



HAL
open science

Earth observation systems optimization for free-space optical communications

Mikael Capelle

► **To cite this version:**

Mikael Capelle. Earth observation systems optimization for free-space optical communications. Other [cs.OH]. INSA de Toulouse, 2018. English. NNT : 2018ISAT0053 . tel-04213647

HAL Id: tel-04213647

<https://theses.hal.science/tel-04213647>

Submitted on 21 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Fédérale



Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :

l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le 11/09/2018 par :

MIKAËL CAPELLE

**EARTH OBSERVATION SYSTEMS OPTIMIZATION FOR
FREE-SPACE OPTICAL COMMUNICATIONS**

JURY

MARIE-JOSÉ HUGUET	Professeur des Universités	Directrice de thèse
NICOLAS JOZEFOWIEZ	Professeur des Universités	Directeur de thèse
SOPHIE DEMASSEY	Maître Assistante HDR	Rapporteur
OLIVIER PÉTON	Professeur	Rapporteur
JEAN-CHARLES BILLAUT	Professeur des Universités	Examineur
FRANÇOIS CLAUTIAUX	Professeur des Universités	Examineur
CÉDRIC PRALET	Ingénieur de Recherche HDR	Examineur
VINCENT MARTINOT	Ingénieur Thales Alenia Space	Invité

École doctorale et spécialité :

EDSYS : Informatique 4200018

Unité de Recherche :

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)

Directeur(s) de Thèse :

Marie-José HUGUET et Nicolas JOZEFOWIEZ

Rapporteurs :

Sophie DEMASSEY et Olivier PÉTON

Acknowledgments

First, I would like to thank my supervisors, Marie-José Huguet and Nicolas Jozefowicz, as well as Xavier Olive and Vincent Martinot from Thales Alenia Space, for giving me the opportunity to work on this fascinating research topic, and for their support throughout these years.

I am thankful to Sophie Demasse and Olivier Péton for the time they took to review this manuscript, and to the other members of my committee, François Clautiaux, Cédric Pralet and Jean-Charles Billaut.

I sincerely thank the members of the ROC team I had the pleasure to work with during this Ph.D and who have contributed, directly or indirectly, to its success. First, the Ph.D students, Idir (three years, we survived), Grégoire and Simon (for all our intense discussions regarding the C++ language), Estèle (I am eagerly waiting for the next presidential election), and also Clément, Jean-Thomas, Oliver, Pierre-Antoine, Ulrich and Yun. Second, the postdoctoral researchers, Azzedine, for all these conversations regarding the future of the geochaotical world, and Margaux, for all these visits of Toulouse by night. And last but not least, Christian, for making me discover wonderful artists and for sharing an office with me during three years — reintegration will be difficult —, Alain (you still owe me a beer), Aude, Christèle (ROC forever), Cyril, Denis (your prose will be missed), Emmanuel, Julien, Laurent, Pierre and Sandra.

I also thank the people of INSA I had the pleasure to work with during this Pd.D: Marie-José, again, for all the opportunities you gave me during these three years, Didier and Patrick, for all these practical Ada and Prolog sessions, and Pascal, for these few musical discussions.

I am also thankful to everyone I have worked with at Thales Alenia Space and who have helped me better understanding the topic I had to work with during this Ph.D. In particular, my sincere thanks to Agnès, Benjamin, Damien, François, Judith, (Marie)² and Sébastien.

I would then like to thank all the people that allowed me to relax when I had to, these three years alone would have been much more difficult.

First, my sincere thanks to Daniel and Jean-Patrick. It is always a pleasure to come and take lessons from you, and I am always eagerly waiting for our weekly sessions. I also thank Luc, Nicolas, Clément, Hervé and Stéphane, for these morning drum sessions would not have been the same without them.

Second, I would like to thank the people from Bisto De Nas and the Colomiers Variety Band. It is always a pleasure to play music with you. My sincere thanks to Didier, Jean-Christophe, Nadine and Philippe for welcoming me in these groups at the beginning of my Ph.D.

Last but certainly not least, huge and sincere thanks to Aurélie, Céline, Claire, Grégory, Philéas, Rémi and Sébastien for all the memories we shared during these three years, which are, I hope, only the beginnings of those to come.

Finally, I would like to thank my family for their continued and unwavering support, and for being there for my defense.

Remerciements

Je tiens tout d’abord à remercier ma directrice et mon directeur de thèse, Marie-José Huguet et Nicolas Jozefowicz, ainsi que Xavier Olive et Vincent Martinot de Thales Alenia Space, pour m’avoir donné l’opportunité de faire cette thèse avec eux et pour en avoir assuré son bon déroulement.

Je souhaite également remercier Sophie Demasse, Olivier Péton, François Clautiaux et Cédric Pralet pour avoir accepté d’examiner et de rapporter cette thèse, ainsi que Jean-Charles Billaut, pour m’avoir fait l’honneur de présider le jury lors de ma soutenance.

Je remercie sincèrement tous les membres de l’équipe ROC que j’ai côtoyé durant cette thèse et qui ont (in)-directement contribué à sa réussite : les doctorants, Idir (trois ans, on a survécu), Grégoire et Simon (nos discussions plus ou moins philosophiques sur le langage C++ me manqueront), Estèle (vivement les prochaines élections), ainsi que Clément, Jean-Thomas, Oliver, Pierre-Antoine, Ulrich et Yun ; les post-doctorants, Azzedine, pour nos conversations régulières sur l’avenir géo-chaotique du monde, et Margaux, pour toutes ces visites nocturnes de Toulouse ; et bien sûr les permanents, en particulier Christian, pour avoir largement enrichi ma culture musicale et pour avoir partagé un bureau avec moi pendant trois ans — la ré-insertion va être difficile — et également, Alain (tu me dois toujours une bière), Aude, Christèle (ROC un jour, ROC toujours), Cyril, Denis (ta prose me manquera), Emmanuel, Julien, Laurent, Pierre et Sandra.

Je remercie également les enseignants et personnels de l’INSA avec qui j’ai eu le plaisir de travailler durant cette thèse : Marie-José, à nouveau, pour toutes les opportunités que tu m’as offertes pendant ces trois ans, Didier et Patrick, pour tous ces travaux pratiques d’Ada et de Prolog, et Pascal, pour ces quelques séances musicales au GEI.

Merci également à toutes les personnes avec lesquelles j’ai eu la chance de discuter et travailler chez Thales Alenia Space, et qui ont fortement participé au bon déroulement de cette thèse, ainsi qu’à l’enrichissement de mes connaissances dans le domaine du spatial. En particulier, merci à Agnès, Benjamin, Damien, François, Judith, (Marie)² et Sébastien, vous côtoyer durant ces trois années a été un immense plaisir.

Je tiens ensuite à remercier toutes les personnes qui m’ont permis de décompresser lorsqu’il le fallait, parce que trois ans tout seul, cela aurait été long !

Tout d’abord, un immense merci à Daniel et Jean-Patrick. C’est toujours un grand plaisir de venir prendre des cours avec vous, et j’attends toujours nos séances hebdomadaires avec impatience. Merci également à Luc, Nicolas, Clément, Hervé et Stéphane, toutes ces matinées de batterie n’auraient pas eu la même saveur sans vous.

Merci également à tous les membres de la banda Bisto De Nas et du Variety Band de Colomiers, c’est toujours un grand plaisir de venir jouer et décompresser

avec vous. Merci Didier, Jean-Christophe, Nadine et Philippe pour m'avoir accueilli parmi vous au début de cette aventure.

Enfin, un immense et sincère merci à Aurélie, Céline, Claire, Grégory (il fallait bien que je te place quelque part!), Philéas, Rémi et Sébastien pour tous les moments que l'on a pu partagé durant ces trois années, et qui ne sont, je l'espère, qu'un avant-goût de ceux qui arrivent !

Enfin, merci à ma famille pour n'avoir jamais douté de moi en 25 ans et pour m'avoir fait le plaisir de venir assister à ma soutenance.

Contents

Introduction	1
1 Industrial context	3
1.1 Earth observation systems	4
1.1.1 Space segment	4
1.1.2 Ground segment	6
1.2 Free-space optical communications	9
1.2.1 Comparison with radio-frequency communications	10
1.2.2 Assumptions for this thesis	13
1.3 Taking clouds into account	15
1.3.1 Considered cloud databases	15
1.3.2 Interpolation of data	17
2 Review of optimization problems in the space industry	21
2.1 Optical ground station network optimization	22
2.1.1 Deep-space probe	22
2.1.2 Low-Earth orbiting satellites	23
2.1.3 Geostationary satellites	27
2.2 Scheduling of Earth observation systems	29
2.2.1 Planning image acquisitions for Earth observation systems	29
2.2.2 Planning mission telemetry for Earth observation systems	30
2.3 Optimization under uncertainties	31
2.3.1 Online optimization	31
2.3.2 Robust optimization	32
2.3.3 Stochastic optimization	33
3 Design of ground station networks for mission telemetry	37
3.1 Problem statement	38
3.1.1 Industrial context and assumptions	38
3.1.2 Maximizing the percentage of data transferred	41
3.1.3 The download points selection problem, $\text{MaxPDT}_{\mathcal{L}}$	44
3.1.4 Complexity results	46
3.2 Hierarchical approaches to solve the MaxPDT problem	48
3.2.1 Algorithms for the enumeration of networks of ground stations	48
3.2.2 Dynamic programming algorithm solve $\text{MaxPDT}_{\mathcal{L}}$	50
3.3 Computational results	55
3.3.1 Computational context	55
3.3.2 Results for the $\text{MaxPDT}_{\mathcal{L}}$ problem	56
3.3.3 Results for the MaxPDT problem	60

Contents

4 Flexible scheduling of acquisition downloads	71
4.1 Problem statement	72
4.1.1 Acquisition plans for Earth observation systems	72
4.1.2 Visibility windows between satellites and ground stations	73
4.1.3 Downlink data rates during visibility windows	74
4.2 Allocating download windows and assigning acquisitions	75
4.2.1 Computing the maximum download capacity	76
4.2.2 Assigning acquisitions to download windows	80
4.2.3 Discussion on the validity of the approach	82
4.3 On-board scheduling of acquisition downloads	82
4.3.1 Greedy algorithm to create download schedules	84
4.3.2 Insertion of download tasks in the schedule	85
4.3.3 Ordering acquisitions for download	87
4.4 Experiments	88
4.4.1 Generation of instances	88
4.4.2 Context and considered sets of experiments	93
4.4.3 Results and analysis	98
Conclusion	105
A Considered networks	109
A.1 List of ground locations	109
A.2 Map of ground locations	111
B Chosen locations in MaxPDT for the 20 year horizon	113
B.1 Chosen locations from \mathcal{N}_{11}	113
B.2 Chosen locations from \mathcal{N}_{16}	114
B.3 Chosen locations from \mathcal{N}_{48}	115
C Résumé en français	117
Bibliography	121

List of Figures

1.1	Ground track of a satellite on a 700-kilometers sun-synchronous orbit.	5
1.2	Visibility windows over five ground reception stations for a satellite on a 700-kilometers sun-synchronous orbit.	7
1.3	Number of accesses per repeat cycle and daily contact time for a low-Earth orbiting satellite on a sun-synchronous orbit.	8
1.4	Azimuth and elevation angles between a satellite and a ground station.	9
1.5	Beam divergence comparison between optical (FSO) and radio-frequency (RF) communications [Franz 2000].	11
1.6	Thickness of the atmospheric layer depending on the elevation angle between a satellite and a ground station.	13
1.7	Numbers of accesses per orbital cycle and daily contact time for a low-earth orbiting satellite on a sun-synchronous orbit for various minimum elevation e_{\min}	14
1.8	Cloud fraction retrieved from the ERA Interim database for January 2, 2004 at 12 P.M.	17
1.9	Example of interpolated cloud fraction from the ERA Interim database.	18
3.1	Example of instance for the optical ground station network optimization problem.	39
3.2	Computation time for instances of the <i>Adj</i> category using $DP_{pdt_{\mathcal{L}}}^+$ or the <i>MILP</i>	58
3.3	Computation time for instances of the <i>Int</i> category using $DP_{pdt_{\mathcal{L}}}^+$ or the <i>MILP</i>	59
3.4	Computation times for each methods using 8 threads, depending on the temporal horizon length.	63
3.5	Computation times for each methods using 8 threads, depending on the number of chosen stations.	64
3.6	Computation time for the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm using 1 thread for \mathcal{N}_{11}	65
3.7	Percentage of instances for which the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ did not find the optimal solution or failed to prove optimality for \mathcal{N}_{48}	67
3.8	Number of nodes evaluated by $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ using 8 threads, for each network, depending on the number of chosen stations.	68
3.9	Mean <i>Percent of Data Transferred</i> found for each network.	69
4.1	Visual representation of an acquisition plan with multiple visibility windows.	73
4.2	Example of visibility window allocation.	74
4.3	Examples of two data rate functions f_v	75
4.4	Example of a feasible solution for the <i>MaxCap</i> problem.	76

List of Figures

4.5	Examples of two volumes functions g_v	78
4.6	Example of volume functions g_v for a MaxCap instance.	78
4.7	Probability density function of the compression ratio with a 12 bits-per-pixel original encoding using a beta-distribution of parameters $\alpha = 2$ and $\beta = 3$	89
4.8	Number of acquisitions every ten minutes for the considered acquisition plans.	90
4.9	Data rate versus elevation for different percentage of cloud cover. . .	91
4.10	Visibility windows for the considered stations in the various scenarios. . .	92
4.11	Example of assignments of acquisitions depending on the chosen matching policy.	94
4.12	Examples of simple volume modifiers and weighted volume modifiers with various values of k_1 and k_2	95
4.13	Examples of volume modifiers ProfitDc [b , d] for various values of b and d for a single acquisition a	96
4.14	Earned profit and number of acquisition downloaded versus number of stations for various configurations.	99
4.15	Number of configurations for which each matching policy has found the best profit and comparison between the profit found.	101
4.16	Number of configurations for which sorting using the ratio between weight and volume or only volume find the “best” solution.	102
4.17	Total profit vs. mean acquisition age for various number of stations.	103
A.1	Map of the possible locations for ground stations.	111
B.1	Chosen locations for $K = 1, \dots, 11$ using \mathcal{N}_{11}	113
B.2	Chosen locations for $K = 1, \dots, 16$ using \mathcal{N}_{16}	114
B.3	Chosen locations for $K = 1, \dots, 48$ using \mathcal{N}_{48} (Part 1).	115
B.4	Chosen locations for $K = 1, \dots, 48$ using \mathcal{N}_{48} (Part 2).	116

List of Tables

3.1	Example of mapping from visibility windows to download points for the data shown in Figure 3.1.	40
3.2	Summary of notations for the definition of the MaxPDT and MaxPDT_{\mathcal{L}} problems.	45
3.3	Comparison of the number of instances solved between $DP_{pdt_{\mathcal{L}}}^+$ and the <i>MILP</i> solver for each category of instances.	57
3.4	Comparison of the number of instances of the <i>All</i> category solved between $DP_{pdt_{\mathcal{L}}}^+$ and the <i>MILP</i> for different probabilities of conflict.	60
3.5	Comparison of solving times between $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$, $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ and the <i>MILP</i> solver for instances with a temporal horizon of 5 years, on \mathcal{N}_{11} and \mathcal{N}_{16} when using one or eight threads.	66
4.1	Summary of notations for the definition of the MaxCap and GrdAsg problems.	83
4.2	An example of ordering of acquisitions for different type of priorities.	88
4.3	Acquisition scenarios for experiments.	90
4.4	Details of the considered visibility window scenarios.	92
A.1	List of the possible locations for ground stations.	110

Introduction

For many years, satellites have been used to monitor Earth from space for scientific or military purposes. With the democratization of the space industry, more and more commercial applications are emerging and these are reshaping space-based businesses into more user-oriented services. To provide users with valuable data, hundreds of satellites orbit around us taking a steadily increasing number of high-precision images. Despite being compressed on-board, these images generate a substantial volume of data that needs to be transferred to the Earth. To download such amount of data, new communication technologies are investigated to increase the quantity of information that can be sent from satellites to the Earth. In this context, free-space optical communications are seen as a key alternative to radio-frequency technologies which are currently facing a number of limitations.

Free-space optical communications transmit information using light, similarly to optical fiber used in computer networks, and can thus provide data rates orders of magnitudes higher than traditional radio-frequency communications. These technologies are already used in commercial applications to relay data through space from Sentinel low-Earth orbiting satellites, belonging to the Copernicus programme, to geostationary satellites of the European Data-Relay System (EDRS). Optical communications from space to ground have already been demonstrated in the past, but some problems remain in order to use them in industrial applications.

To transmit data using optical communications from a satellite to a ground reception station, light has to go through the atmosphere of the Earth. The impact of atmospheric turbulence on optical communications has already been studied a lot, and mitigation techniques that can stabilize the optical channel are already being developed. However, the effects of some elements of the Earth's atmosphere cannot be overcome by advances in technologies. In particular, most clouds completely block light, and in consequence optical communications. The energy required to transmit light through clouds would be excessive for any realistic systems, and thus new methods and procedures have to be considered to mitigate their impact.

This PhD was entirely funded by Thales Alenia Space France under a public-private convention (CIFRE) with the LAAS-CNRS laboratory.

The goal of this thesis is to investigate the use of free-space optical communications for mission telemetry in Earth observation systems, i.e., the download of images from satellites to ground reception stations.

We first consider the problem of designing efficient networks of optical ground stations. Clouds completely block optical communications, thus using a single ground station would not allow for regular downloads of data from satellites. Since on-board buffers have a limited capacity, the system would be underused or many images would be lost. To avoid such scenarios, the currently investigated approaches consist in exploiting networks of optical ground stations. These networks allow for

regular downloads of images, and with good spatial diversities, prevent the system from being saturated due to the presence of clouds.

We then consider the operational problem of scheduling downloads of images using free-space optical communications. In our context, images are compressed on-board and thus the sizes of their files are not known in advance. Furthermore, due to the uncertain nature of clouds, forecasts have to be used to detect their presence during communications between satellites and ground stations. Since these forecasts are not perfect, new reactive procedures have to be developed, shifting part of the decision process to the on-board system in order to reduce uncertainties when creating the download plan.

Manuscript overview

In Chapter 1, we present the industrial context considered for this thesis. We detail the organization of Earth observation systems and the advantages and drawbacks of free-space optical communications compared to radio-frequency ones. We also introduce the concepts required for the following chapters, together with the assumptions made regarding the technologies considered. We conclude this chapter by an overview of existing cloud databases and some details regarding the one we chose.

In Chapter 2, we review existing works regarding combinatorial optimization for the space industry. We focus on works related to the optimization of optical ground station networks and the scheduling of acquisition downloads from a low-Earth orbiting satellite, in an uncertain context. We conclude this chapter by a brief overview of techniques used to solve optimization problems involving uncertainties.

Chapter 3 is dedicated to the first problem considered in this thesis, the design of efficient optical ground station networks for Earth observation missions. We first present the assumptions made to solve this problem and a formal definition of the problem. We then present a hierarchical approach based on a dynamic programming algorithm, with an associated dominance rule tailored to real problem instances, to solve this problem. We conclude this chapter by presenting experimental results for our approaches. This work has been presented in two international conferences [Capelle 2018a, Capelle 2017] and submitted to the *Networks* journal [Capelle 2018b].

In Chapter 4, we present the download scheduling problem we consider in the context of free-space optical communications for Earth observation missions. We propose a two-stage flexible approach with a first ground optimization procedure using forecasts, followed by an on-board parameterizable algorithm. We complete this chapter by an evaluation of the performances of our method on realistic scenarios.

Finally, we conclude this thesis by summing up our contributions, and pointing out possible future works.

Industrial context

Contents

1.1	Earth observation systems	4
1.1.1	Space segment	4
1.1.1.1	Low-Earth orbiting satellites	4
1.1.1.2	Geostationary relay satellite	5
1.1.2	Ground segment	6
1.1.2.1	Mission center and data processing centers	6
1.1.2.2	Ground control stations	6
1.1.2.3	Ground reception stations	7
1.1.2.4	Considered networks	9
1.2	Free-space optical communications	9
1.2.1	Comparison with radio-frequency communications	10
1.2.1.1	Data rates	10
1.2.1.2	Power efficiency and security	10
1.2.1.3	Frequency licensing	11
1.2.1.4	Impact of atmospheric turbulence and clouds	12
1.2.2	Assumptions for this thesis	13
1.2.2.1	Geometrical constraints and considered data rates	13
1.2.2.2	Impact of clouds	14
1.3	Taking clouds into account	15
1.3.1	Considered cloud databases	15
1.3.1.1	Databases used in previous studies	16
1.3.1.2	Chosen database	16
1.3.1.3	Other databases	16
1.3.2	Interpolation of data	17
1.3.2.1	Spatial interpolation	18
1.3.2.2	Temporal interpolation	18

In this chapter, we present the industrial context considered in this thesis. We first introduce concepts related to Earth observation missions and describe the components relevant to our study in Section 1.1. We then present in Section 1.2 the main differences between free-space optical communications and radio-frequency

communications, in the context of Earth observation missions. We conclude this chapter by a discussion on the available cloud databases and the choice we made for this thesis in Section 1.3.

1.1 Earth observation systems

Nowadays, hundreds of satellites orbit around us to observe and monitor the Earth from space. These satellites collect scientific data and take images of the surface of the Earth on a daily basis, thus generating huge amount of data that must be sent to the ground to be processed.

These satellites are part of Earth observation systems that are made of a space segment and a ground segment. The ground segment includes a mission center and a network of ground control and reception stations, while the space segment consists in one or several satellites.

1.1.1 Space segment

The space segment of an Earth observation mission is made of one or multiple satellites. In this thesis, we only consider missions involving a single low-Earth orbiting (LEO) satellite, but the proposed approaches can be applied to other types of non-stationary satellites, such as medium-Earth orbiting (MEO) satellites.

1.1.1.1 Low-Earth orbiting satellites

Low-Earth orbiting (LEO) satellites orbits around the Earth at an altitude lower than 2000 kilometers. These satellites are not stationary and thus can observe or monitor multiple regions of the Earth depending on their orbits.

Figure 1.1 shows the ground track of a satellite on a 700-kilometers sun-synchronous orbit. The ground track of a satellite is the projection of its position on the Earth over time. A sun-synchronous orbit (SSO) is a near-polar orbit around the Earth in which satellites pass over any given point on the surface at the same local mean solar time. These orbits are useful for optical sensors since the illumination angle on the surface will be nearly the same each time the satellite takes an image. Due to its rotation around the Earth, a satellite is on the day-side of the Earth for only about half of its orbit. An orbit is said to be day-ascending (respectively night-ascending) if a satellite on this orbit goes from the south pole to the north pole on the day-side (respectively the night-side) of the Earth. Figure 1.1 represents about one-day of orbital data, and as we can see, the satellite is able to cover the whole Earth in this time but does not see each region the same number of times or for the same duration. Depending on its agility, the satellite is able to acquire images that are more or less distant from its ground track. Constraints regarding communications with ground reception stations are different and we will discuss them later in Section 1.1.2.3.

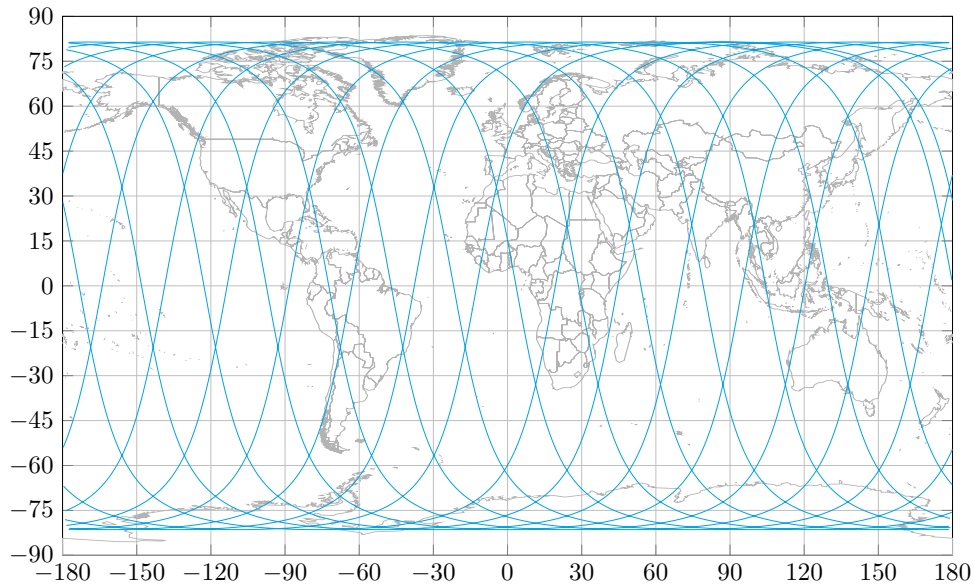


Figure 1.1 – Ground track of a satellite on a 700-kilometers sun-synchronous orbit.

The problems defined and the approaches proposed in Chapters 3 and Chapter 4 are not restricted to sun-synchronous orbits, i.e., no assumptions are made that would be invalid if the orbit was not sun-synchronous.

However, sun-synchronous orbits are interesting for experiments. Since the Earth is rotating, the ground track of the satellite is different on each revolution of the satellite around the Earth. However, sun-synchronous orbits are phased, meaning that after a given number of orbital periods, the satellite will get back on a previous trace. The time it takes for the satellite to get back on its trace is called the repeat cycle. Sun-synchronous repeat cycle are whole numbers of days, making them pretty interesting to generate scenarios over large horizon. Indeed, we can propagate the satellite trajectory over a single repeat cycle and then duplicate this trajectory as much as we want to fill the horizon. Furthermore, there exists a large variety of sun-synchronous orbits with various repeat cycles (from 2 to 40 days).

In this work, we consider a sun-synchronous satellite with an altitude around 700-kilometers and a repeat cycle of about 30 days. We did not use shorter repeat cycles because the shorter the repeat cycle, the wider the distance between revolutions of the satellites. With small repeat cycles, some regions of the Earth are not covered by the satellite, and ground reception stations located in these areas are never visible and thus cannot be used.

1.1.1.2 Geostationary relay satellite

Low-Earth orbiting satellites are only visible from specific regions of the Earth depending on the time of the day and cannot communicate with the ground segment

all the time. On the contrary, Geostationary satellites are positioned in a specific and relative small orbit high above the Earth (around 36000 kilometers). Due to their stationary nature and high altitude, they see, at any time, a large and fixed portion of the Earth. For instance, a geostationary satellite positioned around 0° longitude would see a large portion of Europe, northern Africa and middle-east at any time.

Furthermore, geostationary satellites can communicate with low-earth orbiting satellites during half their orbits, making them interesting to relay data from Earth observation satellites to the Earth.

We do not consider such scenarios in this thesis, but geostationary relay systems already exist, such as the European Data Relay System (EDRS) [Hauschildt 2014] or the NASA Tracking and Data Relay Satellites (TDRS) [Stampfl 1970].

1.1.2 Ground segment

We present here the various components of a ground segment for an Earth observation mission. In Section 1.1.2.3, we focus on the ground reception stations that are key components for the problems studied in this thesis.

1.1.2.1 Mission center and data processing centers

The mission center receives user requests for observations and produces acquisition and download plans to be sent to the satellite via a ground control station. Nowadays, download and acquisition plans are fully computed on the ground and low-level plans are uploaded to the satellite and executed without modification. In Chapter 3 we consider a flexible decision process where the acquisition plan is computed on the ground, but the download plan is computed on-board using information pre-processed on the ground.

The data processing centers receive raw images from satellites via ground reception stations and process them into valuable products for customers.

1.1.2.2 Ground control stations

Ground control stations receive plans from the mission center and upload them to satellites. Ground control stations also receive so-called health telemetries from satellites and upload acknowledgements of image receptions from data processing centers. Compared to mission telemetries, health telemetries are small and are used to ensure that satellites are working properly. In order to have a reactive system, ground control stations have to be located worldwide to allow frequent contacts between them and satellites. In this thesis, we do not consider the use of free-space optical communications for the uplink from ground control stations to satellites, or for the downloads of health telemetries. These are critical components of any space missions and must not be impacted by uncertainties inherent to optical communications.

1.1.2.3 Ground reception stations

Ground reception stations receive mission telemetries from satellites, which, for Earth observation missions, corresponds to images taken by the on-board instruments.

Ground reception stations are the most important components of the ground segment when it comes to downloading data from satellites. Indeed, their locations and their number have a direct impact on the amount of data that can be transmitted from satellites to the Earth.

While orbiting around the Earth, satellites can communicate intermittently with the various ground reception stations during **visibility windows**. The number and the duration of visibility windows depend mainly on the position of the stations on the Earth. For instance, for a sun-synchronous satellite, stations located at high latitude will have more visibility windows than stations located around the equator.

Visibility windows between the satellite and the ground reception stations can be computed by external tools prior to any analysis or optimization. In this thesis, we use the *Systems Tool Kit* (STK) software to compute these visibility windows. We considered a J2 perturbation¹ for the propagation of the satellite trajectory when computing these visibility windows.

