutions having months with very low *PDT*. In order to avoid such solutions, it might be interesting to look at a `MaxMinPDT` problem, in which the objective would be to maximize the minimum monthly *PDT*. A slightly modified version of our dynamic programming algorithm could be used for this purpose.

References

[1] Dee, D. P., S. M. Uppala, A. J. Simmons et al., *The era-interim reanalysis: configuration and performance of the data assimilation system*, Quarterly Journal of the Royal Meteorological Society **137** (2011), pp. 553–597.
URL http://dx.doi.org/10.1002/qj.828

[2] Eilertsten, B. and P. Hyvönen, *Ground station networks vs. geo relay satellites for polar orbiting satellites data download*, in: *Proceedings 12th International Conference on Space Operations (SpaceOps)* (2012).

[3] Guérin, A., G. Lesthievent and J.-L. Issler, *Evaluation of new technological concepts for high data rate payload telemetry*, in: *Proceedings 5th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TTC, ESA-ESTEC)*, 2010.

[4] Interagency Operations Advisory Group (IOAG), O., *Optical link study group final report*, Technical Report IOAG.T.OLSG.2012.V1, Interagency Operations Advisory Group (IOAG) Optical Link Study Group (OLSG) (2012).

[5] Lacoste, F., A. Guérin, A. Laurens et al., *Fso ground network optimization and analysis considering the influence of clouds*, in: *Proceedings 5th European Conference on Antennas and Propagation (EUCAP)* (2011), pp. 2746–2750.

[6] Le Kernec, A., M. Sotom, B. Bénazet et al., *Space evaluation of optical modulators for microwave photonic on-board applications*, in: *Proceedings 8th International Conference on Space Optics (ICSO)*, 2010.

[7] Link, R., M. E. Craddock and R. J. Alliss, *Mitigating the impact of clouds on optical communications*, in: *Aerospace Conference, 2005 IEEE*, 2005, pp. 1258–1265.

[8] Porte, H., A. Le Kernec, L. P. Quesada et al., *Optimization and evaluation in space conditions of multi-ghz optical modulators*, in: *Proceedings 10th International Conference on Space Optics (ICSO)*, 2014.

[9] Poulenard, S., M. Crosnier and A. Rissons, *Ground segment design for broadband geostationary satellite with optical feeder link*, in: *IEEE/OSA Journal of Optical Communications and Networking*, Journal of Optical Communications and Networking (JOCN) **7** (2015), pp. 325–336.

[10] Poulenard, S., M. Ruellan, B. Roy et al., *High altitude clouds impacts on the design of optical feeder link and optical ground station network for future broadband satellite services*, in: *Procceedings SPIE 8971, Free-Space Laser Communications and Atmospheric Propagation XXVI*, 2014.

[11] Roy, B., S. Poulenard, S. Dimitrov et al., *Optical feeder links for high throughput satellites*, in: *Proceedings IEEE International Conference on Space Optical Systems and Applications (ICSOS)* (2015), pp. 1–6.

[12] Takayama, Y., M. Toyoshima and N. Kura, *Estimation of accessible probability in a low earth orbit satellite to ground laser communications*, in: *Radioengineering, Proceedings of Czech and Slovak Technical Universities*, 2010, pp. 249–253.

[13] Wojcik, G. S., H. L. Szymczak, R. J. Alliss et al., *Deep-space to ground laser communications in a cloudy world*, in: *Proceedings SPIE 5892, Free-Space Laser Communications V*, 2005.