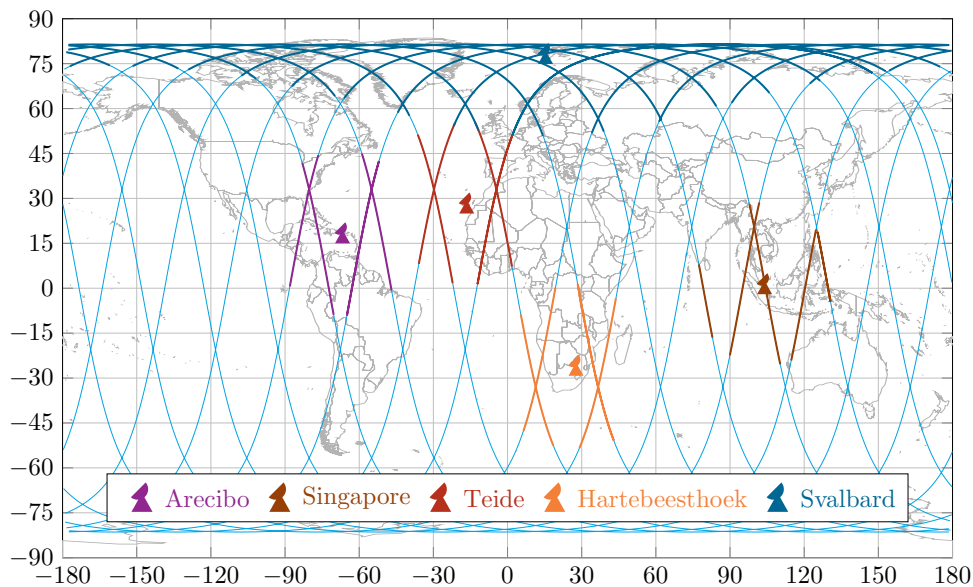


Figure 1.2 – Visibility windows over five ground reception stations for a satellite on a 700-kilometers sun-synchronous orbit.

Figure 1.2 shows visibility windows for five ground stations located at Arcibo

¹When propagating a satellite orbit, it is necessary to account for perturbations due to non-ideal environments, such as the non-spherical nature of the Earth.

(18.34°N), Singapore (1.35°N), Teide (28.30°N), Hartebeesthoek (25.89°S) and Svalbard (78.23°N) for a satellite on a 700-kilometers sun-synchronous orbit. The visibility windows are shown for a single day and are projected on the ground track of the satellite. We can see that high-latitude stations have much more visibility windows per day than mid-latitude stations. This is expected since sun-synchronous orbits are near-polar orbits, i.e., a satellite on a sun-synchronous orbit will pass near the pole during each orbit.

Figure 1.3a shows the number of visibility windows per repeat cycle depending on the latitude of ground stations for a satellite on a 700-kilometers sun-synchronous orbit with a 25 days repeat cycle. High-latitude stations can have up to 10 visibility windows per day on average while stations around the equator only have between 2 and 4 visibility windows per day. Figure 1.3b shows the average daily contact times between a satellite and ground stations of various latitude. As expected, stations with high latitude have higher daily contact times, almost 3 hours for some stations, while mid-latitude stations have less than one hour of daily contact time. Figure 1.3a–1.3b were generated by computing access for the ground locations considered in this thesis (see Appendix A.1), which is why the points are not evenly distributed.

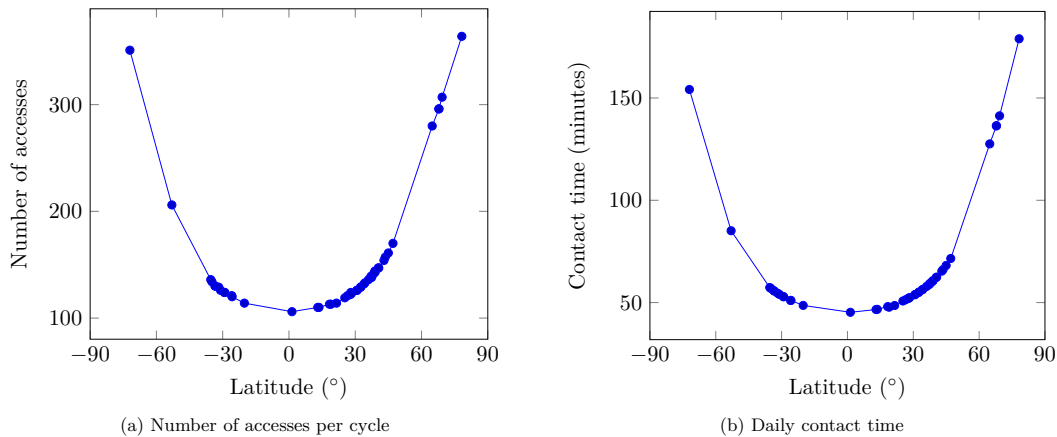


Figure 1.3 – Number of accesses per repeat cycle and daily contact time for a low-Earth orbiting satellite on a sun-synchronous orbit.

Furthermore, during visibility windows, satellites do not always see the ground reception stations from the same angle. Figure 1.4 shows a visual representation of the elevation angle between a satellite and a ground station.

This elevation angle goes from 0° when the satellite is on the horizon, to 90° when the satellite is directly above the ground station. During a visibility window, it is not common for the satellite to pass right above a ground reception station, which is why this angle is often less than 90° . The STK software used to compute visibility windows between a given satellite and pre-defined locations can also provide information regarding the elevation between the satellite and the ground stations during visibility windows.

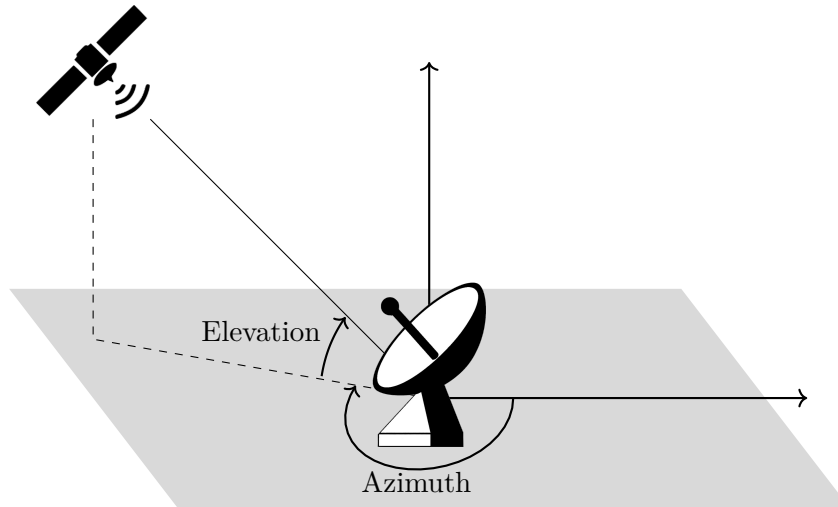


Figure 1.4 – Azimuth and elevation angles between a satellite and a ground station.

In Figure 1.2–1.3, the minimum required elevation is 0° , i.e., we assume that the satellite can communicate as soon as it is above the horizon. In practice, this is not feasible, and considering higher minimum requirements on elevation has a direct impact on the number of accesses and on the daily contact times, as discussed in the next section.

1.1.2.4 Considered networks

In this thesis, we consider multiple sets of locations for ground reception stations. These sets have been created arbitrarily using existing locations of stations or observatories. We consider three sets denoted \mathcal{N}_{11} , \mathcal{N}_{16} and \mathcal{N}_{48} containing respectively 11, 16 and 48 locations. The set \mathcal{N}_{11} consists mostly of European stations, and the set \mathcal{N}_{16} is the one considered in [IOAG 2012]. The set \mathcal{N}_{48} has been created to evaluate the performance of our algorithms and is made of locations from \mathcal{N}_{11} and \mathcal{N}_{16} , and locations selected from the STK database. A list of the considered locations is given Appendix A.1 together with their positions on a world map in Appendix A.2.

1.2 Free-space optical communications

Free-Space Optical (FSO) communications transmit data through space using light in the near-infrared spectrum.

Optical communications between low-Earth orbiting and geostationary relay satellites are already operational. The European Data Relay System is already equipped with a 1.8 gigabits-per-second Laser Communication Terminal (LCT) [Zech 2015]. However, optical communications from space to ground are still an experimental subject. The main difference between LEO-relay and LEO-Earth optical communication links is that optical links between low-Earth orbiting

and geostationary relay satellites do not go through the atmosphere, and are thus not impacted by atmospheric turbulence, clouds or aerosols.

This thesis focuses on the optical downlink from satellites to ground reception stations, and do not consider optical uplink used for telecommands from ground control stations. Such links have already been demonstrated at data rates up to 5.625 gigabits-per-second [Gregory 2013]. To our knowledge, no operational system currently uses space-to-ground optical communications.

The two main challenges in free-space optical communications through the atmosphere are atmospheric turbulence and the presence of blocking elements, such as aerosols or clouds. The impact of atmospheric turbulence on the communication channel is out-of-scope for this thesis.

In the next section, we compare free-space optical communications with current radio-frequency communications. We then present the assumptions we made for this thesis in order to cope with the lack of knowledge regarding free-space optical communications.

1.2.1 Comparison with radio-frequency communications

Current low-Earth orbiting satellites mainly transmit data to the Earth using radio-frequency communications. The three main frequency bands currently used to transmit mission telemetries are the S-Band (2–4 gigahertz), the X-Band (8–12 gigahertz) and the Ka-Band (26.5–40 gigahertz). The W-Band (75–110 gigahertz) is also considered for future communication systems.

1.2.1.1 Data rates

Much like current terrestrial optical-fiber networks, free-space optical communications may offer data rates orders of magnitude higher than current radio-frequency technologies [Guérin 2010b].

S-Band communications are limited to some hundreds of megabits-per-second, while X-Band may reach up to two gigabits-per-second in the future. Ka-Band already provides such data rates, and could reach higher data rates, while W-Band would provide even higher data rates but are highly experimental.

Targeted data rates for free-space optical communications go up to one terabits-per-second for feeder link in geostationary communication satellites. Considered data rates for space-to-ground communication links from low-Earth orbiting satellites are about 10 gigabits-per-second. More experimental technologies such as Wavelength-Division Multiplexing (WDM) could increase this value to 40 or 50 gigabits-per-second.

1.2.1.2 Power efficiency and security

Free-space optical communications use optical channels to transmit data to the Earth. The currently considered wavelengths for these communications are 1064 and 1550 nanometers, which are in the infrared spectrum. In comparison, the smallest

1.2. Free-space optical communications

wavelength in the Ka-Band (at 40 gigahertz) is 7.5 millimeters. Furthermore, typical radio-frequency antennas for low-Earth orbiting satellites have an aperture of about one meter, while telescopes for free-space optical communications on such satellites have an aperture of less than 10 centimeters.

The beam divergence of any light or radio wave in space is proportional to $2.44 * \frac{\lambda}{D}$, where λ is the wavelength and D the aperture diameter. This means that free-space optical communications have much smaller beam divergence than radio-frequency ones. Figure 1.5 shows the impact of a smaller beam divergence when transmitting data from Mars to the Earth (when Mars is as close as possible to the Earth).

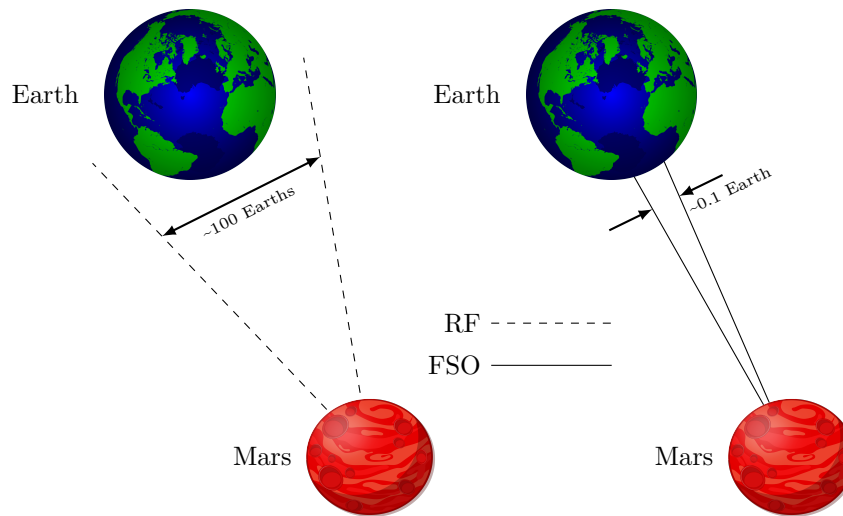


Figure 1.5 – Beam divergence comparison between optical (FSO) and radio-frequency (RF) communications [Franz 2000].

The tiny beam divergence of free-space optical communications has multiple advantages. For a given power intensity on the emitting antenna, the received power is much higher for free-space optical communications. It is much harder, if not impossible, to eavesdrop on free-space optical communications. In fact, when transmitting from a low-Earth orbiting satellite, the region from which data can be intercepted on the ground is less than 200 meters wide, meaning that it is quite easy to control it. Unlike radio-frequency waves, optical waves do not go through walls, so it is impossible to eavesdrop from hidden locations.

Finally, due to their very directed beam and very small on-board telescope, free-space optical terminals are much smaller and require less energy than traditional radio-frequency ones [Kaushal 2016].

1.2.1.3 Frequency licensing

Interference between radio-frequency communications is a major issue nowadays. With more and more satellites orbiting around the Earth, the frequency spectrum

is becoming congested. To cope with this congestion, the frequency bands are divided and allocated to satellites, organizations or services by the mean of licensing. This licensing is done by regulatory authorities such as the International Telecommunication Union (ITU). Frequency spectrum licensing is not free and takes time, and thus has a direct impact on the development cost and time of any space system. Free-space optical communications currently do not require frequency licensing, and may never require it due to the tiny divergence beams of telescopes compared to radio-frequency antennas.

1.2.1.4 Impact of atmospheric turbulence and clouds

Radio-frequency communications in the S-band and X-band are not impacted by atmospheric turbulence, clouds or particles in the atmosphere, while the ones in Ka-band and W-band are impacted by strong weather conditions such as heavy rains or storms.

The major drawback of free-space optical communications compared to radio-frequency communications is that they are strongly impacted by atmospheric turbulence and are totally blocked by clouds and aerosols in the atmosphere.

Atmospheric turbulence can be dealt with using various mitigation techniques such as smaller receiver apertures, adaptive optics or specific encoding [Sodnik 2012]. The mitigation of atmospheric turbulence is out-of-scope for this thesis which is why we assume that these are totally mitigated on the receiver side.

Aerosols are large particles found in the atmosphere such as volcanic dust, sand, or pollution. It is quite hard to obtain data regarding the presence of aerosols, especially consistent data over large temporal horizons. Furthermore, the impact of aerosols on optical communications as yet to be fully investigated in the literature, which is why we do not consider their impact in this thesis.

The most impacting components for free-space optical communications are clouds because the wavelengths used in optical communications cannot go through most clouds. According to recent studies, optical communications could go through some very high and thin clouds [Poulenard 2014], but we do not consider these in this study as they do not seem to have a huge impact on the global performance of the considered systems and we do not have access to a precise enough database.

In order to deal with clouds, a network of ground reception stations must be used with a good diversity in the locations of these stations to ensure regular communications from the satellite. Designing efficient networks is the purpose of Chapter 3.

Furthermore, during operational missions, it is necessary to take into account the uncertainties regarding the presence clouds when planning downloads of mission telemetries using free-space optical communications, as discussed in Chapter 4.

1.2.2 Assumptions for this thesis

In this thesis, we only consider the impact of clouds on the optical communication link. We assume that atmospheric turbulence is mitigated using one of the above mentioned techniques and we ignore the impact of aerosols on free-space optical communications.

1.2.2.1 Geometrical constraints and considered data rates

In Section 1.1.2.3, we defined the elevation angle between a satellite and a ground station.

When using current radio-frequency technologies, communication links can be established as soon as the satellite is 5° above the horizon. This is mainly done to avoid ground obstacles with relatively high height, such as mountains.

When dealing with optical communications, we have to take into account the thickness of the atmospheric layer that the light needs to go through since it increases the impact of atmospheric turbulence. As shown in Figure 1.6, with smaller elevation angles, the thickness of this atmospheric layer is higher than with elevation angles close to 90° (when the satellite is directly above the station).

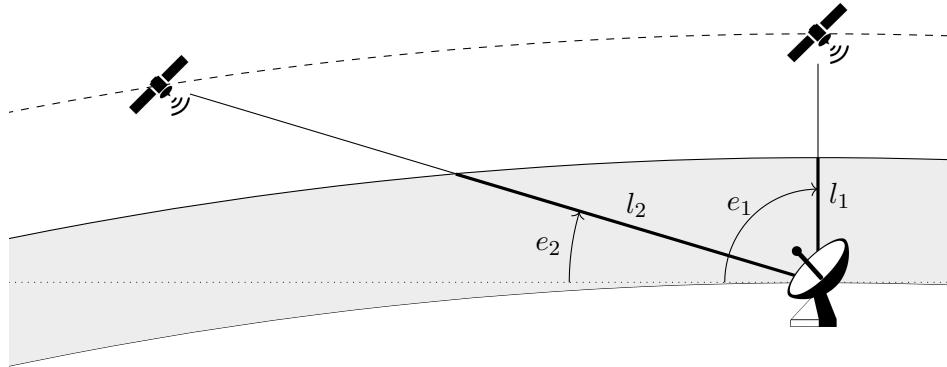


Figure 1.6 – Thickness of the atmospheric layer depending on the elevation angle between a satellite and a ground station.

With thicker layers, atmospheric turbulence is stronger, which directly impact the performance and feasibility of optical links.

To face this problem, one possibility is to establish optical links only when the satellite is above a given elevation. There is no strict consensus on the minimum elevation that should be considered for free-space communications, but most studies are using a 20° minimum elevation, which is the value considered in this thesis. Figure 1.7 shows the number of access and the average daily contact time for a low-Earth orbiting satellite on a 700-kilometers sun-synchronous orbit with a 25-days repeat cycle. As we can see, restricting communications to high elevation angles drastically reduces the number of visibility windows between the satellite and the ground stations, as well as the duration of these visibility windows.

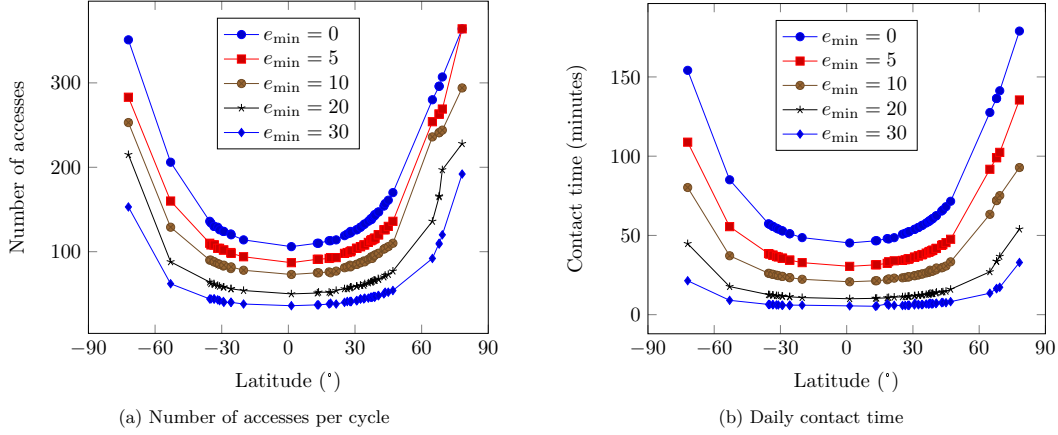


Figure 1.7 – Numbers of accesses per orbital cycle and daily contact time for a low-earth orbiting satellite on a sun-synchronous orbit for various minimum elevation e_{\min} .

Another possibility to cope with the variable thickness of the atmosphere is to use different data rates depending on the elevation angle between the satellite and the ground stations. This can be achieved by using variable modulation or coding depending on the thickness of the atmospheric layers, or, in a multiple channels environment, by reducing the number of used channels at low elevation angles to provide more power to individual channels [Kernec 2014].

1.2.2.2 Impact of clouds

When considering an optical communication link between a satellite and a ground station at time t , we assume that the data rate is proportional to the amount of clouds $c(t)$ over the station at time t :

$$\mathcal{D}_{\mathcal{R}}^c(t) = \mathcal{D}_{\mathcal{R}}(t) * (1 - c(t))$$

where $\mathcal{D}_{\mathcal{R}}(t)$ is the nominal data rate of the communications at time t . This approach was already considered in [Lacoste 2011].

In practice, clouds completely blocks optical communications, but we have to take into account the fact that the amount of clouds $c(t)$ over a station at a given time represents the fraction of the space covered by clouds. Cloud fractions are usually given over cells that are kilometers-wide, and thus we cannot compute exactly the impact of clouds on the optical link.

This formula assumes that clouds are homogeneously distributed over a station, and that it is possible to easily recover communications when they are blocked by clouds. Under this assumptions, the amount of data that can be transmitted to a

station during an interval $[a, b]$ is:

$$V(a, b) = \int_a^b \mathcal{D}_{\mathcal{R}}(t)(1 - c(t))dt$$

If we consider a constant data rate and if we assume that the interval $[a, b]$ is small enough (e.g., a few minutes) so that clouds do not change during the interval, we can reduce this to:

$$V(a, b) = (b - a) * \mathcal{D}_{\mathcal{R}} * (1 - c(a))$$

1.3 Taking clouds into account

In order to analyze and optimize systems for free-space optical communications, it is necessary to gather information regarding clouds.

When dealing with atmospheric data, different approaches can be considered. One is to use statistics regarding these data and either compute new statistics regarding the performance of the considered systems or do randomized simulations to estimate the outcome of these systems. Another possibility is to replay past data in order to characterize the systems.

When using statistics, one must consider the issue of temporal variations. Looking at clouds in particular, cloud statistics are likely different between January and July, but also between 0 A.M. and 12 A.M. Cloud data are available over the past 40 years only, so using a very high temporal precision would be complicated due to the few number of samples available. Furthermore, cloud covers usually do not follow standard probability distributions, and thus are difficult to manipulate.

Stochastic optimization using standard distributions is already very difficult, especially when considering complex systems, which is why we decided to use archived data in this thesis and optimize scenarios against them.

1.3.1 Considered cloud databases

The goal of this thesis is not to provide a cloud analysis, which is why we use existing databases to gather information about clouds.

In order to mitigate yearly variations when optimizing, it is necessary to retrieve information over multiple years, or even decades. Furthermore, in order to assess the performance of our operational procedures, it is interesting to replay scenario against past forecasts instead of generating them.

When searching for existing databases, we looked at the spatial and temporal availability — the database should provide worldwide data over multiple years — but also the spatial and temporal resolutions, and the availability of past forecasts. The spatial and temporal resolutions correspond to the precision in space and time of the cloud database.

Choosing a cloud database is not easy, and different databases may provide different outcomes for our analysis due to the somehow high discrepancy between

databases [Grishechkin 2012b, Stubenrauch 2013]. The selected database was chosen based on the criteria described before, but the approaches we propose can be used with any database.

1.3.1.1 Databases used in previous studies

In previous studies regarding optimization of networks for free-space optical communications, different databases were used. The two main ones were a “state-of-the-art” database [Link 2005, IOAG 2012], and the SAF-NWC facility [Guérin 2010a, Poulenard 2013].

While not explicitly mentioned, it is very likely that the “state-of-the-art” database was a cloud mask database (i.e., a database containing binary values indicating the presence of clouds) created from multiple geostationary weather satellites including multiple GOES, MSG and MTSAT satellites. This database is not available, which is why we do not use it in this thesis.

The SAF-NWC facility is a freely available software that creates cloud masks and cloud classifications using images from the SEVIRI imager on-board the MSG-1 satellite. The created datasets are very precise with a temporal resolution of 15 minutes and a spatial resolution of about 4 kilometers. Unfortunately, SEVIRI images are limited to Europe, Northern Africa and Middle-East due to the position MSG1. Furthermore, creating datasets over multiple years using the SAF-NWC facility requires a lot of pre-processing due to the fast production rate of SEVIRI, which why we do not use it.

1.3.1.2 Chosen database

In this thesis, we use the ERA Interim database [Dee 2011] because it is easy to access, free, and provides worldwide data with a relatively fine spatial resolution (up to $0.125^\circ \times 0.125^\circ$) from 1979 to today.

Furthermore, it provides past forecasts up to 48 hours which proved useful to simulate operational scenarios in Chapter 4.

The dataset provides cloud cover (i.e., fraction of clouds over a given cell) on a regular rectangular grid created using a specific projection and containing 1441×2880 points. On this grid, the spatial resolution corresponds to cells of about 14×14 kilometers at the equator (which are the largest cells on the grid).

Its major drawback is its temporal resolution, which is 6 hours, meaning that we need to interpolate data to obtain cloud covers at any given time.

Figure 1.8 shows an example of cloud cover data retrieved from the ERA Interim database for January the 3rd, 2004 at 12 P.M.

1.3.1.3 Other databases

Aside from the ERA Interim database and the databases used in previous works, there exists a variety of cloud datasets available. We cannot make an exhaustive

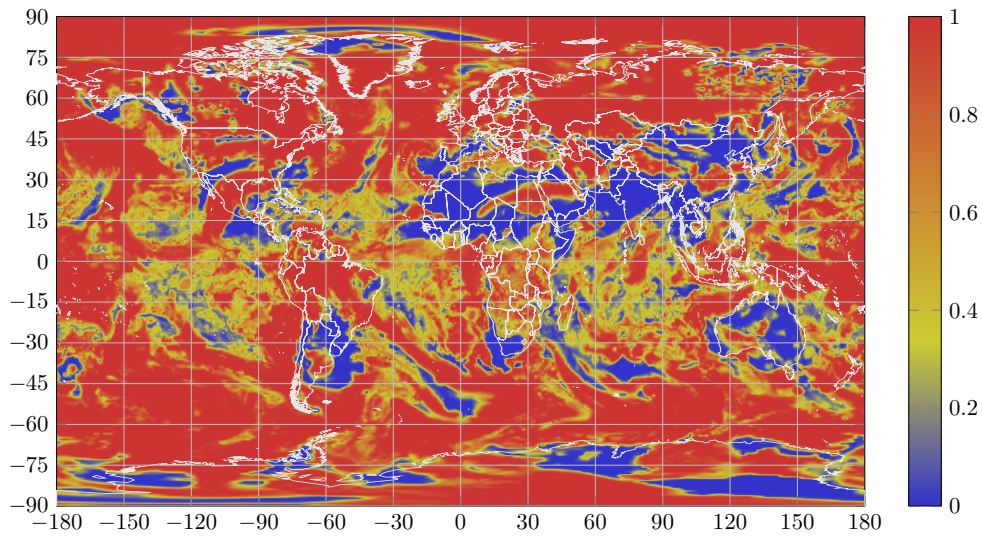


Figure 1.8 – Cloud fraction retrieved from the ERA Interim database for January 2, 2004 at 12 P.M.

list of available cloud databases, but we give here a small overview of the databases we found and that may prove useful for future or comparative analysis.

The ISCCP database [Rossow 1999] is a freely available cloud databases spanning more than 30 years. The database is constructed from mutiple satellites. Cloud fraction is provided worldwide every three hours with a spatial resolution of 280×280 kilometers. The database provides daily or monthly files, with statistical information regarding clouds. The CERES database [Wielicki 1996] is similar to the ERA Interim database. The database provides worldwide cloud cover every hour at a $1 \times 1^\circ$ spatial resolution. We discarded these two databases due to their bad spatial resolutions which did not fit our purpose.

The MERRA-2 database [Gelaro 2017] is a new database with a one hour temporal resolution and a $0.5 \times 0.625^\circ$ spatial resolution. This database would have been a good substitute for the ERA Interim database in our work but is only available since 2014 and in consequence was not found when the review of cloud databases was done for this thesis.

1.3.2 Interpolation of data

As mentioned earlier, the goal of this thesis is not to do a cloud analysis, and since interpolating cloud cover spatially or temporally is not an easy task, we make simplification.

1.3.2.1 Spatial interpolation

Due to the small spatial resolution of the considered database, we choose not to interpolate cloud cover spatially and instead simply consider the cloud fraction over the closest cell in the database. Given a location at coordinates (lat, lon) on the Earth, we choose the cell on the closest line (latitude) i and on the closest column (longitude) j in the database:

$$i = \underset{m}{\operatorname{argmin}} |lat - lat_m|$$

$$j = \underset{n}{\operatorname{argmin}} |lon - lon_n|$$

where lat_m and lon_n correspond respectively to the latitude of the m^{th} line and the n^{th} column in the ERA Interim grid.

1.3.2.2 Temporal interpolation

The ERA Interim database has a 6 hours temporal resolution, meaning that cloud covers are available at 12 A.M., 6 A.M., 12 P.M. and 6 P.M. every day of each year and for each cell. In order to obtain data between these times, we need to interpolate the cloud fraction from the database.

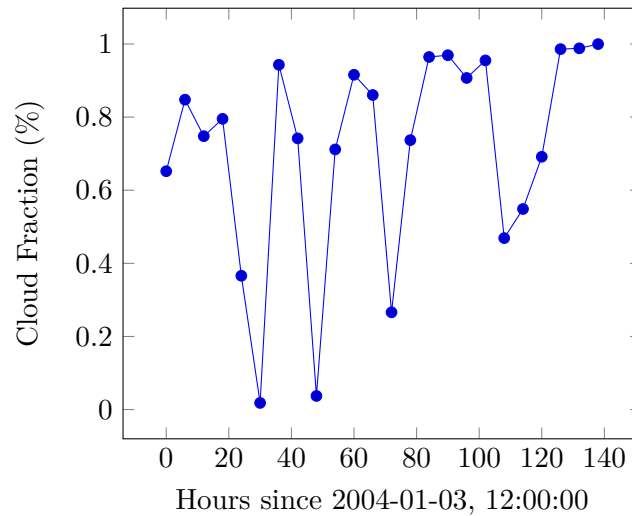


Figure 1.9 – Example of interpolated cloud fraction from the ERA Interim database.

One would be tempted to combine cloud information with other atmospheric data, such as winds, to obtain a better interpolation, but this does not guarantee better results [Grishchkin 2012a] which is why we choose to do a simple linear interpolation. This may not give accurate results for the selected database due to the 6 hours temporal resolution, but it is sufficient for our purpose. Given cloud fraction values $c(t_1)$ and $c(t_2)$ at times $t_1 < t_2$ from the database, we compute the

cloud fraction $c(t)$ at time $t \in [t_1, t_2]$ using:

$$c(t) = c(t_1) + \frac{c(t_2) - c(t_1)}{t_2 - t_1}$$

Figure 1.9 shows an example of temporal interpolation over 6 days from the ERA Interim database for the cell at (43.625, 1.5).

Summary

Due to the increasing volume of data acquired by Earth observation systems, new communications technologies are considered. With their high data rates, free-space optical communications are seen as a key technology to replace current radio-frequency communications. Unfortunately, optical communications are strongly impacted by clouds, and thus their impact have to be taken into account when designing and operating Earth observation systems.

In this thesis, we consider two problems related to the use of optical communications in Earth observation missions. The first one is the design of optical ground station networks, taking into account the impact of clouds on communications, and maximizing the volume of data transferred from a satellite to the Earth. The second one is the schedule of acquisition downloads from a satellite to ground reception stations, in order to maximize the profit from these acquisitions, and taking into account uncertainties regarding clouds.

To tackle these problems and validate the proposed approaches, we use archived cloud data from a worldwide database, over large temporal horizons, in order to deal with seasonal and yearly cloud variations.

Review of optimization problems in the space industry

Contents

2.1	Optical ground station network optimization	22
2.1.1	Deep-space probe	22
2.1.2	Low-Earth orbiting satellites	23
2.1.3	Geostationary satellites	27
2.2	Scheduling of Earth observation systems	29
2.2.1	Planning image acquisitions for Earth observation systems . .	29
2.2.2	Planning mission telemetry for Earth observation systems . .	30
2.3	Optimization under uncertainties	31
2.3.1	Online optimization	31
2.3.2	Robust optimization	32
2.3.3	Stochastic optimization	33

The space industry deals with a huge variety of combinatorial optimization problems for both designing and operating space systems in the context of telecommunication satellites, Earth observation missions, or exploration probes. The main goal is to obtain efficient systems while keeping development and operational costs as low as possible. These optimization procedures often involve complex and highly constrained systems, and have to take into account many uncertainties due to the non-constant and delayed feedback from these systems.

In telecommunication systems, frequency allocation is one of the most studied problems that arise during the development cycle of satellites. Variants of the problem have been proposed and solved using greedy algorithms [Kiatmanaroj 2012, Camino 2014, Couble 2017] or exact approaches based on mathematical programming [Camino 2016] or graph and scheduling theory [Kiatmanaroj 2016].

Other noticeable problems involve the minimization of the duration of validation plan for telecommunication satellites, for which constraint programming [Maillet 2011, Hebrard 2017] and local search [Boche-Sauvan 2014] approaches have been proposed. Local search algorithms are currently also used for the positioning of command devices in spacecraft [Roynette 2016].

With the rise of agile satellites, many studies have been conducted to plan the acquisitions of images for Earth observation missions such as SPOT-6 (2012),

SPOT-7 (2014) or Pleiades (2011) missions. Studies were also conducted to plan the downloads of these acquisitions or the downloads of scientific data from exploration missions, such as the MARS EXPRESS mission. These studies involved various combinatorial optimization methods, such as greedy heuristics, meta-heuristics, mixed-integer programming or constraint programming. In deep-space missions, various planning and scheduling problems have also been studied. For instance, constraint-programming techniques have been successfully applied to schedule experiments for the Rosetta/Philae mission in order to make the best use of the few available resources on-board [Simonin 2012].

This chapter presents existing works related to the design of optical ground station networks (OGSN) and the scheduling of Earth observation satellites. We conclude this state of the art chapter by a brief overview of existing optimization under uncertainties techniques.

2.1 Optical ground station network optimization

This section is dedicated to works related to the optimization of optical ground station networks. Most existing works mainly focus on the technical parts of the problem without exploring in depth the optimization aspects, or without detailing with it in the papers. However, these articles are interesting because they put forward the benefits of free-space optical communications, and provide information regarding assumptions made during the optimization processes, such as how the impact of clouds is taken into account, or from where clouds are retrieved.

While our work targets only low-Earth orbiting satellites, we present related work for various space systems.

2.1.1 Deep-space probe

Deep-space probes were one of the first space systems for which free-space optical communications were considered, and in turn, for which optimization of optical ground station networks was done.

Back in 2005, optimization of an optical ground station network (OGSN) in the United-States (continental US with Hawaii) for a deep-space probe was tried [Link 2005]. System constraints included a limited number of stations (5–6) within a limited area, with a quite high minimum 20° elevation between the space probe and the stations. To take clouds into account, multiple years of archived multi-spectral images with high temporal and spatial resolutions (15–30 minutes, 4×4 kilometers) were retrieved from the GOES¹ West and GOES East satellites. A custom algorithm was proposed to compute cloud-free information (cloud mask) for each pixel in these images. The objective of the optimization process was to find a

¹The Geostationary Operational Environmental Satellites (GOES) are a set of geostationary weather satellites operated by the United States' National Oceanic and Atmospheric Administration (NOAA).

2.1. Optical ground station network optimization

network with an availability above 90%, where the availability corresponded to the fraction of time that at least one station in the network had a cloud-free line of sight (CFLOS) from the space probe. The algorithm used was a two-step algorithm in which the first step consisted in exploring the solution spaces broadly without precisely evaluating each solution (about 40 million solutions), while the second step tried to converge to a local optima by evaluating networks more carefully (about 10–20 networks evaluated in this step).

This work was continued in [Wojcik 2005] to consider a worldwide network, including continental U.S., Hawaii, South America, Europe, Africa, Middle-East, central and southern Asia and Australia. Since GOES East and GOES West only cover America, other satellites were used to retrieve cloud information, such as GOES-9 (for Australia and China), Meteosat-7 (Europe and Africa) and Meteosat-5 (Middle-East and central Asia). Due to the use of new satellites, the temporal resolution increased to one hour, but the algorithm used to compute cloud mask stayed the same since these satellites have similar multi-spectral imagers. Additional system constraints were introduced to reduce the search space of the resolution process. In particular stations could only be located between -41° and $+41^\circ$ latitude, with a minimum altitude of 2 kilometers. With these constraints, a network of 6 stations distributed around the world was not sufficient to reach a 90% availability, but without the altitude constraint, a network of 5 stations was enough.

Some more recent works regarding the optimization of optical ground station networks for deep-space probes have been made in [Alliss 2012]. Constraints were similar to those of [Link 2005, Wojcik 2005], but the cloud database used was a “state of the art database” with a temporal resolution of 15 minutes created using images from multiple geostationary satellites: Geostationary Operational Environmental Satellites (GOES), Meteosat Second Generation (MSG), Multi-function Transport Satellite (MTSAT). Like in previous works, these satellites are all geostationary and thus no cloud data regarding high latitudes can be obtained from them. Unlike in previous works, the goal of the optimization process in this work was to reach a given percentage of data transferred (*PDT*) from the space probe to the Earth. Over the considered period (2005–2010), and considering a data rate of 260 megabits-per-second, a single optical ground station (Tenerife) was enough to provide *PDT* above 90%, while two sites were enough to reach a 99% *PDT*.

2.1.2 Low-Earth orbiting satellites

Low-Earth orbiting (LEO) satellites are space systems primarily used for Earth observations. We refer readers to Chapter 1 for details about their operational behavior. Between 2010 and 2012, multiple studies were conducted regarding the optimization of optical ground station networks for Earth observation missions. These studies considered direct optical link between satellites and optical ground stations, without the use of any relay.

A probabilistic approach was first considered by [Takayama 2010]. By analyzing images from a meteorological satellite (MTSAT) between June 2007 and May 2008

Chapter 2. Review of optimization problems in the space industry

using a custom algorithm, the probability of avoiding cloud blockage was computed for networks made of up to 8 stations. Stations were first positioned only in Japan, in large cities such as Tokyo, Sapporo or Fukuoka. When not considering orbital information, it was shown that networks of 1, 2 and 8 stations could respectively provide monthly probabilities of 39, 48 and 94% to avoid cloud blockage. Results also showed that for the considered networks (in Japan), strong seasonal variations could be observed, up to a 10% improvement in summer for a network of 8 stations.

In the same work, authors introduced the notion of *cumulative access probability*. The cumulative access probability is the probability of having at least one cloud-free visibility window between the satellite and one of the optical ground station after a given amount of time. Authors assumed that the probabilities p_i of the i^{th} visibility window to be cloud-free were all the same, for all stations in the network. They computed the cumulative access probability up to the n^{th} visibility windows, $p(n)$, using the following formula:

$$p(n) = 1 - \prod_{i=1}^n (1 - p_i) \quad (2.1)$$

It must be pointed out that this formula is only valid when the probabilities of cloud-free access are not correlated, which can be assumed for stations that are more than hundreds of kilometers apart [Fuchs 2015].

In [Takayama 2010], authors analyzed the cumulative access probability of a worldwide network made of 6 optical ground stations for the OICETS² low-Earth orbiting satellite. This network was constructed by considering the four stations used in a previous satellite demonstration involving OICETS, and adding two extra stations outside Japan to have a better distribution of visibility windows between the satellite and the ground stations. Experimental results, assuming constant and identical probabilities p_i of cloud-free accesses for all stations, showed that the cumulative access probability exceeded 99% after four hours for $p_i = 0.7$, five hours for $p_i = 0.5$ and eight hours for $p_i = 0.3$. With lower cloud-free probability such as $p_i = 0.1$, the cumulative access probability only reached 96% after 24 hours (the duration of the simulation). For these simulations, the minimum elevation allowed between the satellite and the optical ground stations was 0° , but visibility windows shorter than one minute were discarded.

High-level analysis for various existing Earth observation missions were conducted in [Guérin 2010b]. These missions included satellites from the Copernicus programme (ESA), Post-EPS and a non sun-synchronous satellite, SWOT. The goal of this work was to compare future telemetry payloads using X-Band, Ka-Band and free-space optical communications. Analysis were conducted assuming identical clear-sky probabilities for all high-latitude stations (55%) and all mid-latitude stations (65%). Minimum elevations for establishing communications of 5° for radio-frequency payloads and 20° for optical ones were considered. Results

²The Optical Inter-orbit Communications Engineering Test Satellite (OICETS) is an experimental satellite operated by the Japan Aerospace and Exploration Agency (JAXA).

2.1. Optical ground station network optimization

showed that, using a single mid-latitude station in Europe, it was possible to download 5.9 terabits of data per day using X-Band, 6.3 terabits using Ka-Band and 7 terabits using optical communications. For the considered high-latitude station (Svalbard), the daily volumes increased to 19.1 terabits for X-Band, 23 terabits for Ka-Band and 21.9 terabits for optical communications. When using more advanced optical technologies, such as wavelength-division multiplexing (WDM), the daily download volume using optical communications could reach 26.1 terabits for mid-latitude stations in Europe and 81.8 terabits high-latitude stations. These results showed the potential of free-space optical communications for space-to-ground optical links.

More recent studies [Guérin 2010a, Lacoste 2011, Grishechkin 2012b] used archived cloud data to analyze the impact of free-space optical communications for mission telemetry.

In [Guérin 2010a], a meteorological study realized by METEOFRAANCE (French National Meteorological Agency) from a CNES specification was used to optimize a network of ground stations in Europe and polar areas. Cloud information were computed using 3 years of data from the SEVIRI³ payload on-board the MSG-1 geostationary satellite and one year of data from the METOP low-Earth orbiting satellite. The possible locations for optical ground stations were the locations of the pixels in the SEVIRI and METOP images. Since these images contain thousands of pixels, a dedicated 3-step greedy algorithm was used to find efficient networks. First, stations whose clear sky probabilities were above the 95% percentile were selected in each 1° iso-latitude band. Then, the number of stations was reduced to 30 by coupling cloud information with orbital characteristics and constraints specific to the targeted missions. Finally, 6 stations were chosen according to monthly and yearly clear sky statistics. From these 6 stations, multiple networks of two, three and four stations were created and their availability was analyzed. In this work, the availability of a network was the percentage of days where at least one of the four daily slots (from the SEVIRI and METOP images) had a cloud cover lower than 25%. Results showed that, considering only stations in Europe, the availability of networks could reach 95.9% availability with two stations, and 99.5% with four stations. When adding a polar station (Svalbard) to the best two-sites network, the availability increased by 1.5% over the year, and even by 7% for some months during winter. During this high-level analysis, the time between cloud-free accesses was analyzed. This information is important because most of the time, images must be downloaded quickly to be valuable, and satellites have a finite on-board buffer, and thus need frequent downloads to avoid overflow (i.e., loss of images). For the best network of three stations, the duration between cloud-free visibility windows was shorter than six hours 90.4% of the time but longer than 30 hours 3% of the time. When adding a fourth station, these percentages improved to 93% and 1.7% respectively. In both cases, the longest duration between cloud-free accesses was

³The Spinning Enhanced Visible and Infrared Imager (SEVIRI) is an optical imaging radiometer on-board Meteosat Second Generation (MSG-1) satellite.

Chapter 2. Review of optimization problems in the space industry

the same (66 hours). Since the temporal resolution of the cloud database considered was 6 hours, a database with a higher temporal resolution should be used in order to obtain finer analysis.

Work made in [Guérin 2010a] was extended in [Lacoste 2011]. While more detailed, the optimization process used in [Lacoste 2011] was similar to the previous work, and 10 stations were selected according to it. Since most sites were in Europe (except Svalbard), a clear seasonal behavior was observed with clear-sky percentages higher than 90% in August but lower than 70% for some months. Networks were analyzed in more detail compared to [Guérin 2010a] by coupling cloud information with orbital characteristics from Earth observation missions considered in [Guérin 2010b].

The characteristic analyzed for each network was the total visibility duration, i.e., the total time where the satellite was able to communicate with a ground station, taking clouds into account. This characteristic was computed for a single station i using the following formula:

$$T_i = \sum_{k=0}^N v_i(k) * (1 - c(k)) * T_s$$

where T_i was the total visibility duration taking clouds into account for station i , $c(k) \in [0, 1]$ was the cloud cover above station i during step k , T_s was the sampling time (10 seconds in this work), N was the number of steps in the simulation (3153601 in this work, corresponding to one year) and $v_i(k) \in \{0, 1\}$ was the geometric visibility of station i , i.e., $v_i(k) = 1$ if the satellite was visible from station i and 20° (minimum elevation) above the horizon during step k . Since the temporal resolution of the cloud dataset was much higher than the sampling time, linear interpolation was used to compute $c(k)$.

Results showed that over the three years of analyzed data, the cumulative distribution of daily contact time was quite stable, and taking clouds into account reduced this time by a factor of about 4.5. Comparison were made between a single ground station using X-Band (1 gigabits-per-second, minimum elevation of 5°) and a network of optical ground stations (10.5 and 42 gigabits-per-second with respectively one or four channels). With a single ground station located in Svalbard, X-Band daily volumes ranged from 8.3 to 9.5 terabits, which cannot be reach by any optical ground station with a single channel. Networks made of two to four optical ground stations were then analyzed, and all but one of these networks provided higher daily volumes than the single Svalbard station using X-Band 99% of the time. The only network that did not reach these performances was a France-only network. This showed that ground station diversity is required in order to achieve good performances with free-space optical communications.

In 2010, the Inter-agency Operations Advisory Group (IOAG) established the Optical Link Study Group (OLSG) to investigate business cases for future space systems using optical communications. This group published a final report [IOAG 2012, Alliss 2012, Schulz 2012] in 2012. Multiple scenarios were an-

2.1. Optical ground station network optimization

alyzed in this report, including L2 (second Lagrange point), Lunar, LEO, GEO (Earth relay) and deep-space. The cloud database used was said to be a “state-of-the-art cloud database” made using collected data from GOES, MTSAT and MSG geostationary satellites, combined with MODIS and METOP data for polar areas. The spatial and temporal resolutions of the database were respectively 4×4 kilometers and 15 minutes, spanning a total of 6 years (2006 to 2011). Most scenarios were oversized in term of acquisition volumes according to current observation missions in order to prove the benefits of free-space optical communications. For instance, the low-Earth orbit scenario had a daily acquisition volume 10 times higher than current low-Earth orbiting space systems. These report showed that it was possible to transfer 90% of the data in all scenarios, and almost reach 95% for the LEO scenario using seven stations (from the 16 available). Information regarding the costs of creating optical ground station networks was also included in the final report, showing that using direct space-to-ground free-space optical communications was much cheaper than using geostationary relays.

In [Grishechkin 2012b], a network made of 4 stations in Germany was designed to maximize the visibility duration between the satellite and the optical ground stations. Neither the optimization method nor the cloud database were specified. Constraints imposed on the system were a minimum elevation of 40° and a cloud fraction lower than 30% to have a cloud-free visibility window. Starting with an existing 4-stations network, the optimization process increased the overall visibility duration over the network by 10%.

Recently, [del Portillo 2017] proposed a simulation-based approach to find locations for optical ground stations in order to serve a low-Earth orbiting satellite. Visibility windows between the satellite and the ground stations were computed between 2005 and 2015, and a custom probabilistic model was implemented to simulate cloud covers above the ground locations over the considered horizon. Two genetic algorithms were proposed to solve two different versions of the problem: in the first one, stations could only be chosen from a finite set of 68 locations, in the second one, stations could be located anywhere inside allowed countries. The goal of the problem was to optimize both the cost of the network and the network availability, i.e., the percentage of time where at least one of the station is visible from the satellite and cloud-free, and the latency between cloud-free contacts.

2.1.3 Geostationary satellites

The following references are dedicated to the optimization of optical ground station networks for telecommunications using geostationary satellites. The goal here is not to download as much data as possible but rather to maximize the availability of networks, i.e., the percentage of time where at least K stations of the network are cloud free. In this context, chosen stations are visible all the time since the satellite is on a geostationary orbit. While none of the cited works mention it, this problem is actually the well-known maximum coverage problem, which is known to be NP-Complete.

Chapter 2. Review of optimization problems in the space industry

Works on geostationary satellites are more recent than the previously described ones, thus database are often much more precise. Most of the works done by [Poulenard 2013, Poulenard 2014, Poulenard 2015, Fuchs 2015] targeted only Europe and thus used the SAF-NWC cloud classification with a temporal resolution of 15 minutes and a spatial resolution of 4×4 kilometers.

In [Poulenard 2013], a greedy algorithm was used to select a given number of stations in Europe. This greedy algorithm started with no station, and at each stage, selected the station with the most cloud-free time slots within time slots that were not already cloud-free for the currently selected stations. It can be noticed that this algorithm is well-known for the maximum coverage problem and is essentially the best possible polynomial-time approximation algorithm for this problem, with an approximation ratio of $1 - \frac{1}{e} \approx 0.63212$. Despite the polynomial running time of the greedy algorithm, the computation took multiple hours for only six stations due to the slow data accesses from the cloud database.

In [Poulenard 2014], authors developed the same methodology to optimize networks in Europe and extended Europe (including northern Africa and Middle-East) but with a different approach regarding the impact of clouds on the optical links. By comparing the cloud classification from the SAF-NWC facility with data from the CALIOP Lidar on-board the CALIPSO⁴ satellite. they showed that it was possible to transmit data using free-space optical communications through high semitransparent thin clouds (as defined by SAF-NWC). It was shown that, in order to reach a 99.9% availability, 10 to 11 stations were required in Europe (even when allowing high semitransparent thin clouds) but four stations were sufficient when including Africa and Middle-East. This work was extended in [Poulenard 2015] by optimizing the data network between the optical ground stations using another greedy algorithm.

In [Fuchs 2015, Giggenbach 2015], authors considered a statistical approach to find optical ground station networks in Germany, Europe and worldwide in order to maximize the availability of multiple stations. Ground locations were pre-selected according to infrastructure or political constraints yielding a total of 66 possible stations (22 in Germany, 27 in Europe and 17 in northern Africa or Middle-East). Then, three optimization methods were tried using 5 years of cloud data from the same dataset as [Poulenard 2013, Poulenard 2014]. The first one was a greedy approach which successively selected the sites with best availability, without taking cloud correlation into account. The second one enumerated all possible networks to evaluate their performance. The third one tried to reduce the number of considered sites and networks based on pre-computed cloud statistics. Experiments showed that the first method yielded poor results while the second failed to terminate when the number of chosen stations was greater than four. Regarding the third proposed algorithm, selecting 25 stations and evaluating only 10^5 networks proved to be

⁴The Cloud-Aerosol Lidar with Orthogonal Polarization (CALIOP) is a payload on-board the Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) satellite operated by NASA and CNES. It provides information regarding the attenuation of light due to clouds and aerosols in the atmosphere.

enough to obtain very good performances. Results showed that choosing stations only in Germany was not effective, while a network of 10–12 stations in Europe could provide a 99.9% availability. In comparison, networks including northern Africa and Middle-East could provide up to 99.999% availability with only eight stations.

Finally, [Chen 2015] analyzed the link availability of a 77°E geostationary satellite for networks made of one, two or three stations located in Asia. Networks were chosen using yearly cloud covers computed from satellite images taken between October 2013 and September 2014. Results showed that networks of one, two or three station were able to achieve respectively 74.73%, 93.7% and 97.13% availability.

2.2 Scheduling of Earth observation systems

2.2.1 Planning image acquisitions for Earth observation systems

One of the oldest combinatorial optimization problems in mission planning for Earth observation systems is the scheduling of acquisitions to be made by a satellite. Given a list of user requests, the goal of mission planning is to schedule acquisitions in order to optimize some criteria (total profit, fairness between users), taking into account mechanical constraints of the satellite such as the time required to make an acquisition, to switch from one position to another or to turn on and off components of the satellite.

Various approaches have been considered to tackle this problem. Constraint satisfaction models were proposed for the SPOT-5 mission paired with a (dynamic) Branch & Bound in [Agnèse 1995, Bensana 1996]. Multiple meta-heuristics were considered to solve the problem [Globus 2003, Globus 2004], including genetic algorithms, simulated annealing, squeaky wheel optimization and stochastic hill climbing. A tabu search heuristic was also proposed to select requests in order to maximize profit by [Cordeau 2005].

Over time, multiple variants of the problems have been considered and solved using different approaches such as dynamic planning [Pemberton 2002], mathematical programming and graph theory [Gabrel 2003, Benoist 2004, Augenstein 2014].

Multi-objective variant of the problem considering, e.g., the maximization of the profit and the fairness between users were also considered and solved using genetic algorithms [Tangpattanakul 2013] and local searches [Tangpattanakul 2015].

The initial problem was quickly extended to deal with constellation of multiple satellites. Greedy algorithms, local search heuristics and column generation approaches [Bianchessi 2007, Cordeau 2005, Cordeau 2005] were used to find solutions and upper bounds for the scheduling of observations of a constellation of multiple satellites. Greedy algorithms and local search algorithms were recently considered for the planning of acquisitions for a global ocean surveillance mission involving multiple satellites [Pralet 2011, Verfaillie 2012].

Lots of Earth observation satellites take pictures in the visible spectrum, and thus cannot acquire images when clouds are present. Uncertainties regarding clouds

have been considered using mathematical approaches [Bensana 1999], stochastic integer programming [Liao 2005, Liao 2007] and chance constraints programming [Wang 2016]. More recently, flexible approaches [Maillard 2015b] were considered to take into account uncertainties regarding the energy consumption of the satellite.

2.2.2 Planning mission telemetry for Earth observation systems

Over time, the volume of images acquired by Earth observation systems has increased, and the telemetry payloads, used to send the images to the Earth, have become more and more complex. This created the needs for dedicated procedures and algorithms for the scheduling of image downloads.

These scheduling problems first arose in exploration mission, such as the MARS EXPRESS mission, for which various meta-heuristics based on local and randomized searches built around a CSP model were proposed [Oddi 2003]. Constructive algorithms [Bianchessi 2008] and mathematical model [Pralet 2012] were then proposed for the scheduling and allocation of downlinks for constellation of satellites.

More recently, on-board scheduling approaches were proposed and studied [Pralet 2008]. Within these approaches, some decisions are made on-board, using limited resources but with more up-to-date information regarding available resources, making such approaches interesting compared standard ground-only optimization methods [Verfaillie 2011].

A problem similar to the one we consider in Chapter 4 was investigated in [Pralet 2014, Maillard 2014, Maillard 2015a] for radio-frequency communications. The problem consisted in downloading acquisitions from a low-Earth orbiting satellite, under specific system constraints. Acquisitions were made of files that were stored in separate memory banks (unsharable resources) with uncertain volumes due to on-board compression. Each file had to be downloaded using one of the available channels, and the downloads could not be interrupted. Other system constraints included on-board encryption and orbital constraints. Considered objectives included maximization of the number of downloaded acquisitions in each level of priority or minimization of the mean age of acquisitions (gap between the acquisition and its download).

The above defined problem was first introduced in [Pralet 2014] and was broken down into three successive problems: assigning acquisitions to download windows, scheduling downloads of files by assigning channels to files and ordering downloads on channels and memory banks, and finally assigning temporal values to the download tasks. On-board algorithms were proposed for the three defined problems to remove the uncertainties regarding the sizes of the files. To solve the assignment problem, two greedy algorithms were proposed. The first one assigned acquisitions based on their priorities and weights, while the second one assigned acquisitions using the regret of not downloading a given acquisition during a given download window. Three methods were then proposed to solve the scheduling problem. The two first one were greedy algorithms that would insert downloads in an (initially

empty) plan at the first available positions, while the third one was a local search algorithm used only if the two previous methods had failed. The temporal problem was solved using a custom library, InCELL, which propagates temporal constraints and evaluates optimization criterion.

The fully on-board procedure defined in [Pralet 2014] was compared with a only ground approach (with margin due to uncertainty) and with a shared approach between the ground and the on-board systems in [Maillard 2014, Maillard 2015a]. Multiple methods were compared, with various scenarios, but algorithms were not detailed in these works. These papers used flexible approaches, i.e., ground procedures generating plans which can easily be modified on-board, and showed that delegating some or all decisions to the satellite increased the overall system performance by a large margin.

2.3 Optimization under uncertainties

Within most realistic problems, there are some uncertainties regarding the input parameters such as travel times in routing problems, resource consumption or availability, or task duration in scheduling problems. These uncertainties can be categorized depending on the amount of information available. The most common categories includes probability distributions, discrete or continuous scenarios or no information at all. We present here a simple non-exhaustive overview of frameworks that are used to tackle such problems including online optimization, robust optimization and stochastic optimization.

2.3.1 Online optimization

Online optimization considers problems where information regarding some parameters of the input are not available to the decision process at the beginning of the considered horizon. Such problems arise in scheduling, when release dates and processing times of tasks are not known in advance, in networking, when no information is available about future connection requests, in paging management for operating systems, and in many other fields [Albers 2003].

In online optimization, the time available to take a decision when new information is revealed is often very small, thus heuristics and greedy algorithms are often chosen to solve these problems. The performances of these heuristics or algorithms are usually evaluated regarding their competitive factors in the worst case. The competitive factor in the worst case for an algorithm A corresponds to the worst ratio between the objective found by the algorithm A and the real optimum (if parameters were known in advance).

Formally, the competitive factor in the worst case, $\rho(A)$, can be formulated as follows:

$$\rho(A) = \max_{I \in \mathcal{I}} \frac{f_A(I)}{f^*(I)}$$

where \mathcal{I} is the set of possible instances of the problem, $f_A(I)$ is the objective function

computed by algorithm A on instance I and $f^*(I)$ is the real optimum value of instance I . In this regard, online algorithms are closely related to approximation algorithms.

Online versions of standard problems have been studied, such as the well-known knapsack [Han 2009, Böckenhauer 2014], m -machines scheduling problems [Graham 1966, Albers 1999] or transportation problems [Grötschel 2001].

2.3.2 Robust optimization

In lot of problems involving uncertainties, one expects to have some guarantees regarding the solutions found: solutions must be feasible in all possible scenarios, or the objective value should be as good as possible, whatever the scenario. Robust optimization [Gabrel 2014] is a framework used to solve such problems. Often seen as worst-case optimization, the goal of robust optimization is to find solutions that are robust, i.e., that do not suffer too much from variations in the input parameters. Formally, a robust maximization problem is usually defined as follows:

$$x^* = \operatorname{argmin}_{x \in \bigcap_{s \in S} X^s} \max_{s \in S} f(x, s)$$

where S is the set of possible scenarios (the uncertainty set), X^s the feasible domain in scenario s and $f(x, s)$ the objective function in scenario s .

Initially, uncertainty sets were very conservative, i.e., considering cases where all input parameters took their worst possible values [Soyster 1973]. Eventually, more complex uncertainty sets were considered in order to obtain more realistic solutions [Ben-Tal 2000], but making the robust version of simple problem much harder to solve. More recently, the concepts of *budget of uncertainty* [Bertsimas 2004] and *bw-robustness* [Roy 2010] were introduced. The budget of uncertainty, often denoted Γ , corresponds to the number of input parameters that can vary in any scenarios, e.g., the number of arcs that can take non-nominal values in shortest-path problems. The *bw-robustness* considers robust optimization problems where the fitness of a solution is related to the number of scenarios for which this solution has an objective value smaller than a given value b , or zero if there are scenarios for which this solution has objective values greater than another value w . The idea is to penalize solutions with an objective value above b and forbid solutions with objective values higher than w .

Robust versions of standard deterministic problems are often NP-Hard, even if their deterministic counterparts can be solved in polynomial times. For instance, the robust shortest-path problem (where arc lengths are bounded between two values) is strongly NP-Hard for most classes of graph, and even weakly NP-Hard for very simple classes of graph [Yu 1998]. However, for a given *budget of uncertainty*, the Γ -robust shortest-path problem is polynomial even if the *bw-robust* version is still NP-Hard [Gabrel 2013].

In some robust optimization problems, a given set of decisions can be postponed, and two-stage approaches can be used. Various two-stage problems were studied,

such as robust network flow and design problems [Atamtürk 2007] or robust location transportation problems [Gabrel 2010] where demands of customer are not known in advance.

2.3.3 Stochastic optimization

Stochastic programming encompass problems where uncertainties regarding inputs parameters can be represented by probability distributions [Birge 2011]. Stochastic programs are often very complex to solve, even for very simple distributions [Coffman Jr 1980].

Multiple approaches exist to tackle stochastic optimization problems: maximization of the expected value [Dean 2004], penalization of the objective function weighted by probabilities of scenarios [Yu 2000], or even Monte-Carlo simulation approaches [Morton 1998]. We present here two generic approaches: two-stage optimization and chance constraint programming.

Two-stage stochastic optimization [Schultz 1996] can be used for problems where parts of the decision process can be postponed, and is similar to two-stage robust optimization. The idea is to set some decision variables, in order to maximize the expected value of a second optimization problem. A typical formulation can be written as follows:

$$\min_{x \in \mathcal{X}} f(x) + \mathbb{E}[Q(x, \xi)] \quad \text{with} \quad Q(x, \xi) = \min_{y \in Y(x, \xi)} q(x, y, \xi)$$

where ξ are random variables, \mathcal{X} is the deterministic feasible domain for variables x , $Y(x, \xi)$ is the feasible domain for variables y given decisions on x and values for ξ . Such problems can be very hard to solve depending on the shape of Y and q and the distribution of ξ .

In chance-constraint programming (CCP), probabilistic constraints are introduced in deterministic models. The goal is to optimize some objective function while ensuring that some constraints are valid with given probabilities. A generic CCP program with individual chance constraints can be formulated as follows:

$$\begin{aligned} \min. \quad & f(x) \\ \text{s.t.} \quad & x \in X \\ & \mathbb{P}(x \in \hat{X}_j) \geq \beta, \quad j = 1, \dots, m \end{aligned}$$

where \hat{X}_j are non-deterministic domains for the variable x .

In this program, probability constraints are disjoint, thus each one is enforced individually, meaning that the probability of all constraints being valid simultaneously is much smaller than β .

A stronger version for the same problem would involve joint chance constraints that look at the probability of all (or multiple) constraints being valid simultane-

ously:

$$\begin{aligned} \min. \quad & f(x) \\ \text{s.t.} \quad & x \in X \\ & \mathbb{P}(x \in \bigcap_{j=1}^m \hat{X}_j) \geq \beta \end{aligned}$$

Chance-constraint programs are often very hard to solve, even when the non-deterministic domains are linear combinations of simple random variables. However, in some cases, these domains can be made deterministic, e.g., when only the right-hand side of a constraint is non-deterministic:

$$\begin{aligned} & \mathbb{P}(ax \leq \hat{b}) \geq \beta \\ \Leftrightarrow & 1 - F_{\hat{b}}(ax) \geq \beta \\ \Leftrightarrow & F_{\hat{b}}^{-1}(1 - \beta) \geq ax \end{aligned}$$

where $F_{\hat{b}}$ is the cumulative distribution function (CDF) of \hat{b} , and $F_{\hat{b}}^{-1}$ its inverse. This transformation can only be applied on individual chance constraints, and is restricted to specific combinations of random variables.

Linear programs with joint chance constraints are often much harder to transform, which is why specific tools have been developed to characterize the feasible sets of such programs [Beraldi 2002a, Beraldi 2002b].

Summary

The design of optical ground station networks for data download from low-Earth orbiting satellite has been studied a lot in the past years. To our knowledge, no work has been done from a combinatorial optimization point of view, and most existing works focus on hand-made procedures for the creation of networks. Modeling cloud uncertainties as distribution probabilities is not so easy, as explained in Chapter 1, and the distributions are often non-standard, which is why standard stochastic approaches defined in Section 2.3.3 cannot be easily applied. Furthermore, in order to take account for seasonal variations, it would be necessary to optimize ground station networks at least over one year, which would create large instances that cannot be handle by common stochastic approaches. In order to cope with these issues, we choose to optimize optical ground station networks over past data using the cloud database ERA Interim presented in Section 1.3.

The problem of scheduling acquisition downloads from low-Earth orbiting satellites has already been studied a lot in the past years. Uncertainties regarding the volume of acquisitions has already been taken into account using scheduling procedures shared between the ground control stations and the satellites. To our knowledge, free-space optical communications have not been considered in the past

2.3. Optimization under uncertainties

in this context. The main issue with scheduling for free-space optical communications is the impact of clouds on the optical link. In this thesis we combine both the uncertainties regarding the volumes of acquisitions and the uncertainties regarding the impact of clouds on the downlink. One-stage stochastic optimization is out-of-question here since when a plan is uploaded to a satellite, the plan must be feasible in all scenarios. One-stage robust procedures would likely be too conservative for such problems, especially when considering on-board procedures that can modify the plan quite a lot. In order to make good use of the system resources, two-stage procedures should be considered, with some decisions postponed to an on-board algorithm. In this context, most decisions can be taken or modified on-board, and thus considering approaches where part of the decisions are set in the first stage and cannot be modified in the second stage does not seem interesting. Furthermore, the uncertainties regarding acquisition volumes and clouds above ground stations are not well characterized, which is why we choose to consider a flexible procedure, where a huge part of the decision process is made on-board.

Design of ground station networks for mission telemetry

Contents

3.1	Problem statement	38
3.1.1	Industrial context and assumptions	38
3.1.2	Maximizing the percentage of data transferred	41
3.1.2.1	Definition of the MaxPDT problem	41
3.1.2.2	Mathematical model	43
3.1.3	The download points selection problem, MaxPDT $_{\mathcal{L}}$	44
3.1.3.1	Definition of the MaxPDT $_{\mathcal{L}}$ problem	44
3.1.3.2	Mathematical model	46
3.1.4	Complexity results	46
3.1.4.1	Complexity of the MaxPDT $_{\mathcal{L}}$ and MaxPDT problems	46
3.1.4.2	Special complexity cases for MaxPDT $_{\mathcal{L}}$	47
3.2	Hierarchical approaches to solve the MaxPDT problem	48
3.2.1	Algorithms for the enumeration of networks of ground stations	48
3.2.1.1	Exhaustive enumeration	49
3.2.1.2	Branch and bound algorithm	49
3.2.2	Dynamic programming algorithm solve MaxPDT $_{\mathcal{L}}$	50
3.2.2.1	Definition of the $DP_{pdt_{\mathcal{L}}}$ algorithm	50
3.2.2.2	Dominance rule to prune labels, $DP_{pdt_{\mathcal{L}}}^+$	51
3.2.2.3	Complexity of $DP_{pdt_{\mathcal{L}}}^+$ for special classes of instances	55
3.3	Computational results	55
3.3.1	Computational context	55
3.3.2	Results for the MaxPDT $_{\mathcal{L}}$ problem	56
3.3.2.1	Parameters for the MaxPDT $_{\mathcal{L}}$ experiments	56
3.3.2.2	Overall performances of the considered approaches	57
3.3.2.3	Performances on each category of instances	57
3.3.3	Results for the MaxPDT problem	60
3.3.3.1	Parameters for the MaxPDT experiments	61
3.3.3.2	Comparison between the considered approaches	62
3.3.3.3	Impact of multithreading on the performances	65
3.3.3.4	Analysis of the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ results	66

In this chapter, we consider the problem of finding a network of ground stations maximizing the percentage of data acquired by a low-Earth orbiting satellite that can be transferred to the Earth using free-space optical communications. Given a high-level acquisition mission and a set of possible locations for the ground stations, our goal is to find an efficient network with a limited budget (cost of opening stations, number of stations).

The chapter is organized as follows. In Section 3.1, we formally define **MaxPDT**, the problem of finding an optical ground station networks maximizing the percentage of data transferred (**PDT**), and one of its sub-problem, **MaxPDT_L**, and provides some results regarding their complexities. Then, we propose in Section 3.2 a dynamic programming algorithm to solve **MaxPDT_L** and two exact hierarchical approaches combining this algorithm with different enumeration methods to solve the **MaxPDT** problem. Computational results regarding **MaxPDT** and **MaxPDT_L** are then presented in Section 3.3.

These results have been presented in [Capelle 2018a, Capelle 2017] and submitted in [Capelle 2018b].

3.1 Problem statement

In this section, we explain the assumptions considered to define the Maximum Percent Data Transferred (**MaxPDT**) problem. Then, we formally define the **MaxPDT** problem and one of its sub-problem, **MaxPDT_L**, that we will later use to solve **MaxPDT** and derive its complexity.

3.1.1 Industrial context and assumptions

Considering a low-Earth orbiting satellite and a given set $\mathcal{R} = \{r_1, \dots, r_N\}$ of possible locations for optical ground stations with associated costs p_r ($r \in \mathcal{R}$), we aim to find a subset $\mathcal{R}^* \subseteq \mathcal{R}$ having a total cost lower than a given value K that maximizes the percentage of data transferred (**PDT**), i.e., the percentage of data acquired by the satellite that can be successfully transferred to the Earth. We assume that each location $r \in \mathcal{R}$ can only host a single station.

We assume that the satellite has a buffer of limited size $B \geq 0$ that is empty at the beginning of the time horizon and must be empty at the end. We consider that the time horizon $\mathcal{H} = [T_{start}, T_{end}]$ is divided into a set $\mathcal{S} = \{s_1, \dots, s_M\}$ of successive acquisition slots, and that a given amount of data, i.e., an acquisition volume, $a^s > 0$, is acquired at the beginning t^s of each slot $s \in \mathcal{S}$. There is no gap between two successive acquisition slots, thus the end time of slot s_i is the beginning time $t^{s_{i+1}}$ of slot s_{i+1} . By definition we have $t^{s_1} = T_{start}$ and for simplicity we assume $t^{s_{M+1}} = T_{end}$.

While orbiting around the Earth, the satellite is able to reach intermittently the various locations (of optical ground stations) during visibility windows (see

3.1. Problem statement

Section 1.1.2). We denote by \mathcal{V} the set of all visibility windows. Each visibility window $v \in \mathcal{V}$ is associated with a start time $t_v^{sta} \geq T_{start}$, an end time $t_v^{end} < T_{end}$, a unique location (or station) $\tau_v \in \mathcal{R}$, a data rate function $d_v : [t_v^{sta}, t_v^{end}] \rightarrow \mathbb{R}^+$ and a set γ_v of overlapping visibility windows:

$$\gamma_v = \left\{ v' \in \mathcal{V} : t_v^{sta} \leq t_{v'}^{sta} < t_v^{end} \quad \text{or} \quad t_v^{sta} \leq t_{v'}^{end} < t_v^{end} \right\}$$

Two overlapping visibility windows cannot be both used simultaneously for downloading data and are thus in mutual exclusion.

Example 1. Let us consider the example in Figure 3.1 containing four slots, three stations, and eight visibility windows (denoted from A to G). The acquisition volume corresponding to each slot is indicated above the vertical dotted lines that delimit the slots. The stations and their associated visibility windows are represented with different colors on distinct lines. In this example, some visibility windows are overlapping (such as A and B or D and E). The data rate functions of the visibility windows are not shown in the figure.

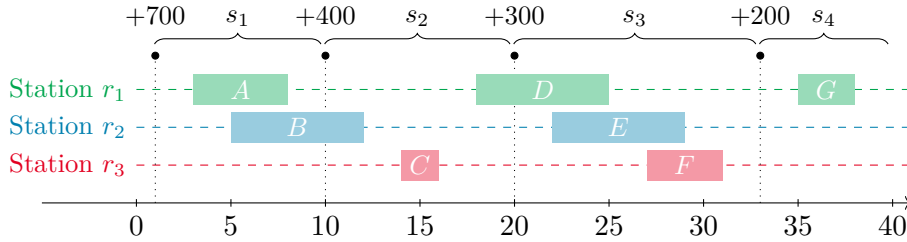


Figure 3.1 – Example of instance for the optical ground station network optimization problem.

Characteristics of optical links during communications between satellites and optical ground stations are not well known. Multiple parameters, mainly cloud interference, may influence the established link during a visibility windows. This creates very complicated data rate functions d_v . Furthermore, considered horizons are very large (multiple decades) while slots and visibility windows require a granularity of the order of tens of minutes. Having to deal with complex data rate functions over multiple decades would be infeasible, and likely excessive for the problem we consider, which is why we are going to reduce each visibility window to an instantaneous event called a **download point**.

We consider a mapping function $f_{v \rightarrow w}$ from visibility windows $v \in \mathcal{V}$ to download points $w \in \mathcal{Q}$. For each visibility window $v \in \mathcal{V}$, the mapping function creates one download point $w = f_{v \rightarrow w}(v) = (\sigma_w, \tau_w, \rho_w, \gamma_w)$, defined as follows:

- σ_w is the slot within which the visibility window starts, i.e., σ_w is the **only** slot $s_i \in \mathcal{S}$ such that:

$$t^{s_i} \leq t_v^{sta} < t^{s_{i+1}}$$

Chapter 3. Design of ground station networks for mission telemetry

- τ_w is the station associated to the visibility window, i.e., $\tau_w = \tau_v$;
- ρ_w is the download capacity or volume of the visibility window, which can be computed beforehand using the data rate function d_v :

$$\rho_w = \int_{t_v^{sta}}^{t_v^{end}} d_v(t) dt$$

- γ_w is the set of download points conflicting with w , i.e., the set of download points associated to visibility windows overlapping v :

$$\gamma_w = \{f_{v \rightarrow w}(v') : v' \in \gamma_v\}$$

Two download points are in conflict if their associated visibility windows are overlapping.

Example 2. Table 3.1 shows the mapping of visibility windows to download points for the example shown in Figure 3.1. Visibility windows parameters are shown on the left side of the table, while parameters of the corresponding download points are shown on the right side. For this example, we considered a simple constant data rate function μ identical for all visibility windows: $\forall t \in [T_{start}, T_{end}]$, $\mu(t) = 20$.

Visibility windows, $v \in \mathcal{V}$						Download points, $w = f_{v \rightarrow w}(v)$				
v	t_v^{sta}	t_v^{end}	τ_v	d_v	γ_v	w	σ_w	τ_w	ρ_w	γ_w
A	3	8	r_1	μ	$\{B\}$	A_w	s_1	r_1	100	$\{B_w\}$
B	5	12	r_2	μ	$\{A\}$	B_w	s_1	r_2	140	$\{A_w\}$
C	14	16	r_3	μ	\emptyset	C_w	s_2	r_3	40	\emptyset
D	18	25	r_1	μ	$\{E\}$	D_w	s_2	r_1	140	$\{E_w\}$
E	22	29	r_2	μ	$\{D, F\}$	E_w	s_3	r_2	140	$\{D_w, F_w\}$
F	27	31	r_3	μ	$\{E\}$	F_w	s_3	r_3	80	$\{E_w\}$
G	35	38	r_1	μ	\emptyset	G_w	s_4	r_1	60	\emptyset

Table 3.1 – Example of mapping from visibility windows to download points for the data shown in Figure 3.1.

In order to ease the explanations in the following sections, extra notations for download points are introduced. We denote by \mathcal{Q} the set of all download points:

$$\mathcal{Q} = \{f_{v \rightarrow w}(v) : v \in \mathcal{V}\} \quad (3.1)$$

For each slot $s \in \mathcal{S}$, we denote by \mathcal{Q}^s the set of download points inside s :

$$\mathcal{Q}^s = \{w \in \mathcal{Q} : \sigma_w = s\} \quad (3.2)$$

Similarly, for each location $r \in \mathcal{R}$, we denote by \mathcal{Q}_r the set of download points associated with r :

$$\mathcal{Q}_r = \{w \in \mathcal{Q} : \tau_w = r\} \quad (3.3)$$

It can be noticed that sets $(\mathcal{Q}^s)_{s \in \mathcal{S}}$ define a partition of \mathcal{Q} ($\bigcup_{s \in \mathcal{S}} \mathcal{Q}^s = \mathcal{Q}$ and $\forall s_i, s_j \in \mathcal{S}, \mathcal{Q}^{s_i} \cap \mathcal{Q}^{s_j} = \emptyset$). Sets $(\mathcal{Q}_r)_{r \in \mathcal{R}}$ also define a partition of \mathcal{Q} .

Example 3. Considering the example in Table 3.1, we obtain the following sets:

$$\begin{aligned} \mathcal{Q} &= \{A_w, B_w, C_w, D_w, E_w, F_w, G_w\} \\ \mathcal{Q}^{s_1} &= \{A_w, B_w\}, \quad \mathcal{Q}^{s_2} = \{C_w, D_w\}, \quad \mathcal{Q}^{s_3} = \{E_w, F_w\}, \quad \mathcal{Q}^{s_4} = \{G_w\} \\ \mathcal{Q}_{r_1} &= \{A_w, D_w, G_w\}, \quad \mathcal{Q}_{r_2} = \{B_w, E_w\}, \quad \mathcal{Q}_{r_3} = \{C_w, F_w\} \end{aligned}$$

Based on the assumption that visibility windows can be reduced to instantaneous events (download points), finding a network of optical ground stations maximizing the percentage of data transferred to the Earth can be re-formulated as two interleaved selection problems. The first problem corresponds to the selection of ground station locations and the second problem corresponds to the selection of download points.

3.1.2 Maximizing the percentage of data transferred

The MaxPDT problem consists in selecting a subset $\mathcal{R}' \subseteq \mathcal{R}$ of locations and a subset $\mathcal{Q}' \subseteq \mathcal{Q}$ of download points, in order to maximize the percentage of data transferred from a satellite to the Earth.

3.1.2.1 Definition of the MaxPDT problem

Definition 1. Instance of the MaxPDT problem

An instance of the MaxPDT problem is a 5-tuple $(K, B, \mathcal{R}, \mathcal{S}, \mathcal{Q})$ where $K > 0$ is the maximum cost allowed for opening stations, $B > 0$ is the size of the satellite buffer, \mathcal{R} ($|\mathcal{R}| = N$) is the set of available locations for ground stations, \mathcal{S} ($|\mathcal{S}| = M$) is the set of slots and \mathcal{Q} is the set of download points.

For each location $r \in \mathcal{R}$, $p_r \geq 0$ is the cost of opening a station at r and $\mathcal{Q}_r \subseteq \mathcal{Q}$ is the set of download points associated with r .

For each slot $s \in \mathcal{S}$, $a^s \in [0, B]$ is the amount of data acquired at the beginning of s and $\mathcal{Q}^s \subseteq \mathcal{Q}$ is the set of download points associated with s .

For each download point $w \in \mathcal{Q}$, $\tau_w \in \mathcal{R}$ is the location of w , $\sigma_w \in \mathcal{S}$ is the slot of w , $\rho_w \geq 0$ is the amount of data that can be downloaded using w and $\gamma_w \subset \mathcal{Q}$ is the set of download points in conflict with w .

By construction, each download point $w \in \mathcal{Q}$ is associated to a single location and a single slot:

$$\begin{aligned} \forall s_1, s_2 \in \mathcal{S} : \mathcal{Q}^{s_1} \cap \mathcal{Q}^{s_2} &= \emptyset \\ \forall r_1, r_2 \in \mathcal{R} : \mathcal{Q}_{r_1} \cap \mathcal{Q}_{r_2} &= \emptyset \end{aligned}$$

Definition 2. Feasible solution for the MaxPDT problem

Given an instance $(K, B, \mathcal{R}, \mathcal{S}, \mathcal{Q})$ of the MaxPDT problem, a feasible solution is a subset of locations and a subset of download points $(\mathcal{R}', \mathcal{Q}')$ with $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ such that:

$$\sum_{r \in \mathcal{R}'} p_r \leq K \quad (3.5a)$$

$$\mathcal{Q}' \subseteq \bigcup_{r \in \mathcal{R}'} \mathcal{Q}_r \quad (3.5b)$$

$$\forall w \in \mathcal{Q}', \quad \gamma_w \cap \mathcal{Q}' = \emptyset \quad (3.5c)$$

Equation (3.5a) indicates that the total costs of the chosen locations (opened stations) is lower than the maximum value allowed K . Equation (3.5b) enforces that download points can only be used if their associated locations are chosen. Equation (3.5c) indicates that if a download point w is chosen, none of the download points in conflict with w are selected.

Definition 3. Data loss of a solution for the MaxPDT problem

Given a solution $(\mathcal{R}', \mathcal{Q}')$ of the MaxPDT problem, the amount of data loss is the amount of data acquired by the satellite that has not been successfully transferred to the Earth. Data loss occurs when there is not enough space in the buffer to store the data acquired by the satellite at the beginning of a slot.

The function *losses* describes the computation of the amount of data loss. Given a set \mathcal{S} of M slots, a set of chosen download points \mathcal{Q}' and a buffer size B :

$$\text{losses}(\mathcal{Q}', \mathcal{S}, B) = b^{s_M} + \sum_{s \in \mathcal{S}} l^s \quad (3.6)$$

where l^s is the amount of data loss **during** the slot s and b^s the amount of data in the buffer **at the end** of the slot s . By convention, b^{s_0} is the amount of data at the beginning of the temporal horizon. We assume that the amount of data in the buffer at the end of the temporal horizon, b^{s_M} , is lost. The values of b^s and l^s can be computed using the following recursive equations:

$$b^{s_0} = 0 \quad (3.7a)$$

$$b^{s_i} = \max(0, \min(b^{s_{i-1}} + a^{s_i}, B) - \sum_{w \in \mathcal{Q}^{s_i} \cap \mathcal{Q}'} \rho_w), \quad i \in \{1, \dots, M\} \quad (3.7b)$$

$$l^{s_i} = \max(0, b^{s_{i-1}} + a^{s_i} - B), \quad i \in \{1, \dots, M\} \quad (3.7c)$$

At the beginning of the temporal horizon, the buffer is empty (3.7a). At the end of a slot s , the amount of data in the buffer is the amount of data in the buffer at the end of the previous slot, to which we add the acquisition volume of slot s and subtract the amount of data downloaded during slot s (3.7b). The amount of

3.1. Problem statement

data loss during a slot s is the amount of data acquired at the beginning of the slot a^s that did not fit in the buffer (3.7c).

Definition 4. Percentage of data transferred for a MaxPDT solution

Given a solution $(\mathcal{R}', \mathcal{Q}')$ of the MaxPDT problem, the percentage of data transferred is the amount of data acquired by the satellite during the time horizon that has been successfully downloaded:

$$pdt(\mathcal{Q}', \mathcal{S}, B) = 1 - \frac{\text{losses}(\mathcal{Q}', \mathcal{S}, B)}{\sum_{s \in \mathcal{S}} a^s} \quad (3.8)$$

The objective of the MaxPDT problem is to find a feasible solution $(\mathcal{R}', \mathcal{Q}')$ which maximizes the percent data transferred, which is the same as minimizing the losses $\text{losses}(\mathcal{Q}', \mathcal{S}, B)$ (defined in 3.6).

3.1.2.2 Mathematical model

To model the MaxPDT problem, we consider two types of binary variables for the selection of locations and the selection of download points. For each location $r \in \mathcal{R}$, we define a binary variable $y_r = 1$ if and only if the location r is chosen. For each download point $w \in \mathcal{Q}$, we define a binary variable $x_w = 1$ if and only if the download point w is used to download data. Moreover, we introduce two real variables for each slot $s \in \mathcal{S}$ to model the amount of data in the buffer and the amount of data loss (similar to the intermediate variables used in Definition 3.6). Variables $b^s \in \mathbb{R}^+$ represents the amount of data in the buffer **at the end of** slot s and variables $l^s \in \mathbb{R}^+$ represents the amount of data loss **during** slot s . Using this set of variables and the parameters given in Definition 1 (summarized in Table 3.2), the mathematical model for the MaxPDT problem can be formulated as follows:

$$\min. \quad \sum_{s \in \mathcal{S}} l^s \quad (3.9a)$$

$$\text{s.t.} \quad x_w \leq y_r, \quad r \in \mathcal{R}, \quad w \in \mathcal{Q}_r \quad (3.9b)$$

$$x_w + x_{w'} \leq 1, \quad w \in \mathcal{Q}, \quad w' \in \gamma_w \quad (3.9c)$$

$$b^{s_i} + l^{s_i} \geq b^{s_{i-1}} + a^{s_i} - \sum_{w \in \mathcal{Q}^{s_i}} x_w \rho_w, \quad s_i \in \mathcal{S} \quad (3.9d)$$

$$0 \leq b^{s_i} \leq B - a^{s_{i+1}}, \quad s_i \in \mathcal{S}, i \neq M \quad (3.9e)$$

$$b^{s_0} = b^{s_M} = 0 \quad (3.9f)$$

$$\sum_{r \in \mathcal{R}} p_r y_r \leq K \quad (3.9g)$$

$$x_w \in \{0, 1\}, \quad w \in \mathcal{Q} \quad (3.9h)$$

$$y_r \in \{0, 1\}, \quad r \in \mathcal{R} \quad (3.9i)$$

$$b^s \geq 0, \quad l^s \geq 0, \quad s \in \mathcal{S} \quad (3.9j)$$

Objective (3.9a) minimizes the amount of data loss. Since acquisition volume are inputs and since $b^{s_M} = 0$, the total amount of data loss is simply the sum of the data loss during each slot. Constraints (3.9b) and (3.9c) prevent downloads on stations that are not chosen ($y_r = 0$) and on conflicting download points. Constraints (3.9d) and (3.9e) force the amount of data at the end of a slot s_i to be consistent with the amount at the beginning of s_i and s_{i+1} , and to be less than the buffer size B minus the acquisition of slot s_{i+1} (i.e., at the end of slot s_i , there must be at least $a^{s_{i+1}}$ free space in the buffer). Constraint (3.9f) says that the initial amount of data in the buffer is 0 and forces the final amount of data in the buffer to be 0. Constraint (3.9g) forces the total cost of the network to be less than the maximum cost allowed K . Constraint (3.9h–3.9j) defines the domain of the decision variables.

3.1.3 The download points selection problem, $\text{MaxPDT}_{\mathcal{L}}$

The MaxPDT problem can grow very quickly in size, especially when we consider very large horizons (multiple decades). In order to tackle it more efficiently, we choose to first focus on one of its sub-problem: the selection of download points. The next section will be dedicated to the analysis of this problem, which is denoted $\text{MaxPDT}_{\mathcal{L}}$. We will first formally derive it from MaxPDT , and then provide some complexity results that can be extended to the MaxPDT problem.

3.1.3.1 Definition of the $\text{MaxPDT}_{\mathcal{L}}$ problem

Definition 5. Instance of the $\text{MaxPDT}_{\mathcal{L}}$ problem

An instance of the $\text{MaxPDT}_{\mathcal{L}}$ problem is a triplet $(B, \mathcal{S}, \mathcal{Q})$ where B , \mathcal{S} and \mathcal{Q} have the same meaning as in the MaxPDT problem (see Definition 1).

An instance of the $\text{MaxPDT}_{\mathcal{L}}$ problem is an instance of the MaxPDT problem where a selection of locations has previously been done.

Definition 6. Feasible solution for the $\text{MaxPDT}_{\mathcal{L}}$ problem

Given an instance $(B, \mathcal{S}, \mathcal{Q})$ of the $\text{MaxPDT}_{\mathcal{L}}$ problem, a feasible solution is a subset of download points $\mathcal{Q}' \subseteq \mathcal{Q}$ (3.10a) such that no two download points are in conflict (3.10b).

$$\mathcal{Q}' \subseteq \mathcal{Q} \tag{3.10a}$$

$$\forall w \in \mathcal{Q}', \quad \gamma_w \cap \mathcal{Q}' = \emptyset \tag{3.10b}$$

Definition 7. Optimal solution for the $\text{MaxPDT}_{\mathcal{L}}$ problem

Given an instance $(B, \mathcal{S}, \mathcal{Q})$ of the $\text{MaxPDT}_{\mathcal{L}}$ problem, an optimal solution is a feasible solution with minimum losses.

An optimal solution for $\text{MaxPDT}_{\mathcal{L}}$ is similar to an optimal solution for MaxPDT . An optimal solution for the MaxPDT problem can be found by solving a $\text{MaxPDT}_{\mathcal{L}}$ problem for each feasible subset of locations, and then taking the best solution found.

Inputs of the problem

T_{start}	Start of the temporal horizon.	T_{end}	End of the temporal horizon.
\mathcal{R}	Set of possible locations.	N	Number of possible locations.
p_r	Cost of opening a station on location r .	K	Maximum allowed costs for the network of stations.
B	Size of the buffer.		
<hr/>			
\mathcal{S}	Set of slots.	M	Number of slots.
t^s	Start of slot s .	a^s	Amount of data acquired at the beginning of slot s .
<hr/>			
\mathcal{V}	Set of visibility windows.	τ_v	Location reachable during visibility window v .
t_v^{sta}	Start of visibility window v .	t_v^{end}	End of visibility window v .
d_v	Data rate function (evolution) during visibility window v .	γ_v	Set of visibility windows overlapping with visibility window v .
<hr/>			
$f_{v \rightarrow w}$	Mapping function between visibility and download points.	\mathcal{Q}	Set of download points.
\mathcal{Q}^s	Set of download points associated with slot s .	\mathcal{Q}_r	Set of download points associated with location r .
σ_w	Slot of download point w .	τ_w	Location of download point w .
ρ_w	Amount of data that can be downloaded using w .	γ_w	Set of download points conflicting with w .
<hr/>			
Decision variables			
y_r	$y_r = 1$ if and only if location r is chosen.	x_w	$x_w = 1$ if and only if download point w is selected.
b^s	Amount of data in the buffer at the end of slots.	l^s	Amount of data lost during slot s .

Table 3.2 – Summary of notations for the definition of the MaxPDT and MaxPDT $_{\mathcal{L}}$ problems.

3.1.3.2 Mathematical model

The mathematical model for the $\text{MaxPDT}_{\mathcal{L}}$ problem is obtained by removing variables y_r and then removing constraints (3.9b) and (3.9g) from the model of the MaxPDT problem (3.9).

3.1.4 Complexity results

In this section, we give some complexity results for the MaxPDT and $\text{MaxPDT}_{\mathcal{L}}$ problems. We then provide results for special cases of instances of the $\text{MaxPDT}_{\mathcal{L}}$ problem regarding the distribution of download points and conflicts between them.

3.1.4.1 Complexity of the $\text{MaxPDT}_{\mathcal{L}}$ and MaxPDT problems

Proposition 1. *The $\text{MaxPDT}_{\mathcal{L}}$ problem is strongly NP-hard.*

Proof. Let us consider the decision problem associated with $\text{MaxPDT}_{\mathcal{L}}$: is it possible to find a solution \mathcal{Q}' such that $\text{losses}(\mathcal{Q}') \leq \phi$, ϕ being an arbitrary positive real value?

The proof is based on the reduction from the Weighted Independent Set problem (WIS), which is known to be NP-complete in the strong sense. The weighted independent set problem consists, given a graph $\mathcal{G} = (V, E)$ and weights $u : V \rightarrow \mathbb{Z}$, in finding a subset $S \subseteq V$ of vertices such that no two vertices in S are adjacent and such that the sum of the weights of vertices in S is greater than an arbitrary positive value Φ .

Obviously, $\text{MaxPDT}_{\mathcal{L}}$ is NP since, given a solution \mathcal{Q}' , $\text{losses}(\mathcal{Q}')$ can be computed in linear time using the formula given in (3.6).

From a WIS instance $(\mathcal{G} = (V, E), \Phi)$, we build up an instance of $\text{MaxPDT}_{\mathcal{L}}$ in the following way. The instance contains a single slot s_1 ($\mathcal{S} = \{s_1\}$) with an associated acquisition volume $a^{s_1} = \sum_{v \in V} u_v$ and a set of download points $\mathcal{Q} = \mathcal{Q}^{s_1}$. The set \mathcal{Q} contains a download point w_v for each vertex $v \in V$ with $\sigma_{w_v} = s_1$, $\rho_{w_v} = u_v$ and $\gamma_{w_v} = \{w_{v'} : (v, v') \in E\}$. This means that two download points are in conflict if their corresponding vertices are connected in the graph \mathcal{G} . The buffer size is $B = a^{s_1}$.

A feasible solution \mathcal{Q}' for this instance of $\text{MaxPDT}_{\mathcal{L}}$ is obviously a feasible solution S for the WIS instance due to the conflict constraints. Furthermore, finding a solution S_{Φ} such that $\sum_{v \in S_{\Phi}} u_v \geq \Phi$ is the same as finding a solution \mathcal{Q}'_{ϕ} of the $\text{MaxPDT}_{\mathcal{L}}$ instance, with $\phi = B - \Phi$ such that $\text{losses}(\mathcal{Q}'_{\phi}, \{s_1\}, B) \leq \phi = B - \Phi$.

3.1. Problem statement

Indeed, from (3.6) and (3.7a–3.7c), we can compute $losses(\mathcal{Q}'_\phi, \{s_1\}, B)$:

$$\begin{aligned}
losses(\mathcal{Q}'_\phi, \{s_1\}, B) &= l^{s_1} + b^{s_1} \\
&= \max(0, b^{s_0} + a^{s_1} - B) + b^{s_1} \\
&= b^{s_1} \\
&= \max(0, \min(b^{s_0} + a^{s_1}, B) - \sum_{w \in \mathcal{Q}^{s_1} \cap \mathcal{Q}'} \rho_w) \\
&= \max(0, a^{s_1} - \sum_{w \in \mathcal{Q}'} \rho_w) \\
&= a^{s_1} - \sum_{w \in \mathcal{Q}'} \rho_w \\
&= \sum_{v \in V} u_v - \sum_{v \in \mathcal{Q}'} u_v = B - \Phi
\end{aligned}$$

□

Corollary 1. *The MaxPDT problem is strongly NP-hard.*

Proof. The $\text{MaxPDT}_{\mathcal{L}}$ problem is a sub-problem of the MaxPDT problem where locations have already been chosen, it is thus trivial to reduce $\text{MaxPDT}_{\mathcal{L}}$ to MaxPDT, making MaxPDT strongly NP-hard. □

3.1.4.2 Special complexity cases for $\text{MaxPDT}_{\mathcal{L}}$

We propose here some complexity results for $\text{MaxPDT}_{\mathcal{L}}$ regarding instances with special distributions of download points and conflicts between download points.

Definition 8. We define the class of **intra** instances as the class of instances within which there are no conflicts between download points that are not in the same slot, i.e., given an instance $(B, \mathcal{S}, \mathcal{Q})$ of the **intra** class:

$$\forall w \in \mathcal{Q} : \gamma_w \subset \mathcal{Q}^{\sigma_w}$$

Definition 9. We define the class of **interval** instances as the class of instances within which the set of conflicts between download points represents intersection constraints between intervals, similarly to interval or intersection graph.

Proposition 2. *Solving instances of $\text{MaxPDT}_{\mathcal{L}}$ that are in **both** the **intra** and the **interval** classes can be done in polynomial time.*

Proof. Since there are no conflicts between download points not in the same slot, an optimal solution is a solution where the amount of data downloaded within each slot is maximized. Within any slot s , selecting a subset of stations maximizing the amount of data downloaded is the same as finding a maximum-weighted independent set in an interval graph with $|\mathcal{Q}^s|$ vertices (see the reduction above). This can be done in $\mathcal{O}(|\mathcal{Q}^s|)$ [Hsiao 1992] if the download points are correctly sorted.

Chapter 3. Design of ground station networks for mission telemetry

Thus, finding the optimal solution can be done in $\mathcal{O}(\sum_{s \in \mathcal{S}} |Q^s|) = \mathcal{O}(|Q|)$ if the download points are correctly sorted. Since sorting all the download points can be done in $\mathcal{O}(|Q| \cdot \log |Q|)$, the final complexity of the algorithm for instances in both the intra and the interval classes is $\mathcal{O}(|Q| \cdot \log |Q|)$. \square

It is worth noticing that instances of the $\text{MaxPDT}_{\mathcal{L}}$ problem constructed from real scenarios fall within the **interval** class. Some of these might be from the **intra** class — this will mostly depends on the chosen stations and orbit of the satellite — but this will not be the general case.

3.2 Hierarchical approaches to solve the MaxPDT problem

In this section we will present hierarchical approaches to solve the MaxPDT problem based on a dynamic programming algorithm to solve the $\text{MaxPDT}_{\mathcal{L}}$ problem.

In real instances, the number N of possible locations for the ground stations is often very small (some dozens) but the temporal horizon \mathcal{H} is large (multiple years). We propose to separate the decision process into two cooperative algorithms: a first algorithm that enumerates every possible subset of stations \mathcal{R}' , and a second algorithm that solves an instance of $\text{MaxPDT}_{\mathcal{L}}$ built from each of these subsets.

3.2.1 Algorithms for the enumeration of networks of ground stations

We propose two algorithms to enumerate the feasible subsets of locations. The first algorithm is an exhaustive enumeration and the second one is a branch and bound. Both of these enumerations use another algorithm, denoted \mathcal{A} , to solve a $\text{MaxPDT}_{\mathcal{L}}$ instance and return a list of selected download points (Q') together with the corresponding objective for the given instance. Such an algorithm will be presented in Section 3.2.2.

In order to obtain an instance of $\text{MaxPDT}_{\mathcal{L}}$ from an instance of MaxPDT for a subset of locations, we define the following operation.

Definition 10. Projection of a set \mathcal{S} of slots on a set \mathcal{R}' of locations: $\mathcal{S} \downarrow \mathcal{R}'$

We define the projection of a set \mathcal{S} of slots on a set \mathcal{R}' of locations as an updated set $\mathcal{S}_P = \mathcal{S} \downarrow \mathcal{R}'$ where download points on locations not in \mathcal{R}' have been removed. Given two related slots $s \in \mathcal{S}$ and $s^P \in \mathcal{S}_P$, the following hold:

$$a^{s^P} = a^s \quad \text{and} \quad Q^{s^P} = Q^s \cap Q_{\mathcal{R}'}, \quad (3.11)$$

where $Q_{\mathcal{R}'}$ is the set of all download points on stations in \mathcal{R}' :

$$Q_{\mathcal{R}'} = \bigcup_{r \in \mathcal{R}'} Q_r \quad (3.12)$$

3.2. Hierarchical approaches to solve the MaxPDT problem

The selection of feasible subsets of stations and the projection can be respectively seen as the enforcement of constraints (3.9g) and (3.9b) of the mathematical program proposed in Section 3.1.2.2.

3.2.1.1 Exhaustive enumeration

To solve the MaxPDT problem, the first algorithm we propose, $\mathcal{EE}_{pdt}(\mathcal{A})$, is a simple exhaustive enumeration that tries every possible combination of locations and uses algorithm \mathcal{A} to compute the percentage of data transferred corresponding to these combinations. Algorithm 1 gives an overview of $\mathcal{EE}_{pdt}(\mathcal{A})$.

Algorithm 1 Exhaustive enumeration algorithm

```

 $pdt_{\max} \leftarrow 0, \quad \mathcal{R}^* \leftarrow \emptyset, \quad \mathcal{Q}^* \leftarrow \emptyset$ 
for each  $\mathcal{R}' \subseteq \mathcal{R}$  do
  if  $\sum_{r \in \mathcal{R}'} p_r \leq K$  then
     $\mathcal{S}_P \leftarrow \mathcal{S} \downarrow \mathcal{R}'$ 
     $(pdt, \mathcal{Q}') \leftarrow \mathcal{A}(B, \mathcal{S}_P, \bigcup_{r \in \mathcal{R}'} \mathcal{Q}_r)$ 
    if  $pdt > pdt_{\max}$  then
       $pdt_{\max} \leftarrow pdt$ 
       $\mathcal{R}^* \leftarrow \mathcal{R}'$ 
       $\mathcal{Q}^* \leftarrow \mathcal{Q}'$ 
    end if
  end if
end for

```

3.2.1.2 Branch and bound algorithm

To solve the MaxPDT problem, the second algorithm we propose, $\mathcal{BB}_{pdt}(\mathcal{A})$, is a binary branch and bound algorithm.

In this algorithm, branching is done by imposing or forbidding the opening of a station at a possible location for which no decision has been made yet. On any given node of the branch and bound procedure, we split the set of locations \mathcal{R} into three disjoint subsets \mathcal{R}^+ , \mathcal{R}^- and $\mathcal{R}^?$ corresponding respectively to chosen locations, forbidden ones and still undefined ones.

The search tree is initialized with a single node with $\mathcal{R}^+ = \mathcal{R}^- = \emptyset$ and $\mathcal{R}^? = \mathcal{R}$. Nodes on the tree are processed in decreasing order of their upper bounds.

While processing a node, the algorithm branches on the location providing the highest amount of download. Given a solution \mathcal{Q}' obtained using \mathcal{A} for a projection (see Definition 10) of the slots on $\mathcal{R}^+ \cup \mathcal{R}^?$, the algorithm chooses the station r^{next} such that:

$$r^{next} = \arg \max_{r \in \mathcal{R}^?} \sum_{w \in \mathcal{Q}_r \cap \mathcal{Q}'} \rho_w$$

To compute an upper bound for a given node, we use algorithm \mathcal{A} with a projection of the slots on $\mathcal{R}^+ \cup \mathcal{R}^?$, i.e., we compute the objective value for a network

using all chosen locations (\mathcal{R}^+) and all undefined ones ($\mathcal{R}^?$). Since opening new stations can only increase the percentage of data transferred, this produces a valid upper bound.

When the branching procedure reaches a node that cannot be extended with any location in $\mathcal{R}^?$ without exceeding the maximum cost allowed, the procedure stops exploring the corresponding branch and this node is considered a leaf node:

$$leaf(\mathcal{R}^+, \mathcal{R}^-, \mathcal{R}^?) \iff \left(\forall r \in \mathcal{R}^?, \quad p_r + \sum_{r' \in \mathcal{R}^+} p_{r'} > K \right)$$

When a leaf node is reached, the percentage of data transferred of the corresponding solution is computed and compared to the percentage of the current best solution. If the new solution is better (i.e., its *PDT* is greater), the current best solution is updated.

3.2.2 Dynamic programming algorithm solve $\text{MaxPDT}_{\mathcal{L}}$

In the following, we present a dynamic programming algorithm, $DP_{pdt_{\mathcal{L}}}$, that can be used to solve $\text{MaxPDT}_{\mathcal{L}}$ for any classes of instances. This algorithm can stand for algorithm \mathcal{A} within the methods presented in Section 3.2.1.

3.2.2.1 Definition of the $DP_{pdt_{\mathcal{L}}}$ algorithm

The proposed $DP_{pdt_{\mathcal{L}}}$ algorithm manages a set \mathcal{H} of labels, where each label h is a tuple $(b_h, l_h, \Gamma_h, \mathcal{W}_h)$ with b_h the current amount of data in the buffer, l_h the accumulated amount of data loss, Γ_h the set of conflicting download points (i.e., the set of download points that cannot extend this label) and \mathcal{W}_h the list of download points used.

Algorithm 2 shows a formal description of the algorithm. The algorithm processes with the following steps:

1. a label $h_0 = (0, 0, \emptyset, \emptyset)$ is created and added to the initially empty set of labels \mathcal{H} (line 1);
2. the slots in \mathcal{S} are processed in chronological order (regarding their start times). For each slot $s \in \mathcal{S}$, the following actions are performed:
 - (a) all existing labels are updated to take into account the amount of data acquired at the beginning of s (lines 3–6);
 - (b) for each download point $w \in \mathcal{Q}^s$ and for each label $h \in \mathcal{H}$, a new label h' is created if the download point w is not in conflict with the label h , i.e., $w \notin \gamma_w$ (lines 7–16);
3. labels in \mathcal{H} are updated to take into account constraint (3.9f) (the data in the buffer at the end of the temporal horizon is lost) and the label with the minimum amount of data loss at the end is chosen (lines 18–22).

3.2. Hierarchical approaches to solve the MaxPDT problem

Algorithm 2 Dynamic Programming Algorithm $DP_{pdt_{\mathcal{L}}}$

```

1:  $\mathcal{H} \leftarrow \{(0, 0, \emptyset, \emptyset)\}$  ▷ step 1
2: for each  $s \in \mathcal{S}$  do ▷ step 2
3:   for each  $h \in \mathcal{H}$  do ▷ step 2.1
4:      $b_h \leftarrow \min(b_h + a^s, B)$ 
5:      $l_h \leftarrow l_h + \max(0, b_h + a^s - B)$ 
6:   end for
7:   for each  $w \in \mathcal{Q}^s$  do ▷ step 2.2
8:      $\mathcal{H}^+ \leftarrow \emptyset$ 
9:     for each  $h \in \mathcal{H}$  do
10:      if  $w \notin \Gamma_h$  then
11:         $h' \leftarrow (\max(0, b_h - \rho_w), l_h, \mathcal{W}_h \cup \{w\}, \Gamma_h \cup \gamma_w)$ 
12:         $\mathcal{H}^+ \leftarrow \mathcal{H}^+ \cup \{h'\}$ 
13:      end if
14:    end for
15:     $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}^+$ 
16:  end for
17: end for
18: for each  $h \in \mathcal{H}$  do ▷ step 3
19:    $l_h \leftarrow l_h + b_h$ 
20:    $b_h \leftarrow 0$ 
21: end for
22: return  $\operatorname{argmin}_{h \in \mathcal{H}}(l_h)$ 

```

3.2.2.2 Dominance rule to prune labels, $DP_{pdt_{\mathcal{L}}}^+$

In order to prevent the set of labels \mathcal{H} from growing too rapidly, we need to remove labels that cannot lead to optimal solutions by means of a dominance rule between two labels.

Definition 11. We say that a label h_1 dominates a label h_2 if $h_1 \neq h_2$ and:

$$\Gamma_{h_1}^+ = \Gamma_{h_2}^+ \quad \text{and} \quad \left(\begin{array}{l} b_{h_1} < b_{h_2} \wedge l_{h_1} \leq l_{h_2} \\ \text{or } b_{h_1} = b_{h_2} \wedge l_{h_1} < l_{h_2} \\ \text{or } b_{h_1} = b_{h_2} \wedge l_{h_1} = l_{h_2} \wedge \mathcal{W}_{h_1} \prec \mathcal{W}_{h_2} \end{array} \right) \quad \begin{array}{l} (3.13a) \\ (3.13b) \\ (3.13c) \end{array}$$

Where Γ_h^+ is a subset of Γ_h containing only download points that have not yet been processed. This is due to the fact that labels can be compared only if they have the same sets of conflicts in the future.

Condition (3.13a) compares labels according to the amount of data loss (current objective value) and amount of data in the buffer. If there are less data loss and less data in the buffer in h_1 than in h_2 , the solution for h_1 is better than the one for h_2 . Since $\Gamma_{h_1}^+ = \Gamma_{h_2}^+$, any choice possible for extending h_2 is also valid for h_1 (the only constraints for extension come from conflicting download points), so if h_1

Chapter 3. Design of ground station networks for mission telemetry

is better than h_2 , there will be at least one solution made from extending h_1 that will be better than any solutions created by extending h_2 , thus h_1 dominates h_2 . Condition (3.13c) is only used to avoid solutions with identical objective values. Two solutions may have the same amount of data loss and amount of data in the buffer, keeping both of them would be inefficient, so we remove the one with the worst set of download points used (\prec must be a strict total order, e.g., a lexicographic comparison between the two sets of download points).

Pruning of labels is done within step 2.2 in Algorithm 2, at the beginning of the outer for loop (between line 7 and 8). This dominance rule ensures that no label than can lead to an optimal solution will be pruned. Furthermore, we know that if all labels $h \in \mathcal{H}$ have an empty set of conflicts ($\Gamma_h = \emptyset$) and an empty buffer $b_h = 0$, they can be compared using (3.13b–3.13c) and a single one dominates all the others. This guarantees that at the end of Algorithm 2, a single label remains. We call the combinations of the $DP_{pdt_{\mathcal{L}}}$ algorithm with the above defined dominance rule $DP_{pdt_{\mathcal{L}}}^+$.

Example 4. We will carry out the $DP_{pdt_{\mathcal{L}}}^+$ execution on the example presented in Section 3.1.1. We assume here that the three stations in the example are selected and that the buffer has a size of 1000. At each step of the algorithm, we display the current list of labels \mathcal{H} in a table.

Step 1 We start by initializing \mathcal{H} with the initial label:

b_h	l_h	\mathcal{W}_h	Γ_h
h_1	0	0	\emptyset

Step 2 We start processing the first slot s_1 ($a^{s_1} = 700$) by updating all labels inside \mathcal{H} according to step 2.1:

b_h	l_h	\mathcal{W}_h	Γ_h
h_1	700	0	\emptyset

We process the first download point A_w ($\rho_{A_w} = 100$ and $\gamma_{A_w} = \{B_w\}$). This creates a new label h_2 by extending h_1 that we add to \mathcal{H} :

b_h	l_h	\mathcal{W}_h	Γ_h
h_1	700	0	\emptyset
h_2	600	0	$\{A_w\}$

We process the second download point B_w ($\rho_{B_w} = 140$ and $\gamma_{B_w} = \{A_w\}$). Since

3.2. Hierarchical approaches to solve the MaxPDT problem

Γ_{h_2} contains B_w , we can only extend h_1 and create a new label h_3 :

	b_h	l_h	\mathcal{W}_h	Γ_h
h_1	700	0	\emptyset	\emptyset
h_2	600	0	$\{A_w\}$	$\{B_w\}$
h_3	560	0	$\{B_w\}$	$\{A_w\}$

We start processing the second slot s_2 ($a^{s_2} = 400$) by updating all labels inside \mathcal{H} . Since the buffer size is limited to 1000, some data are lost when updating h_1 ($b_{h_1} + a^{s_2} = 700 + 400 > 1000$):

	b_h	l_h	\mathcal{W}_h	Γ_h
h_1	1000	100	\emptyset	\emptyset
h_2	1000	0	$\{A_w\}$	$\{B_w\}$
h_3	960	0	$\{B_w\}$	$\{A_w\}$

Before processing the third download point, the dominance rule is applied (at the beginning of step 2.2). Since A_w and B_w have already been processed, we have $\Gamma_{h_1}^+ = \Gamma_{h_2}^+ = \Gamma_{h_3}^+$. We can see that h_3 dominates h_1 and h_2 due respectively to equations 3.13b and 3.13a.

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_3	960	0	$\{B_w\}$	\emptyset

For the sake of readability, we will only display future conflicts Γ_h^+ in the tables starting from here.

Once the dominance rule has been applied, we can process the third download point C_w ($\rho_{C_w} = 40$ and $\gamma_{C_w} = \emptyset$). Since Γ_{h_3} does not contain C_w , we can extend it and create a new label h_4 :

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_3	960	0	$\{B_w\}$	\emptyset
h_4	920	0	$\{B_w, C_w\}$	\emptyset

Here we can notice that since C_w has no conflicts, any label h_x extended to a new label h_y will be removed by the next application of the dominance rule. Here h_3 was extended to h_4 , and h_4 now dominates h_3 :

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_4	920	0	$\{B_w, C_w\}$	\emptyset

Chapter 3. Design of ground station networks for mission telemetry

We now process D_w ($\rho_{D_w} = 140$ and $\gamma_{D_w} = \{E_w\}$) and create a new label h_5 by extending h_4 . Then we start processing s_3 ($a^{s_3} = 300$) and update h_4 and h_5 :

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_4	1000	220	$\{B_w, C_w\}$	\emptyset
h_5	1000	80	$\{B_w, C_w, D_w\}$	$\{E_w\}$

The dominance rule cannot remove any labels at this point because $\Gamma_{h_4}^+$ and $\Gamma_{h_5}^+$ are different. By processing E_w ($\rho_{E_w} = 140$ and $\gamma_{E_w} = \{D_w, F_w\}$), a new label h_6 is created by extending h_4 (we cannot extend h_5 because Γ_{h_5} contains E_w):

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_4	1000	220	$\{B_w, C_w\}$	\emptyset
h_5	1000	80	$\{B_w, C_w, D_w\}$	\emptyset
h_6	860	220	$\{B_w, C_w, E_w\}$	$\{F_w\}$

We then apply the dominance rule which removes h_4 (dominated by h_5) and process F_w ($\rho_{F_w} = 80$ and $\gamma_{F_w} = \{E_w\}$) which creates a new label h_7 by extending h_5 (h_6 cannot be extended because Γ_{h_6} contains F_w):

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_5	1000	80	$\{B_w, C_w, D_w\}$	\emptyset
h_6	860	220	$\{B_w, C_w, E_w\}$	\emptyset
h_7	920	80	$\{B_w, C_w, D_w, F_w\}$	\emptyset

The labels are then updated for s_4 ($a^{s_4} = 200$), and before processing G_w , the dominance rule is applied removing h_5 and h_6 which are dominated by h_7 . h_7 is then extended to h_8 when processing G_w ($\rho_{G_w} = 60$ and $\gamma_{G_w} = \emptyset$):

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_7	1000	200	$\{B_w, C_w, D_w, F_w\}$	\emptyset
h_8	940	200	$\{B_w, C_w, D_w, F_w, G_w\}$	\emptyset

Step 3 Labels are updated at the end of the algorithm to remove data in the buffer (the buffer must be empty at the end of the time horizon):

	b_h	l_h	\mathcal{W}_h	Γ_h^+
h_7	0	1200	$\{B_w, C_w, D_w, F_w\}$	\emptyset
h_8	0	1140	$\{B_w, C_w, D_w, F_w, G_w\}$	\emptyset

Finally, h_8 dominates h_7 , thus the optimal solution is:

$$\mathcal{Q}^* = \{B_w, C_w, D_w, F_w, G_w\}$$

3.2.2.3 Complexity of $DP_{pdt_{\mathcal{L}}}^+$ for special classes of instances

Proposition 3. *The $DP_{pdt_{\mathcal{L}}}^+$ algorithm is Fixed-Parameter Tractable in the maximum number of download points inside a slot for instances of the **intra** class.*

Proof. At the beginning of the algorithm, the set \mathcal{H} contains a single label. We will show that, for instances of the *intra* class, if there is a single label in \mathcal{H} at the beginning of the processing of a slot s , then a single label remains in \mathcal{H} at the end of the processing of s (due to the dominance rule).

Assume there is a single label $h = (b_h, l_h, \Gamma_h, \mathcal{W}_h)$ in \mathcal{H} at the beginning of the processing of a slot s . First, b_h and l_h are updated (no new labels are created). Then the download points inside s are processed and new labels are created. All these labels have the same updated value of l_h (see algorithm 2).

At the end of the processing of the last download point inside s , the pruning happens. Since for instances of the *intra* class, there cannot be conflict between download points not in the same slot, all labels will have the same reduced set of conflicts $\Gamma^+ = \emptyset$. Since all labels have the same amount of data loss l_h , a single label in \mathcal{H} dominates all the others. Thus this label will be the only one remaining in \mathcal{H} at the end of the processing of s . For any slot s , the processing of the download points inside \mathcal{Q}^s is done in $\mathcal{O}(2^{|\mathcal{Q}^s|})$, thus the complexity of the $DP_{pdt_{\mathcal{L}}}^+$ algorithm for instances of the **intra** class is:

$$\mathcal{O}(|\mathcal{S}| \cdot 2^{\alpha_{max}})$$

where $\alpha_{max} = \max_{s \in \mathcal{S}} |\mathcal{Q}^s|$. □

3.3 Computational results

In this section, we present computational results for both $\text{MaxPDT}_{\mathcal{L}}$ and MaxPDT . We first present results obtained by solving randomly generated instances for the $\text{MaxPDT}_{\mathcal{L}}$ problem, and then results obtained for realistic instances of the MaxPDT problem.

3.3.1 Computational context

All algorithms and mathematical models for the $\text{MaxPDT}_{\mathcal{L}}$ and MaxPDT problems are implemented in C++. The *MILP* solver used is *CPLEX 12.7*. All experiments are run on a cluster with Xeon E5-2695 v3 and Xeon E5-2695 v4 processors. The slight differences between these two types of processors are negligible for our experiments.

3.3.2 Results for the MaxPDT_ℒ problem

In this section, we explain how we generate random instances for the MaxPDT_ℒ problem, and we compare the proposed $DP_{pdt_{\mathcal{L}}}^+$ algorithm with the mathematical model for MaxPDT_ℒ by studying the performances of each method depending on the types of generated instances.

3.3.2.1 Parameters for the MaxPDT_ℒ experiments

Experiments for the MaxPDT_ℒ problem detailed in this section are made using a single core (thread) with 6 gigabytes of dedicated RAM. A time limit of one hour is used for both the $DP_{pdt_{\mathcal{L}}}^+$ algorithm and the *MILP* solver.

Instances We generate random instances for the MaxPDT_ℒ problem, grouped into 3 categories depending on their conflicts:

- *Adj*: Conflicts can only occur between successive download points.
- *Int*: Conflicts can only occur within a slot (i.e., download points in different acquisition slots cannot be in conflict). This category corresponds to the **intra** class of instances defined in Section 3.1.4.
- *All*: Conflicts are not constrained.

For each category, instances are generated using a given number of slots, a random number (within a given range) of download points per slot, a fixed buffer size, a probability of conflict between download points, and randomly generated acquisition and download volumes.

For instances of the *Int* or *Adj* categories, we consider instances with 100, 500, 1000, 2000, 3000, 4000 and 5000 slots and with number of download points per slot within [0..5], [5..10], [10..20] and [20..40].

Instances of the *All* category are much harder to solve than instances from the *Int* or *Adj* categories (see Table 3.3), which is why the parameters used to generate these instances are different from the parameters used for the *Int* and *Adj* categories. For instances of the *All* category, we consider instances with 10, 50, 100, 200, 300, ..., 900 and 1000 slots and with number of download points per slot within [0..5], [5..10] and [10..20].

For each category of instances and for each combination of parameters (number of slots and number of download points), four probabilities of conflict are considered: 0.2, 0.4, 0.6, 0.8, and four different buffer sizes.

Download volumes of download points are uniformly distributed between 100 and 200 gigabits for all instances. The acquisition volumes are uniformly distributed between $150 * M$ and $250 * M$, where M is the maximum possible number of download points per slot. The buffer sizes used are multiples of the average acquisition volumes $\hat{a} = 200 * M$: $B = \hat{a}, 2\hat{a}, 4\hat{a}, 8\hat{a}$.

These parameters are chosen to evaluate the performance of the $DP_{pdt_{\mathcal{L}}}^+$ algorithm and compare it with the *MILP* model. Although the number of slots is

3.3. Computational results

somehow realistic (slots are typically one hour long, so 5000 slots represent a bit more than 6 months), it is unlikely to encounter that much download points per slot or so many conflicts in real applications.

For each category of instances (*Adj*, *Int*, *All*) and for each combination of parameters (number of slots, number of download points per slot, probability of conflict and buffer size), we generate four instances for a total of 5888 instances.

3.3.2.2 Overall performances of the considered approaches

An overview of the performances of each method depending on the category of the instances is presented in Table 3.3. In this Table, the third column indicates the number of instances tested per category, and the three right-most columns indicate respectively the number of instances for which the optimal solution is found, a feasible solution is found or no solutions is found.

	$DP_{pdt_{\mathcal{L}}}^+$			<i>MILP</i>		
	Optimal	Non optimal	Not found	Optimal	Non optimal	Not found
<i>Adj</i>	1792	0	0	1792	0	0
<i>Int</i>	1792	0	0	1587	165	40
<i>All</i>	369	48	1887	336	1496	472

Table 3.3 – Comparison of the number of instances solved between $DP_{pdt_{\mathcal{L}}}^+$ and the *MILP* solver for each category of instances.

Both methods have no problem with instances of the *Adj* categories. The $DP_{pdt_{\mathcal{L}}}^+$ algorithm manages to solve all instances of the *Int* category, even those with up to 40 download points per slot, while the *MILP* fails to solve some of these. For instances of the *All* category, while the *MILP* provides solutions (even if non-optimal) for about 80% of the instances, the $DP_{pdt_{\mathcal{L}}}^+$ algorithm only manages to provide solutions for about 18% of the instances. Even if the number of instances solved optimally is slightly larger for $DP_{pdt_{\mathcal{L}}}^+$, some instances solved to optimality by the *MILP* are not solved up to optimality by the $DP_{pdt_{\mathcal{L}}}^+$ algorithm.

Since it is trivial to find feasible solutions for any kind of instances (select no stations), instances for which both methods do not manage to provide solutions are instances for which the system run out of memory and kill the program.

In the following, we look in details at the performance of both methods when solving instances of the *Adj* and *Int* categories, and give some information regarding the performance of both methods for the *All* category.

3.3.2.3 Performances on each category of instances

The computation times of the *MILP* model and of the $DP_{pdt_{\mathcal{L}}}^+$ algorithm for instances of the *Adj* and *Int* categories are shown in Figures 3.2 and 3.3 respectively.

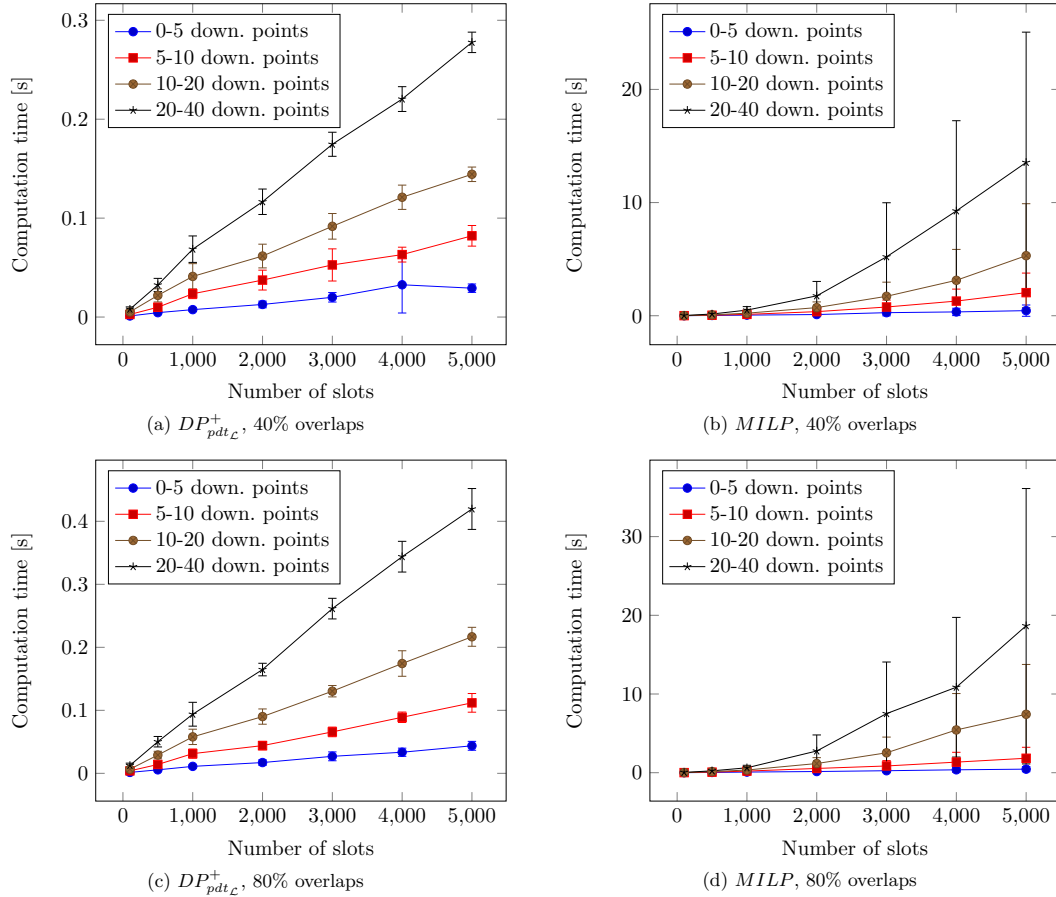


Figure 3.2 – Computation time for instances of the *Adj* category using $DP_{pdt_{\mathcal{L}}}^+$ or the *MILP*.

Each curve shows data averaged over the four instances of each group (same sets of parameters) and over the various buffer sizes used. We choose to represent results this way because these experiments show that the buffer size has little influence on the solving time of the algorithms. Instances of the *Int* category for which the *MILP* does not find solutions (or does not find the optimal solution) are included in these graphs with a computation time of 3600 seconds (the time limit used) so that each pair of plots correspond to the same set of instances.

We can see that for these two categories of instances, the $DP_{pdt_{\mathcal{L}}}^+$ algorithm is orders of magnitude faster than the *MILP* solver.

For instances of the *Adj* category, Figures 3.2a–3.2c show a near-linear behavior for $DP_{pdt_{\mathcal{L}}}^+$ regarding the number of slots and the number of download points per slot, but an exponential behavior for the *MILP*. Furthermore, $DP_{pdt_{\mathcal{L}}}^+$ seems to be less impacted by minor changes in the input instances than the *MILP* model. For both methods, the number of conflicts (percentage of overlaps) has little impact on the computation times for the considered instances, which is why results are only

3.3. Computational results

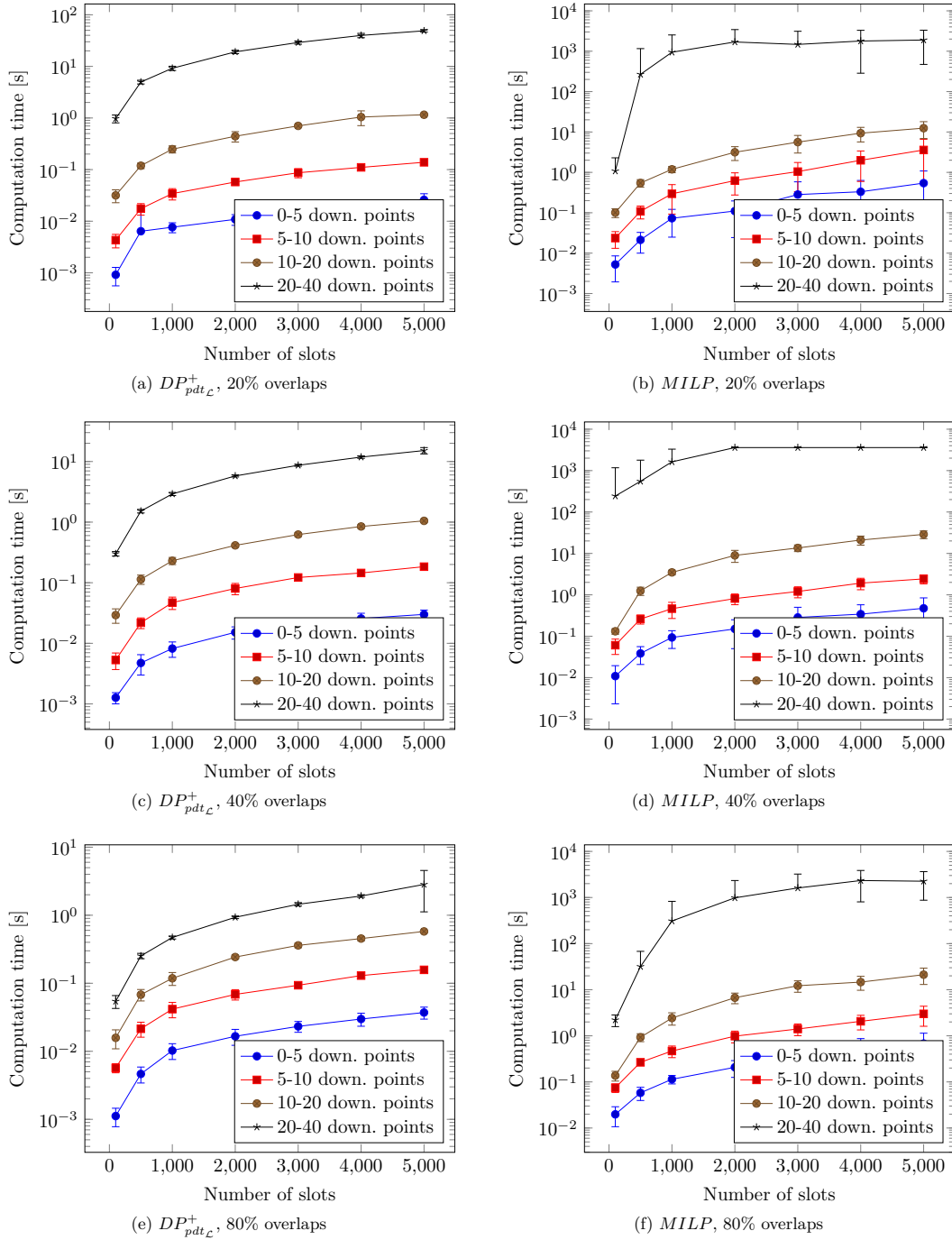


Figure 3.3 – Computation time for instances of the *Int* category using $DP_{pdt_c}^+$ or the *MILP*.

provided for the 40% and 80% probabilities of conflict.

For instances of the *Int* category, we can see, in Figures 3.3a–3.3e, that as the number of slots in the instances grows larger, the computation time of the $DP_{pdt_{\mathcal{L}}}^+$ algorithm seems to stabilize to a linear behavior. This behavior is explained by the *FPT* behavior of $DP_{pdt_{\mathcal{L}}}^+$ for instances of the *Int* category — the fixed parameter is the maximum number of download points per slot, which is constant for each plot in these figures. These plots also show that $DP_{pdt_{\mathcal{L}}}^+$ is more efficient at solving instances with a large number of overlaps. This can easily be explained by the fact that if a label is in conflict with a download point, the label is not duplicated, thus the more conflicts there are, the less label duplication occur. Figures 3.3b–3.3f show a similar but less pronounced behavior regarding the number of conflicts for the *MILP* solver.

The number of solved instances from the *All* category (to optimality or not) by both methods for different probabilities of conflict is shown in Table 3.4.

Conflict	$DP_{pdt_{\mathcal{L}}}^+$			<i>MILP</i>		
	Optimal	Non optimal	Not found	Optimal	Non optimal	Not found
20%	32	0	544	64	459	53
40%	65	0	511	80	395	101
60%	96	1	479	86	346	144
80%	176	47	353	106	296	174

Table 3.4 – Comparison of the number of instances of the *All* category solved between $DP_{pdt_{\mathcal{L}}}^+$ and the *MILP* for different probabilities of conflict.

For each probability of conflict, 576 instances are evaluated, and the columns respectively represent the number of instances where the optimal solution, a feasible solution or no solutions is found.

Similarly to instances of the *Adj* and *Int* categories, we see that $DP_{pdt_{\mathcal{L}}}^+$ is able to solve more instances when the percentage of conflicts is high. As this percentage increases, the *MILP* seems to have issue finding feasible solutions — the number of instances without solution increases — but at the same time the solver manages to find optimal solutions more often.

These experiments show that $DP_{pdt_{\mathcal{L}}}^+$ is much faster at solving instances of $\text{MaxPDT}_{\mathcal{L}}$ than the *MILP* model, even if both methods fail to solve complex instances of $\text{MaxPDT}_{\mathcal{L}}$ where there are a large number of conflicts and slots, and where conflicts are not confined.

3.3.3 Results for the MaxPDT problem

In this section, we explain how we generate realistic instances for the MaxPDT problem. We then compare the performances of the approaches presented in Section 3.2 and analyze a bit more in-depth the behavior of the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm.

3.3.3.1 Parameters for the MaxPDT experiments

Experiments for the MaxPDT problem detailed in this section are first made using a single core (thread) with 3.5 gigabytes of dedicated RAM, and then 8 threads with 28 gigabytes of RAM. In both cases, a time limit of two hours is used for the $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm, the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm and the *MILP* solver.

Instances We generate custom instances for a realistic scenario inspired from the concepts of operations from [IOAG 2012, Guérin 2010b, Lacoste 2011]: data rate (before cloud impact) $\mathcal{D}_{\mathcal{R}} = 10.5$ gigabits-per-second, buffer size $B = 2300$ gigabits, a slot s every hour and a constant acquisition volume at the beginning of each slot $a^s = 500$ gigabits. Visibility windows are computed using the *Systems Tool Kit (AGI, STK)* software, for a sun-synchronous low-Earth orbiting satellite with an altitude of 700 kilometers and a local time of 10:30 A.M, between 1990 and 2010. We use the *ERA Interim* cloud database [Dee 2011] to approximate the cloud cover $c_v \in [0, 1]$ during any visibility window v , using cloud data from the previous decades (between 1990 and 2010). Based on these historic cloud covers, we assume that the data rate function d_v of a visibility window v is constant during the visibility window and proportional to the cloud cover c_v over the station τ_v at the beginning of the visibility window:

$$d_v(t) = \mathcal{D}_{\mathcal{R}} * (1 - c_v), \quad \forall t \in [t_v^{sta}, t_v^{end}]$$

The download volume ρ_w of a download point w associated to a visibility window v can thus be computed as:

$$\rho_w = \int_{t_v^{sta}}^{t_v^{end}} \mathcal{D}_{\mathcal{R}}(1 - c_v) dt = \mathcal{D}_{\mathcal{R}}(1 - c_v)(t_v^{end} - t_v^{sta})$$

We discard download points that have too small capacity ($\rho_w < 1$ gigabits).

We consider three different sets of locations composed of 11 (\mathcal{N}_{11}), 16 (\mathcal{N}_{16}) and 48 (\mathcal{N}_{48}) possible locations for optical ground stations. These sets are described in Appendix A.1. We could not find realistic information for the costs of opening stations at the different locations so we choose to simply select varying numbers K of stations (between 1 and 11, 16 or 48 depending on the considered set of locations).

We generate instances of various temporal horizons of 1, 2, 4, 5, 10 and 20 years. All instances are generated using the same 20 years horizon (1990–2009), meaning that there are respectively 20, 10, 5, 4, 2, 1 instances of each duration.

For each of the considered set of locations, we look for networks consisting of $K = 1 \dots N$ stations (with $N = 11, 16$ or 48 depending on the set of locations). For \mathcal{N}_{48} with 48 stations, we only run the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ because the *MILP* model already has troubles solving instances for the two smaller sets, and the number of possible networks for an exhaustive enumeration is too large for a two hours time limit.

For the 20 year horizon, chosen locations for each set and value of K are shown

in Appendix B.

In the next section, we compare the three approaches on the \mathcal{N}_{11} and \mathcal{N}_{16} sets. Then we analyze the impact of using multiple threads for each method. Finally, we look at the results of the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm in details and introduce the results of experiments using \mathcal{N}_{48} .

3.3.3.2 Comparison between the considered approaches

Figure 3.4 shows the computation times (means and standard deviations) for the *MILP* solver and both the $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ and $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithms regarding the length of the temporal horizon in the instances for various choices of K (number of chosen stations). Figures 3.4a–3.4b show that the *MILP* solver either does not succeed to provide solution, or does not manage to prove optimality (the time limit of 7200 seconds is reached) for some instances with large temporal horizons (20 years for \mathcal{N}_{11} and 10 or 20 years for \mathcal{N}_{16}). These figures also show an exponential behavior with respect to the length of the temporal horizon and the number of chosen stations in the instances for the *MILP* model. Figures 3.4c–3.4f show that both our enumeration methods have a near-linear behavior regarding the length of the temporal horizon. This can be explained by the FPT behavior of the $DP_{pdt_{\mathcal{L}}}^+$ on instances with no conflict between slots. In real instances, there are few conflicts between slots and the number of download points per slot does not depend on the length of the temporal horizon. Figure 3.4 shows that the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ enumeration is faster than the $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ enumeration for the generated instances, and that both enumerations outperform the *MILP* solver. Moreover, the standard deviations of the computation time for our hierarchical approaches are lower than the standard deviations for the *MILP* solver, meaning that our methods are not really impacted by slight differences in the input instances.

Figure 3.5 shows the computation times (averages and standard deviations) for the *MILP* solver and both algorithms regarding the number of chosen stations for various temporal horizons. We do not display results for the 2-year and 5-year temporal horizons because these are very similar to the 1-year and 4-years horizons. These plots present similar information as Figure 3.4 but in a different way. Figures 3.5c–3.5d show an exponential behavior for the exhaustive enumeration $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$. Instances where the number of chosen stations is close to half the number of available stations are the hardest to solve. This is expected since the highest number of possible subsets of locations appears in these cases. While Figure 3.5f depicts the same exponential behavior for the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm, Figure 3.5e shows a more chaotic behavior for \mathcal{N}_{11} . Figures 3.5a–3.5b also show an exponential behavior for the *MILP* solver but slightly shift to the left — the hardest instances seem to be those where the number of chosen stations is slightly less than half the number of available locations.

These results show that solving realistic instances of MaxPDT by combining an enumeration method and the $DP_{pdt_{\mathcal{L}}}^+$ algorithm is much faster than using a commercial *MILP* solver. Experiments also show that the branch and bound enumeration

3.3. Computational results

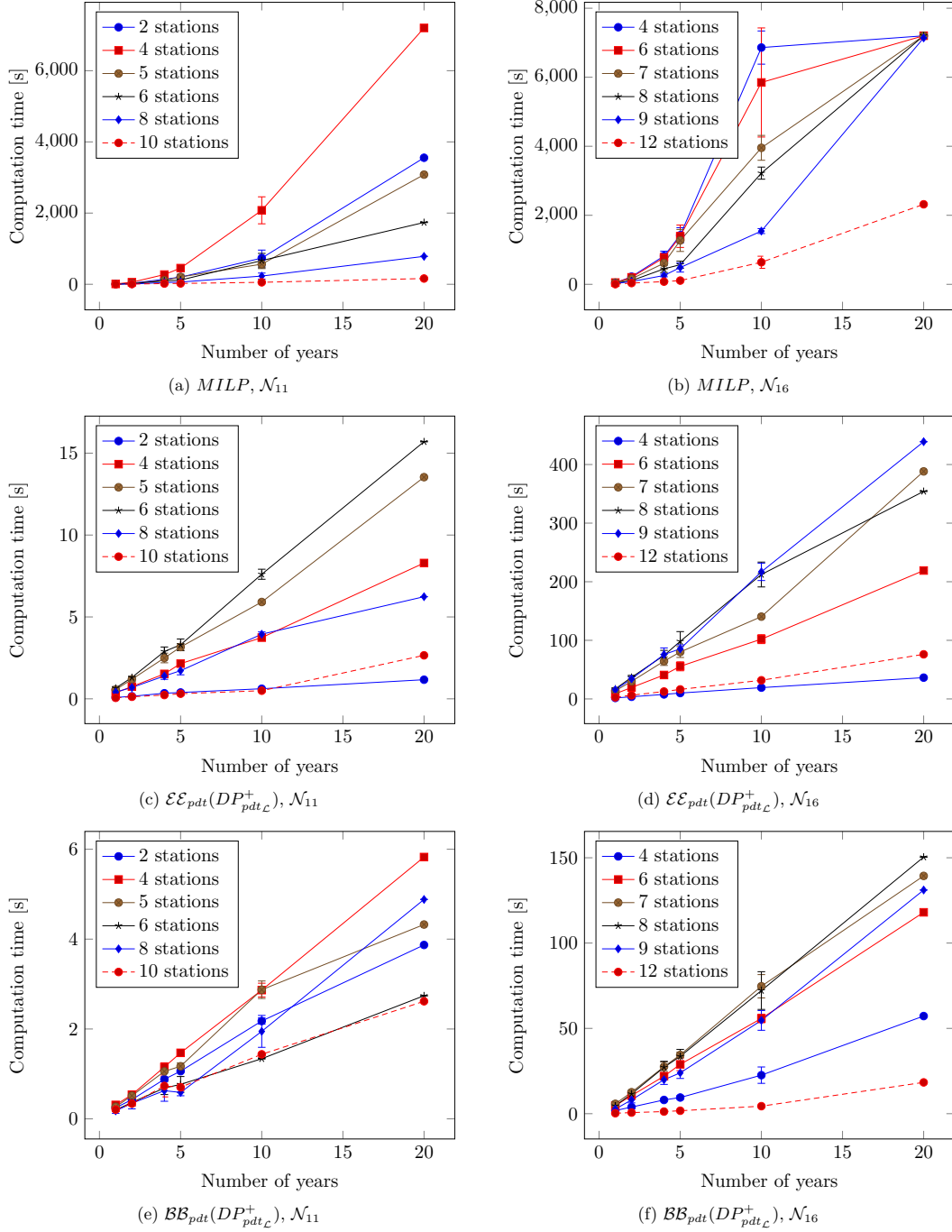


Figure 3.4 – Computation times for each methods using 8 threads, depending on the temporal horizon length.

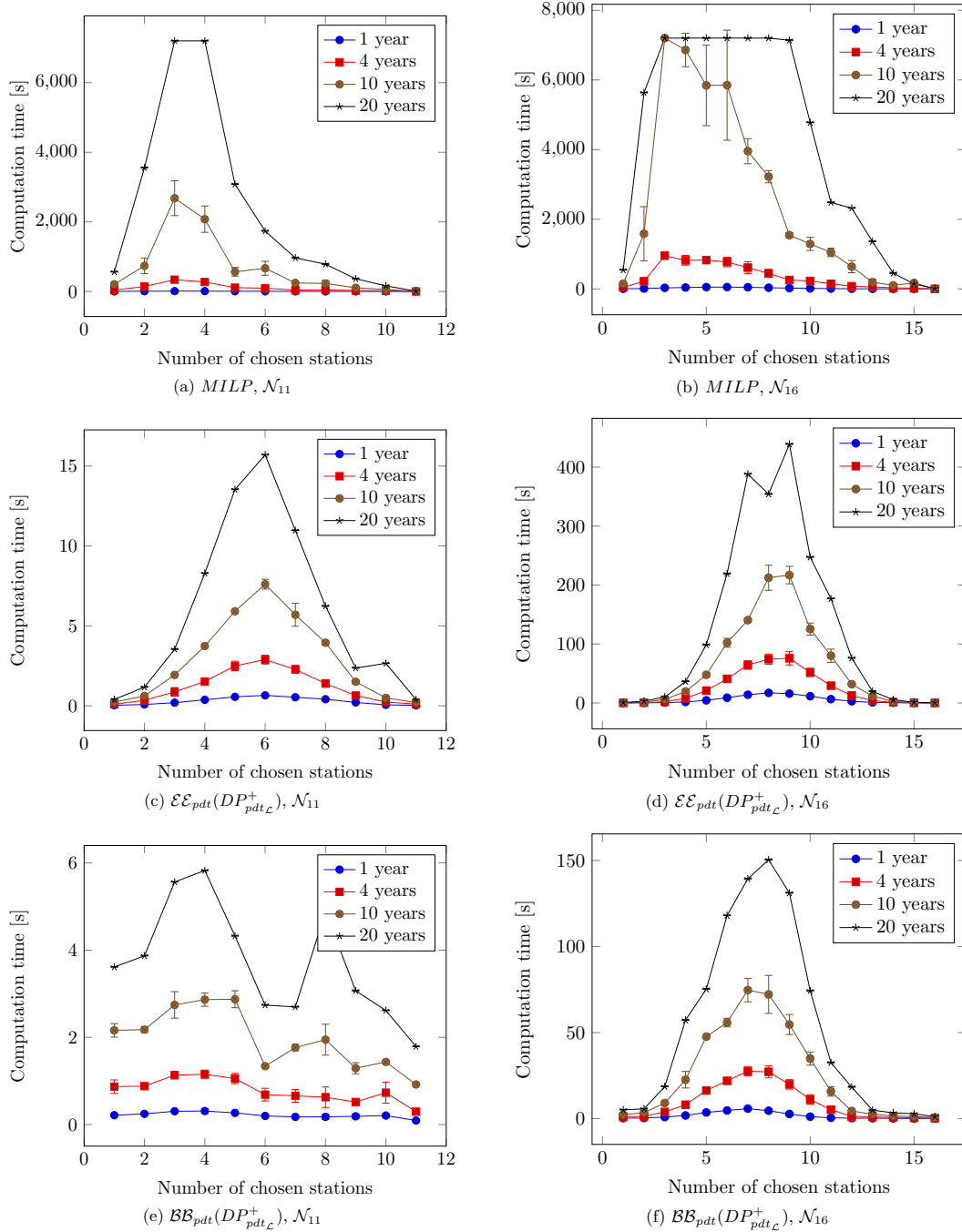


Figure 3.5 – Computation times for each methods using 8 threads, depending on the number of chosen stations.

is often faster than the complete enumeration.

3.3.3.3 Impact of multithreading on the performances

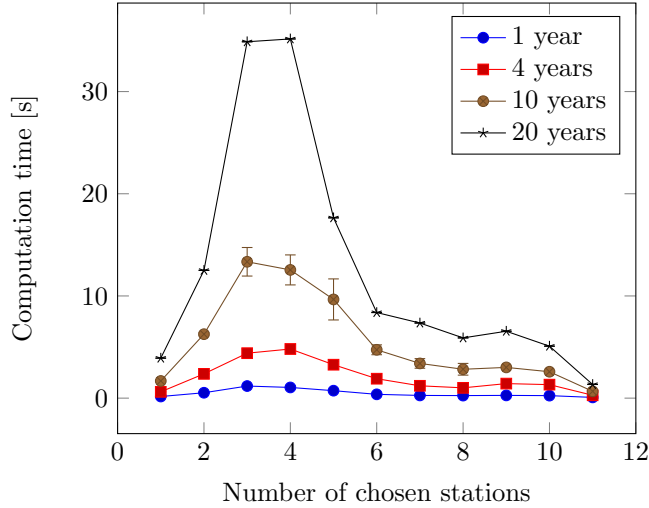


Figure 3.6 – Computation time for the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm using 1 thread for \mathcal{N}_{11} .

Figure 3.6 shows that the exponential behavior of the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm using one thread is similar to the behavior of the *MILP* solver using 8 threads for \mathcal{N}_{11} . The chaotic behavior observed in Figure 3.5e can be explained by the size of the instances — such small instances can depict strange behaviors with a multi-threaded algorithm.

Table 3.5 compares the computation times when one or eight threads are used for the *MILP* solver and both enumerations. Results are presented for various number of stations, for both the \mathcal{N}_{11} and \mathcal{N}_{48} sets and are averaged over all instances with a temporal horizon of 5 years. For each method, the two first columns represent the average solving time (in seconds) when using either one or eight threads, while the third column shows the speed up when using eight threads compared to using a single one. For some instances, the speed up is greater than the number of available threads. This can be explained by inconsistent loads of the execution platform between the runs or by the slight differences between the processors available on the platform. Since the goal here is only to compare the impact of multi-threading between the different approaches, these problems are of no consequences.

For complex instances (when K is close to half the number of available stations), the $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ clearly takes advantage of the 8 threads, while the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ only has a speed up close to 8 for the hardest instances. When the instances are simpler to solve, the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ fails to takes advantages of the threads compared to the $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$. This behavior is easily explained by the complexity of multi-threading a branch and bound compared to an exhaustive enumeration.

K	$\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$			$\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$			<i>MILP</i>			
	1 th.	8 th.	ratio	1 th.	8 th.	ratio	1 th.	8 th.	ratio	
\mathcal{N}_{11}	2	3.16	1.06	2.98	1.49	0.39	3.85	2135.00	200.56	10.65
\mathcal{N}_{11}	4	6.11	1.47	4.16	14.64	2.15	6.81	1871.52	453.79	4.12
\mathcal{N}_{11}	5	4.19	1.17	3.58	24.78	3.16	7.84	932.53	206.17	4.52
\mathcal{N}_{11}	6	2.03	0.77	2.65	28.71	3.30	8.70	291.33	118.65	2.46
\mathcal{N}_{11}	8	1.23	0.59	2.08	10.26	1.72	5.96	64.77	62.29	1.04
\mathcal{N}_{11}	10	1.16	0.70	1.65	0.89	0.31	2.84	28.12	23.66	1.19
\mathcal{N}_{16}	4	64.51	9.48	6.81	62.17	10.04	6.19	5593.90	1421.25	3.94
\mathcal{N}_{16}	6	187.33	28.92	6.48	373.87	55.66	6.72	4802.77	1392.55	3.45
\mathcal{N}_{16}	7	274.81	34.46	7.98	709.01	80.08	8.85	3521.69	1271.03	2.77
\mathcal{N}_{16}	8	226.76	33.60	6.75	636.93	98.31	6.48	2157.92	568.72	3.79
\mathcal{N}_{16}	9	144.35	23.92	6.03	843.58	85.47	9.87	1491.61	486.14	3.07
\mathcal{N}_{16}	12	7.72	1.73	4.47	108.35	16.25	6.67	328.47	108.56	3.03

Table 3.5 – Comparison of solving times between $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$, $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ and the MILP solver for instances with a temporal horizon of 5 years, on \mathcal{N}_{11} and \mathcal{N}_{16} when using one or eight threads.

In both enumerations, each run of $DP_{pdt_{\mathcal{L}}}^+$ is done on a separate thread, so eight instances of $\text{MaxPDT}_{\mathcal{L}}$ can be solved simultaneously, but the synchronization process is more complex for $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ than for $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$. While the *MILP* solver is also faster when using eight threads, the speed up is not close to the expected value of eight, meaning the relative gap between the *MILP* solver and our enumerations is greater when using eight threads.

3.3.3.4 Analysis of the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ results

Figure 3.7 shows the percentage of instances of \mathcal{N}_{48} for which the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ algorithm does not find the optimal solution, or fails to prove optimality.

With very small and high numbers of chosen stations ($K < 4$ or $K > 13$), this algorithm always finds the optimal solution, which is why these values are not shown in Figure 3.7. When the number of chosen stations is very small $K < 4$, the algorithm is likely able to enumerate all the networks, while with high number of chosen stations $K > 13$, the algorithm finds non-improvable solution (with a 100% percentage of data transferred) quite quickly, and thus prunes the whole search tree. Between these values, the algorithm has trouble finding optimal solutions, especially for large temporal horizons (10 and 20 years).

Figures 3.8a–3.8c show the number of nodes processed by $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ depending on the number of required stations for the three considered sets of loca-

3.3. Computational results

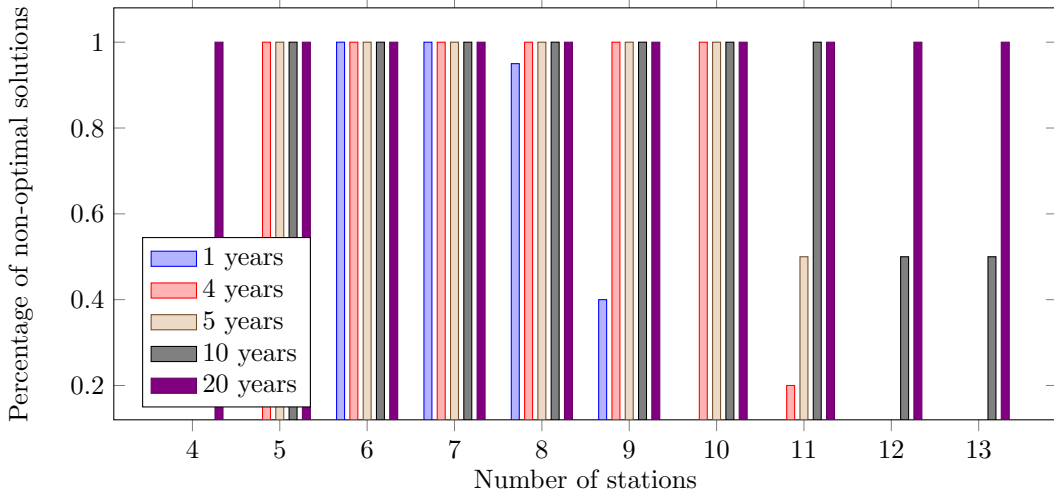


Figure 3.7 – Percentage of instances for which the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ did not find the optimal solution or failed to prove optimality for \mathcal{N}_{48} .

tions. For \mathcal{N}_{11} and \mathcal{N}_{16} , the numbers of possible networks are presented, and we can see that the number of nodes processed by $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ are much lower than the number of possible networks, even if in theory, the number of nodes processed could be slightly larger. For \mathcal{N}_{48} , the number of possible networks is orders of magnitudes larger than the number of nodes processed, e.g., the number of possible networks for 24 stations is $C_{24}^{48} \approx 3.10^{13}$, while the number of nodes processed never exceeds 2.10^6 . For \mathcal{N}_{11} and \mathcal{N}_{16} , the number of nodes processed seems not to be correlated with the size of the temporal horizon, which is somehow expected since $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ only branches on stations and not on download points. For \mathcal{N}_{48} , the number of nodes processed increases with smaller temporal horizons and smaller numbers of chosen stations ($3 < K < 14$). This is expected from the results in Figure 3.7 because for these numbers of required stations, the algorithm reaches the time limit. Thus, it is able to process more nodes for small temporal horizons since the time required to process a node is proportional to the length of the temporal horizon (due to the near-linear performance of the $DP_{pdt_{\mathcal{L}}}^+$ algorithm on realistic instances). When the number of chosen station is high (typically greater than 15), there are a lots of networks that reach the optimal percentage of data transferred (100%), which is why the numbers of nodes processed are relatively small. This can be visualized in Appendix B.3. When the number of chosen stations is greater than 20, the algorithm simply selects the K first locations.

Figures 3.9a–3.9c show the maximum percentage of data transferred for different number of chosen stations for the three considered sets of locations. For each set, there seems to be little improvement above $K = 7$ stations. With a temporal horizon \mathcal{H} of 20 years, the differences between using seven stations and the maximum possible number of stations for \mathcal{N}_{11} , \mathcal{N}_{16} and \mathcal{N}_{48} are respectively 0.64%, 0.4% and 0.23%. Choosing stations within \mathcal{N}_{11} , the percentage of data transferred

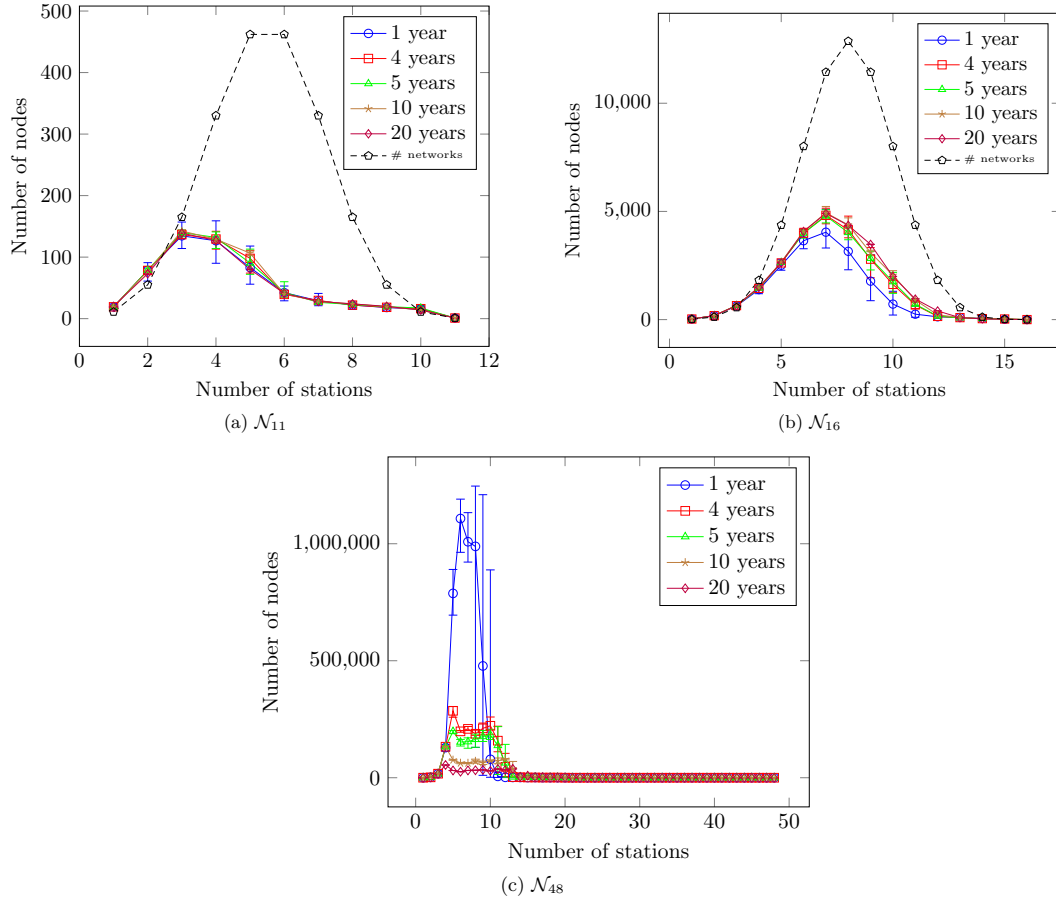


Figure 3.8 – Number of nodes evaluated by $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ using 8 threads, for each network, depending on the number of chosen stations.

is limited to 96%, while \mathcal{N}_{16} achieves a 99.99% percentage with 15 or 16 stations. Using the set \mathcal{N}_{48} , the algorithm manages to transfer 100% of the data with only 14 stations. Whatever the size of the temporal horizon \mathcal{H} , the mean percentage of data transferred is almost the same, but the variations are typically larger with few stations than with many. These variations seem to be less important for \mathcal{N}_{16} or \mathcal{N}_{48} than for \mathcal{N}_{11} . This can be explained by the greater flexibility in the choice of stations for larger sets of locations.

These results show that even if the number of stations has a strong impact on the percentage of data transferred, having good diversity in the locations of the stations allows for even better performance.

Summary

In this chapter, we considered the problem MaxPDT of designing networks of ground stations maximizing the amount of data transferred from a low-Earth orbiting satel-

3.3. Computational results

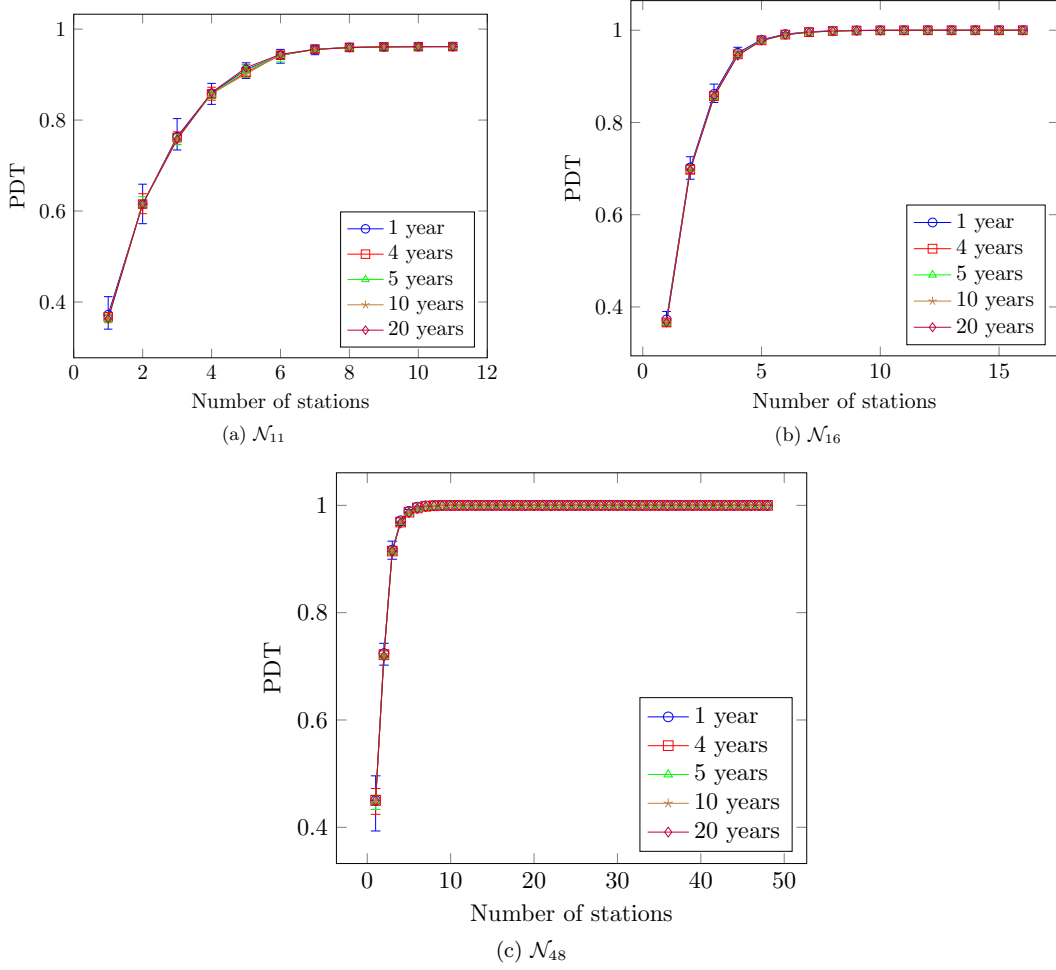


Figure 3.9 – Mean *Percent of Data Transferred* found for each network.

lite to the ground using optical communications. We modelled this problem using an aggregation of visibility windows into download points. Within MaxPDT , we analyze the download points selection problem, $\text{MaxPDT}_{\mathcal{L}}$, and gave complexity results regarding its general form. We proposed mixed-integer linear programs for both MaxPDT and $\text{MaxPDT}_{\mathcal{L}}$. To solve $\text{MaxPDT}_{\mathcal{L}}$, we proposed a dynamic programming algorithm $DP_{pdt_{\mathcal{L}}}^+$ that exhibits an FPT-behavior for realistic instances. We combine this algorithm with two enumeration methods, $\mathcal{EE}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ and $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$, in order to solve the MaxPDT problem. We showed that for realistic instances, both enumerations methods strictly outperform a commercial *MILP* solver and that the $\mathcal{BB}_{pdt}(DP_{pdt_{\mathcal{L}}}^+)$ is able to find very good and optimal solutions, even for large input sets of locations.

Flexible scheduling of acquisition downloads

Contents

4.1 Problem statement	72
4.1.1 Acquisition plans for Earth observation systems	72
4.1.2 Visibility windows between satellites and ground stations	73
4.1.3 Downlink data rates during visibility windows	74
4.2 Allocating download windows and assigning acquisitions	75
4.2.1 Computing the maximum download capacity	76
4.2.1.1 Computing capacities of a download windows	77
4.2.1.2 Mixed-Integer Linear Program for the MaxCap problem	78
4.2.1.3 Decomposition of the MaxCap problem	79
4.2.1.4 Taking uncertainties into account	80
4.2.2 Assigning acquisitions to download windows	80
4.2.2.1 Building a flexible plan	81
4.2.2.2 Mixed-Integer Linear Program for the GrdAsg problem	81
4.2.2.3 Complexity of the GrdAsg problem	82
4.2.3 Discussion on the validity of the approach	82
4.3 On-board scheduling of acquisition downloads	82
4.3.1 Greedy algorithm to create download schedules	84
4.3.2 Insertion of download tasks in the schedule	85
4.3.3 Ordering acquisitions for download	87
4.4 Experiments	88
4.4.1 Generation of instances	88
4.4.1.1 Creation of acquisition scenarios	88
4.4.1.2 Computation of visibility windows	90
4.4.2 Context and considered sets of experiments	93
4.4.2.1 Definition of policies and modifiers	93
4.4.2.2 Considered sets of experiments	96
4.4.3 Results and analysis	98
4.4.3.1 Global performances	98
4.4.3.2 Impact of the matching policy	100
4.4.3.3 Impact of the on-board parameters	100

In this chapter, we consider the problem of planning downloads of acquisitions from a low-Earth orbiting satellite. Given a static acquisition plan and a set of visibility windows between the satellite and a set of optical ground stations, our goal is to create a download schedule with maximum profit. In the context of free-space optical communications, clouds and atmospheric turbulence impact the communication links during visibility windows between the satellite and the stations, and create uncertainties regarding the capacities of these visibility windows. Furthermore, due to variable bit-rate on-board image compression [Camarero 2012], the actual volumes of images are not known until their acquisitions. Considering these two types of uncertainties, it would either be impossible or very limiting to only consider scheduling of downloads from the ground. Thus, we propose a two-step decision process where a first optimization is made on the ground, at regular interval, while the final schedule is created on-board using a custom greedy algorithm.

The chapter is organized as follows. Section 4.1 defines more formally the problem we consider. Section 4.2 presents the ground optimization procedure and two related problems, `MaxCap` and `GrdAsg0`, while Section 4.3 describes the on-board procedure that we use to create schedules. Section 4.4 presents results of simplified simulations regarding this problem on realistic scenarios.

4.1 Problem statement

We consider the problem of planning downloads of acquisitions over a given horizon $\mathcal{H} = [T_{start}, T_{end}]$. We assume that prior to the beginning of the horizon T_{start} , an acquisition plan has been uploaded to the satellite, and that this plan cannot be modified until T_{end} .

4.1.1 Acquisition plans for Earth observation systems

The acquisition plan corresponds to a set $\mathcal{A} = \{a_1, \dots, a_M\}$ of M acquisitions to be made. Each acquisition $a \in \mathcal{A}$ has a release date r_a (time at which the acquisition is available for download) and a due date d_a .

Acquisitions are in reality made of multiple files stored in various memory banks, but since we consider a single download channel (unlike previous works [Maillard 2014, Pralet 2014, Maillard 2015a]), there cannot be conflicts when reading data from memory banks, so we consider each acquisition as a single file. Due to operational constraints, each acquisition a must be downloaded during a single visibility window and cannot be split over multiple ones.

While the actual volume of an acquisition $a \in \mathcal{A}$ is uncertain, we have some information regarding its possible size. For each acquisition, we know its minimum q_a^{min} , maximum q_a^{max} , and expected q_a^{exp} volumes, respectively from the minimum accepted compression ratio, the maximum possible compression ratio and the expected compression ratio. Once the acquisition is fully acquired (at r_a), we know,

on-board, its actual volume q_a^{act} . By definition, we have:

$$q_a^{min} \leq q_a^{exp} \leq q_a^{max} \text{ and } q_a^{min} \leq q_a^{act} \leq q_a^{max}, \quad \forall a \in \mathcal{A}$$

A profit p_a is associated to each acquisition $a \in \mathcal{A}$. The profit of an acquisition is constant and does not depend on the age of the acquisition when downloaded. If the acquisition is downloaded before its deadline, the profit p_a is earned, otherwise no profit is made.

4.1.2 Visibility windows between satellites and ground stations

Much like in the previous problem, optical ground stations are visible from the satellite during visibility windows. Over the consider horizon, the satellite can download acquisitions using a set $V = \{v_1, \dots, v_N\}$ of N visibility windows. A visibility windows v starts at t_v^{start} and ends at t_v^{end} .

From now on, we will assume that the set V of visibility windows is sorted according to their start times:

$$\forall v_i, v_j \in V : i < j \iff t_{v_i}^{start} \leq t_{v_j}^{start}$$

Example 5. Figure 4.1 shows a basic representation of an example of the problem with 13 acquisitions and 9 visibility windows. In this example, some visibility windows are overlapping (e.g., v_1 and v_2). However, the satellite can only communicate with a single station at a time, and thus can only use one visibility window at a time.

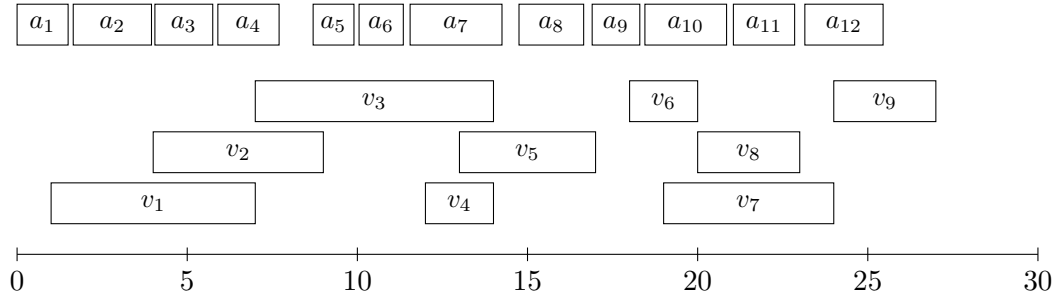





Figure 4.1 – Visual representation of an acquisition plan with multiple visibility windows.

Furthermore, in order to start downloading data to a station during a visibility window v , the satellite needs to initialize the communication. The time required to initialize a communication is complex to compute, mainly due to geometric aspects (e.g., position of the satellite when the initialization starts). In this work, we assume that this time, Δ , is constant and identical for all visibility windows. We also assume that the time required to switch from one visibility window to another is equal to the same constant Δ . Moreover, when switching from one visibility window v_i to

another v_j , the satellite cannot begin the initialisation before the start time $t_{v_j}^{start}$ of the second visibility window v_j . This means that, if a visibility window is used to download acquisitions, the time available for downloads during this visibility window is at most its duration minus Δ . We assume that once the satellite has switched from one visibility window to another, it cannot go back to the first one.

Taking these constraints into account, the first problem is to find which visibility windows should be used and for how long. This create an allocation of visibility windows to what we call **download windows**.

Example 6. Figure 4.2 shows an example of visibility window allocation corresponding to the example in Figure 4.1. Initialisation times are represented by  patterns. The  patterns represent portions of windows that are discarded (i.e., not used). The portion of the visibility windows that are actually used to download acquisitions are represented with  patterns. We call these portions **download windows** and since visibility windows cannot be interrupted and started again, each visibility window contains at most one download window.

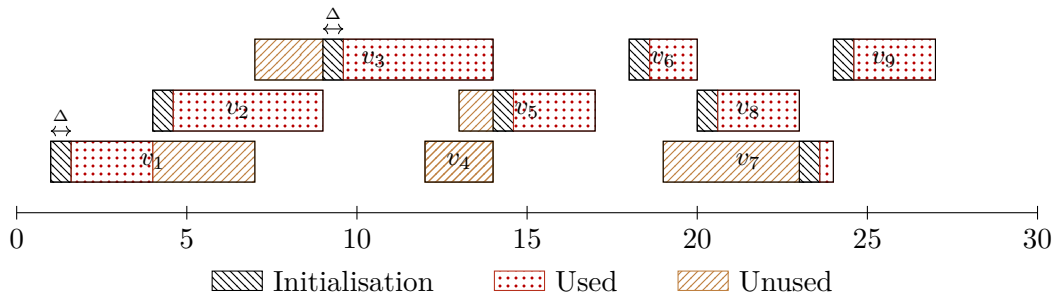


Figure 4.2 – Example of visibility window allocation.

4.1.3 Downlink data rates during visibility windows

We assume that the data rate during any visibility window can be viewed as piecewise constant function of the time, and that, since clouds are not exactly known before the start of the visibility window, this function cannot be exactly known before the actual visibility window.

Each visibility window v has its own data rate function f_v because the function depends on orbital information (angles between the satellite and the station during the visibility window) and atmospheric (clouds) information.

Formally, we define $f_v : [t_v^{start}, t_v^{end}] \rightarrow \mathbb{R}^+$ the actual data rate function of the visibility window v , i.e., the data rate function that accounts for the real clouds and atmospheric turbulence.

Example 7. Figure 4.3 shows two examples of data rate functions.

While the actual data rate function f_v of a visibility window v is not known before the start of this visibility window t_v^{start} , we can use forecasts to guess it.

4.2. Allocating download windows and assigning acquisitions

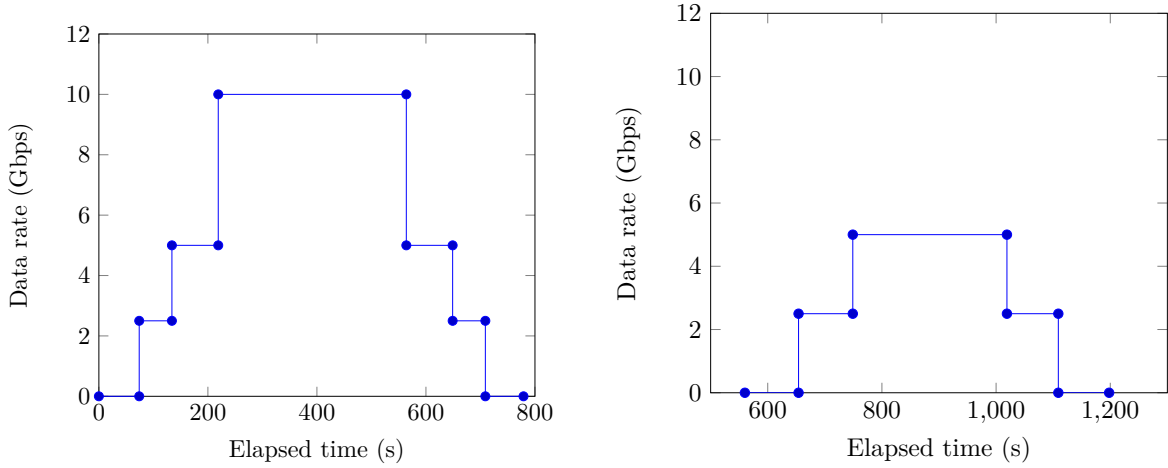


Figure 4.3 – Examples of two data rate functions f_v .

We denote by f_v^t the prediction for the data rate function f_v made at time t . We consider that predictions made at or after the beginning of a visibility window are perfect, i.e.:

$$\forall v \in V, \forall t \geq t_v^{start} : f_v^t = f_v$$

We assume that the predictions of the data rate functions are also piecewise constant functions. Due to inconsistency and volatility of forecasts, there is no clear relation between the successive predictions, or between the predictions and the actual data rate function.

4.2 Allocating download windows and assigning acquisitions

Considering the context defined in Section 4.1, our goal is to maximize the total earned profit. Profit is earned when an acquisition is downloaded before its deadline. Considering the many uncertainties related to the download link and to the volume of acquisitions, we decide to share the optimization process between the ground and the on-board system. The idea is to allocate download windows within visibility windows and create an assignment of acquisitions to download windows on the ground, and then to re-assign acquisitions and compute download schedules on-board.

This section is dedicated to the ground optimization procedures. The goal of these procedures is to allocate download windows within visibility windows, and create a first assignments of acquisitions to download windows.

4.2.1 Computing the maximum download capacity

We consider a problem where, given a set of visibility windows with various data rates, the goal is to allocate download windows within these visibility windows in order to maximize the amount of data that can be downloaded. This problem is denoted **MaxCap** problem.

Definition 12. Instance of the **MaxCap** problem

An instance (V, Δ) of the **MaxCap** problem consists of a set $V = \{v_1, \dots, v_N\}$ of visibility windows and a setup time Δ . A visibility windows v_i is associated with a start time $t_{v_i}^{start}$, an end time $t_{v_i}^{end}$ and a data rate function f_{v_i} . Each (forecast) data rate function f_{v_i} is a piecewise constant function over $[t_{v_i}^{start}, t_{v_i}^{end}]$:

$$f_{v_i} : [t_{v_i}^{start}, t_{v_i}^{end}] \rightarrow \mathbb{R}^+$$

The setup time Δ is the time needed to initialise communications between the satellite and the ground station.

Definition 13. Feasible solution for the **MaxCap** problem

Given an instance (V, Δ) of the **MaxCap** problem, a feasible solution is a set of download windows $\mathcal{W} = \{w_1, \dots, w_K\}$ ($K \leq N$), their respective start times $t_{w_i}^{start}$, end times $t_{w_i}^{end}$ and their associated visibility windows ν_{w_i} such that:

$$\forall w_i, w_j \in \mathcal{W} : \nu_{w_i} \neq \nu_{w_j} \quad (4.1a)$$

$$\forall w \in \mathcal{W} : [t_w^{start}, t_w^{end}] \subseteq [t_{\nu_w}^{start} + \Delta, t_{\nu_w}^{end}] \quad (4.1b)$$

$$\forall w_i, w_j \in \mathcal{W} : t_{w_i}^{end} + \Delta \leq t_{w_j}^{start} \text{ or } t_{w_j}^{end} + \Delta \leq t_{w_i}^{start} \quad (4.1c)$$

Equation (4.1a) ensures that each visibility window contains at most one download window. Equation (4.1b) guarantees that each download window is embedded inside a single visibility window and that there is enough time to initialize the communications before the start of any download window. Equation (4.1c) ensures that two download windows are not overlapping and respect the setup time.

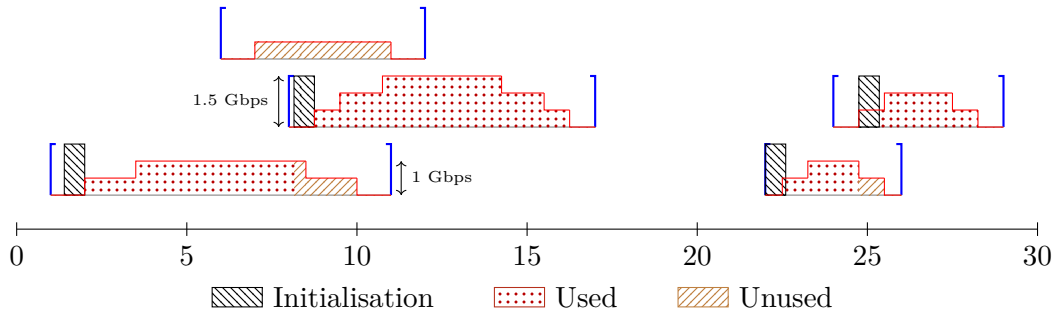








Figure 4.4 – Example of a feasible solution for the **MaxCap** problem.

Example 8. Figure 4.4 contains an example of a feasible solution for a **MaxCap** instance. Each download window is embedded inside a single visibility window de-

4.2. Allocating download windows and assigning acquisitions

limited by  patterns. Setup times are represented by  patterns and allocated portions of visibility windows to download windows by  patterns. The  patterns represent part of visibility windows that are not allocated and thus not used. Download windows are delimited on the left by setup times ( patterns) and on the right by ends of visibility windows or non-allocated portions ( patterns).

Definition 14. Capacity of a solution for the **MaxCap** problem

Given a solution (\mathcal{W}) for the **MaxCap** problem, the associated capacity is the total volume that can be downloaded using the set \mathcal{W} of download windows:

$$cap(\mathcal{W}) = \sum_{w \in \mathcal{W}} \int_{t_w^{start}}^{t_w^{end}} f_{\nu_w}(t) dt \quad (4.2)$$

The objective of the **MaxCap** problem is to find a feasible solution (\mathcal{W}) of maximal capacity $cap(\mathcal{W})$.

4.2.1.1 Computing capacities of a download windows

Given an instance (V, Δ) of the **MaxCap** problem, we define, for each visibility window $v \in V$, F_v , the antiderivative of f_v , as follows:

$$\begin{aligned} F_v(t) &: [t_v^{start}, t_v^{end}] \rightarrow \mathbb{R}^+ \\ t &\rightarrow \int_{t_v^{start}}^t f_v(x) dx \end{aligned} \quad (4.3)$$

Since f_v is a piecewise constant function on $[t_v^{start}, t_v^{end}]$, F_v is a piecewise linear and continuous function on $[t_v^{start}, t_v^{end}]$. By definition, $F_v(t_v^{start}) = 0$.

For each visibility window v , we then define the volume function if v , g_v , as follows:

$$g_v(t) = \begin{cases} 0 & \text{if } t < t_v^{start} \\ F_v(t) & \text{if } t \in [t_v^{start}, t_v^{end}] \\ F_v(t_v^{end}) & \text{if } t > t_v^{end} \end{cases} \quad (4.4)$$

The volume function g_v represents the amount of data that could have been downloaded so far using v . Since F_v is a piecewise linear and continuous function on $[t_v^{start}, t_v^{end}]$, and since $F_v(t_v^{start}) = 0$, g_v is a piecewise linear and continuous function on the temporal horizon \mathcal{H} . Figure 4.5 shows the volume functions associated to the data rate functions shown in Figure 4.3.

Example 9. Figure 4.6 shows a visual representation using volume functions corresponding to the example in Figure 4.4.

Using these volume functions, we can compute the capacity ρ_w of each download window w :

$$\rho_w = g_{\nu_w}(t_w^{end}) - g_{\nu_w}(t_w^{start}) \quad (4.5)$$

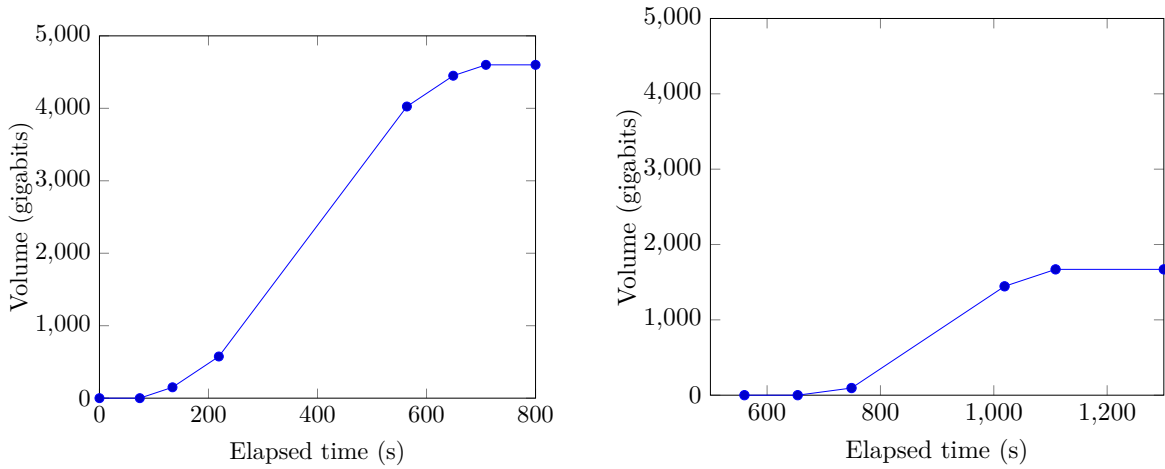


Figure 4.5 – Examples of two volumes functions g_v .

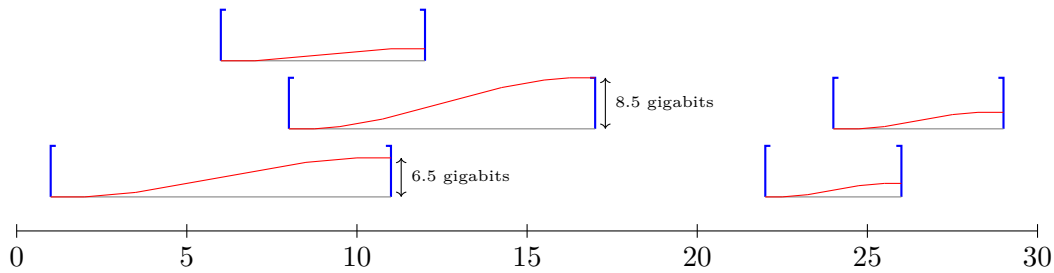


Figure 4.6 – Example of volume functions g_v for a MaxCap instance.

Finally, the capacity $cap(\mathcal{W})$ of a solution of the MaxCap problem is simply the sum of the capacity of the download windows in \mathcal{W} :

$$cap(\mathcal{W}) = \sum_{w \in \mathcal{W}} \rho_w \tag{4.6}$$

4.2.1.2 Mixed-Integer Linear Program for the MaxCap problem

To model this problem, we first consider two sets of real variables $t_v^s \geq 0$ and $t_v^e \geq 0$ representing the time at which the communication starts (after the initialisation) and ends during a visibility window v . If $t_v^s = t_v^e$, the visibility window v is not used. We also consider two sets of real variables c_v^s and c_v^e corresponding to the values of g_v at t_v^s and t_v^e . Finally, we consider a set of binary variables y_v , with $y_v = 1$ if and only if the visibility window v is used, i.e., $t_v^e > t_v^s$.

Using these sets of variables and the parameters given in Definition 12 (summarized in Table 4.1), the mathematical model for the MaxCap problem can be

4.2. Allocating download windows and assigning acquisitions

formulated as follows:

$$\max. \quad \sum_{v \in V} (c_v^e - c_v^s) \quad (4.7a)$$

$$\text{s.t.} \quad t_v^s \leq t_v^e, \quad v \in V \quad (4.7b)$$

$$t_v^s \geq y_v * (t_v^{start} + \Delta), \quad v \in V \quad (4.7c)$$

$$c_v^s = g_v(t_v^s), \quad v \in V \quad (4.7d)$$

$$c_v^e = g_v(t_v^e), \quad v \in V \quad (4.7e)$$

$$t_v^e - t_v^s \leq (t_v^{end} - t_v^{start})y_v, \quad v \in V \quad (4.7f)$$

$$t_{v_j}^s \geq t_{v_i}^e + \Delta y_{v_i}, \quad \forall v_i, v_j \in V, j > i \quad (4.7g)$$

$$y_v \in \{0, 1\}, \quad v \in V \quad (4.7h)$$

$$t_v^s, t_v^e, c_v^s, c_v^e \geq 0, \quad v \in V \quad (4.7i)$$

Objective (4.7a) is to maximize the total download capacity (see Equations (4.5–4.6)). Constraints (4.7b) indicate that the communication must end after its start. Constraints (4.7d–4.7e) link the starting and ending time of the communication to their respective cumulated volume. We know that for each $v \in V$, g_v is a piecewise linear function, thus constraints (4.7d–4.7e) can be modeled using special order set of type 2 (SOS2) [Beale 1969, Keha 2004]. Constraints (4.7f) force y_v to take a non-zero value if the window is used, i.e., if $t_v^s \neq t_v^e$. Constraints (4.7g) enforce that if visibility window v_i is used (y_{v_i}), no visibility windows can be used before $t_{v_i}^e + \Delta$.

In the above model, constraints (4.7g) apply to all pair of visibility windows. We can reduce the number of constraints by only creating constraint for overlapping visibility windows (taking into account the time Δ required to switch from one window to another). Formally, we only need to create constraints for pair of windows $v_i, v_j \in V, j > i$ such that:

$$t_{v_j}^{start} < t_{v_i}^{end} + \Delta$$

In order to obtain download windows from a solution of the above mathematical program, we need to look at the values of the t_v^s and t_v^e variables. For each visibility window v , we allocate a download window w within v if $t_v^s \neq t_v^e$ with $t_w^{start} = t_v^s$, $t_w^{end} = t_v^e$ and $\nu_w = v$. If $t_v^s = t_v^e$, the visibility windows is not used, and no download window is allocated within it.

4.2.1.3 Decomposition of the MaxCap problem

In real instances, there are few overlaps, and overlapping visibility windows are often found in small groups. In order to speed up the resolution of **MaxCap**, we propose to solve the problem by splitting it into smaller disjoint problems.

Definition 15. Let V_i and V_j be two subsets of V . We said that V_i and V_j are **unrelated** if V_i and V_j are disjoint and no window in V_i overlaps with a window

in V_j :

$$\forall v_k \in V_i, v_l \in V_j : t_{v_l}^{start} < t_{v_k}^{end} + \Delta \text{ or } t_{v_k}^{start} < t_{v_l}^{end} + \Delta \quad (4.8)$$

Proposition 4. *Let $U = \{V_1, \dots, V_K\}$ be a set of **pairwise unrelated** sets of visibility windows. For each $V_i \in U$, let \mathcal{W}_i^* be the optimal solution of the **MaxCap** instance (V_i, Δ) . Then, the set $\mathcal{W}^* = \bigcup_{i=1}^K \mathcal{W}_i^*$ of download windows is the optimal solution for the **MaxCap** instance $(\bigcup_{i=1}^K V_i, \Delta)$.*

Proof. Let $U = \{V_1, \dots, V_K\}$ be a set of **pairwise unrelated** sets of visibility windows. For each $V_i \in U$, let \mathcal{W}_i^* be the optimal solution of each **MaxCap** instance (V_i, Δ) . Let $\mathcal{W} = \bigcup_{i=1}^K \mathcal{W}_i^*$ be a solution to the **MaxCap** instance $(\bigcup_{i=1}^K V_i, \Delta)$.

Since for all $1 \leq i \leq K$, \mathcal{W}_i is a feasible solution for the (V_i, Δ) instance, we know that Equation (4.1a) is satisfied for any download window w in \mathcal{W} . Furthermore, since the sets V_i ($1 \leq i \leq K$) are unrelated, Equation (4.1b) is also satisfied. Using Equation (4.1a), we know that any two download windows in two different sets are also separated by at least Δ , so Equation (4.1c) is also satisfied. Since Equations (4.1b–4.1c) are satisfied, \mathcal{W} is a feasible solution to the **MaxCap** instance $(\bigcup_{i=1}^K V_i, \Delta)$.

We are going to show that if \mathcal{W} is not the optimal solution, then at least one of the \mathcal{W}_i^* solution was not optimal, which is a contradiction. If \mathcal{W} is not optimal, then either there is at least one download w_p that can be modified to increase its capacity, or there exists a visibility window v_q within which a new download window can be allocated (or both). Let us assume that the modified or newly created download window belongs to the solution \mathcal{W}_u^* of the (V_u, Δ) . Since the instances (V_i, Δ) are independent (due to the unrelated sets of visibility windows), modifications on the \mathcal{W}_u^* solution do not impact other solutions. Let us assume there exists a solution \mathcal{W}_u^* within which we can add or extend a download window to increase the objective value of the \mathcal{W} solution. Since this change does not impact other solutions \mathcal{W}_i^* ($i \neq u$), their capacities remain the same, and thus the increase in capacity of \mathcal{W} comes only from an increase in capacity of \mathcal{W}_u^* , so \mathcal{W}_u^* was not optimal, a contradiction. \square

4.2.1.4 Taking uncertainties into account

When computing download windows on the ground, we do not have access to the real data rate functions of visibility windows, so we need to use the forecast functions. Assuming the optimization process is run at time t , we will use the forecasts made at time t for the visibility windows f_v^t .

4.2.2 Assigning acquisitions to download windows

Once download windows have been computed, we need to assign acquisitions to them. In a deterministic context, we could consider downloads of acquisitions as tasks having variable duration depending on the size of acquisitions and the download data rates and try to schedule these tasks within download windows (that can be seen as resources).

4.2. Allocating download windows and assigning acquisitions

Doing so would create very large and complex scheduling problems due to the precision required for the duration of the tasks (smaller than a second for the considered data rates) compared to the extent of the temporal horizon (several hours). Furthermore, we are actually facing multiple sources of uncertainties, and by solving the above mentioned scheduling problem, we would likely create plans that would become infeasible once uncertainties are resolved on-board.

To cope with these problems, we choose to use a **flexible** approach where we only pre-assign acquisitions to download windows on the ground without precisely scheduling their downloads. This way, we only upload an assignment of acquisitions to download windows to the satellite, and an on-board procedure is used to schedule the downloads. We call this problem the ground assignment problem, **GrdAsg**.

4.2.2.1 Building a flexible plan

In order to assign acquisitions to download windows, we have to decide to which download windows an acquisition can be downloaded. We denote by $\mathcal{W}[a]$ the set of download windows to which acquisition a can be assigned. Aside strict constraints due to political issue (e.g., some acquisitions can only be download using a subset of the available stations), we have to take into account the release and due dates of acquisitions. Since we only consider a packing problem, we have to create matching policies indicating to which download windows each acquisition can be downloaded. For instance, we have to decide if we allow the download of an acquisition a using a download window w if the acquisition a is released during w . We propose in Section 4.4 two matching policies called **soft** and **hard**.

Furthermore, in order to prioritize acquisitions with high profits and to avoid downloading acquisitions at the last minute, we propose to use modified volumes \bar{q}_a and modified profits \bar{p}_{aw} for acquisitions. The way volumes and profits are modified is a parameter of the **GrdAsg** problem. We propose multiple modifiers for both volumes and profits in Section 4.4.2.1.

4.2.2.2 Mixed-Integer Linear Program for the GrdAsg problem

To model the problem, we consider a set of binary variables x_{aw} equal to one if and only if acquisition a is assigned to download window w .

$$\max. \quad \sum_{a \in \mathcal{A}} \sum_{w \in \mathcal{W}} \bar{p}_{aw} x_{aw} \quad (4.9a)$$

$$\text{s.t.} \quad \sum_{w \in \mathcal{W}} x_{aw} \leq 1, \quad a \in \mathcal{A} \quad (4.9b)$$

$$\sum_{a \in \mathcal{A}} \bar{q}_a x_{aw} \leq \rho_w, \quad w \in \mathcal{W} \quad (4.9c)$$

$$x_{aw} = 0, \quad a \in \mathcal{A}, w \notin \mathcal{W}[a] \quad (4.9d)$$

$$x_{aw} \in \{0, 1\}, \quad a \in \mathcal{A}, w \in \mathcal{W} \quad (4.9e)$$

Objective (4.9a) maximizes the sum of modified profits of assigned acquisitions. Constraints (4.9b) ensure that each acquisition is assigned to at most one download window. Constraints (4.9c) ensure that the capacity ρ_w of each download window is not exceeded. Constraints (4.9d) ensure that acquisitions are only assigned to allowed download windows.

4.2.2.3 Complexity of the GrdAsg problem

This problem is a generalization of the Multiple Knapsack Problem (MKP) where items can only be assigned to a subset of the available knapsacks. Since MKP is a strongly NP-hard optimization problem [Kellerer 2004], the considered ground assignment problem is also strongly NP-hard.

4.2.3 Discussion on the validity of the approach

The approach used in the MaxCap problem is to consider the capacity of a download window as the integral of the data rate functions over the interval of definition of the download window, without taking into account the assignment of acquisitions to download windows.

This approach has the advantage of being simple. By not taking complex constraints into account, the problem reduces to a simple maximization of the capacity in MaxCap, followed by a packing approach in GrdAsg.

However, this approach has some drawbacks. By maximizing the capacity without taking the acquisition plan into account, some non-optimal choice can be made, e.g., by switching from one visibility window to another too soon and not being able to download an acquisition that could only be downloaded during the first visibility window due to its deadline.

The GrdAsg problem also suffers from two drawbacks making most assignments infeasible. First, depending on the considered matching policy, some acquisitions can be assigned to download windows incorrectly, e.g., if the time between the release of an acquisition and the end of a download window is not sufficient to download the acquisition. Secondly, when downloading an acquisition, the data rate cannot be changed, thus the actual usable capacities of download windows are often slightly less than the capacities computed using the MaxCap approach. This is why an on-board algorithm is in charge of creating the final download schedule prior to any download windows.

4.3 On-board scheduling of acquisition downloads

On-board, we have access to the flexible plan uploaded after the last ground optimization, which means that we know the allocation of download windows, i.e., when we can start communicating with ground stations, and which acquisitions have been assigned to which download window.

4.3. On-board scheduling of acquisition downloads

Inputs of the problem	
T_{start} Start of the temporal horizon.	T_{end} End of the temporal horizon.
\mathcal{A} Set of acquisitions.	M Number of acquisitions.
r_a Release date of acquisition a .	d_a Deadline of acquisition a .
p_a Profit of acquisition a .	
q_a^{act} Real volume of acquisition a .	q_a^{exp} Expected volume for acquisition a .
q_a^{min} Minimum volume for acquisition a .	q_a^{max} Maximum volume of acquisition a .
V Set of visibility windows.	N Number of visibility windows.
t_v^{start} Start of visibility window v .	t_v^{end} End of visibility window v .
Δ Setup time.	f_v Data rate function of visibility window v .
f_v^t Forecast data rate function of visibility window v .	g_v Volume function of visibility window v (4.4).
\mathcal{W} Set of download windows.	ν_w Visibility window containing w .
t_w^{start} Start of download window w .	t_w^{end} End of download window w .
ρ_w Capacity of download window w .	
Decision variables	
t_v^s Time to start using visibility window v .	t_v^e Time to stop using visibility window v .
c_v^s Value of the volume function of v at t_v^s .	c_v^e Value of the volume function of v at t_v^e .
y_v $y_v = 1$ if and only if visibility window v is used.	
x_{aw} $x_{aw} = 1$ if and only if acquisition a is assigned to download window w	

Table 4.1 – Summary of notations for the definition of the MaxCap and GrdAsg problems.

Due to the uncertainties present while creating the flexible plan on the ground, it may not be possible to execute the plan as it is but it may serve as an input for the on-board procedure described in this section. Once computed on the ground, the download windows cannot be modified, thus the on-board procedure is only used to re-assign acquisitions, if required, and to schedule their downloads within download windows.

This procedure is run at the beginning of each download window. At this point, we assume that we know the exact clouds and turbulence above the station corresponding to the download window, and we also know the actual sizes of acquisitions that have already been acquired.

Information regarding clouds and turbulence can be obtained, e.g., by the use of a small uplink that would be integrated to the beacon used to initialize the communication.

4.3.1 Greedy algorithm to create download schedules

The heuristic used on-board is based on a greedy algorithm that starts with an empty plan and tries to insert acquisitions in a specific order in this plan.

Before each download window w , we start by filtering out acquisitions that cannot be downloaded this download window, i.e., acquisitions that have already been downloaded, acquisitions that are not allowed to be downloaded during the current download window (hard constraint due, e.g., to political or business policies) or acquisitions whose deadlines are before the start of w or whose release dates are after the end of w . This filtering procedure is referred as `FILTER` in the following. We then sort filtered acquisitions using a custom ordering between acquisitions. We discuss possible orderings in Section 4.3.3 based on the assignments made on the ground using `GrdAsg` and profits of acquisitions. We refer to this procedure as `SORT` in the following:

$$\mathcal{A}^S = \text{SORT}(w, \text{FILTER}(w, \mathcal{A}))$$

Once acquisitions have been filtered and sorted, we create the download schedule by inserting acquisitions one by one, in the order defined by the `SORT` procedure, as soon as possible in the current plan. In order to compute the time required to download an acquisition, we consider actual volumes q^{act} for acquisitions that have already been acquired and maximum volumes q^{max} for acquisitions to be acquired during the download window.

In order to create the download schedule, we consider a list of tasks \mathcal{T} initially containing two fictive tasks corresponding to the start and the end of a window. A download task is represented by a 3-tuple and is denoted $task(a, s, e)$ where a is the corresponding acquisition (or *null* for the fictive tasks) and s and e are respectively the start and the end of the download task. We use $\mathcal{T}[i]$ to represent the i^{th} task in \mathcal{T} . Given $\mathcal{T}[i]$, we use $start(\mathcal{T}[i])$ and $end(\mathcal{T}[i])$ to access the start and the end of the task. The `TRYINSERTBETWEEN` procedure tries to create and insert a download

4.3. On-board scheduling of acquisition downloads

task for a given acquisition between two time points in the current download plan. This procedure is defined in Algorithm 4.

We consider that the following procedures are available and run in constant time ($O(1)$): $head(\mathcal{T})$ to retrieve the first element (head) of the list, $next(\mathcal{T}, atask)$ to retrieve the task right after $atask$ in \mathcal{T} and $insertBefore(\mathcal{T}, atask_1, atask_2)$ to insert $atask_2$ before $atask_1$ in \mathcal{T} .

Algorithm 3 shows a formal description of the on-board scheduling algorithm. We start by filtering and sorting the acquisitions in \mathcal{A} and creating an initial plan containing two fictive tasks corresponding to the beginning and the end of the download window. Then, we try to insert each acquisition $a \in \mathcal{A}^S$ in the plan one after the other. For each acquisition, we retrieve its volume vol (lines 5–9), considering maximum volume for acquisition that are yet to be released. Then, for each two consecutive tasks in the current plan ($curr$ and $next$), we try to create a new download task for acquisition a between the end of $curr$ and the start of $next$ (line 15). If the insertion is possible ($res \neq \emptyset$), we insert this task between $curr$ and $next$ (lines 17–18), otherwise we check the next consecutive pair of download tasks. Finally, if we did not manage to insert a download task for a in the current plan, we reschedule a by putting it in the list of unaffected tasks for future download windows (lines 23–25).

4.3.2 Insertion of download tasks in the schedule

The TRYINSERTBETWEEN procedure takes a download window w , an acquisition a , a volume q , and an interval $[r, d]$, and tries to create a download task for a (assuming the volume of a is q) within the interval $[r, d]$.

We consider in this procedure the data rate function f_w of w , which is a restriction of the data rate function f_v of the visibility window v containing w . This data rate function is piecewise constant and can thus be expressed in the following way:

$$f_w(t) = \sum_{k=1}^{K_w} \gamma_w^k \mathbb{1}_{[\alpha_w^k, \beta_w^k]}(t) \quad (4.10)$$

where K_w is the number of steps in f_w , γ_w^k is the constant value of the k^{th} step, and $\mathbb{1}_{[\alpha_w^k, \beta_w^k]}$ is the indicator function over the interval $[\alpha_w^k, \beta_w^k]$ defining the k^{th} step, i.e.:

$$\mathbb{1}_{[\alpha_w^k, \beta_w^k]}(t) = \begin{cases} 1 & \text{if } t \in [\alpha_w^k, \beta_w^k] \\ 0 & \text{otherwise} \end{cases} \quad (4.11)$$

We assume that $\forall k \in K_w, \gamma_w^k$ is strictly greater than zero. Steps with null data rate are only found at the beginning and the end of visibility windows (when the elevation angle is too small to transmit data), and thus can be removed from the data rate functions corresponding to download windows.

Algorithm 4 shows a formal description of the TRYINSERTBETWEEN procedure. The goal is to find the first piece of the data rate function f_w that can be used to

Algorithm 3 On-board procedure that schedules acquisition downloads before download windows.

```

1: procedure ONBOARDSCHEDULING( $w, \mathcal{A}$ )
2:    $\mathcal{A}^S \leftarrow \text{SORT}(w, \text{FILTER}(w, \mathcal{A}))$ 
3:    $\mathcal{T} \leftarrow \{ \text{task}(\text{null}, t_v^{\text{start}}, t_v^{\text{start}}), \text{task}(\text{null}, t_v^{\text{end}}, t_v^{\text{end}}) \}$ 
4:   for each  $a \in \mathcal{A}^S$  do
5:     if  $r_a < t_w^{\text{start}}$  then
6:        $vol \leftarrow q_a^{\text{act}}$ 
7:     else
8:        $vol \leftarrow q_a^{\text{max}}$ 
9:     end if
10:     $inserted \leftarrow false$ 
11:     $curr \leftarrow \text{head}(\mathcal{T})$ 
12:    while  $curr \neq \emptyset$  and not  $inserted$  do
13:       $next \leftarrow \text{next}(\mathcal{T}, curr)$ 
14:      if  $\text{start}(next) \geq r_a$  then
15:         $res \leftarrow \text{TRYINSERTBETWEEN}(w, a, vol, \text{end}(curr), \text{start}(next))$ 
16:        if  $res \neq \emptyset$  then
17:           $\text{insertBefore}(\mathcal{T}, next, res)$ 
18:           $inserted \leftarrow true$ 
19:        end if
20:         $curr \leftarrow next$ 
21:      end if
22:    end while
23:    if not  $inserted$  then
24:       $\text{RESCHEDULE}(a)$ 
25:    end if
26:  end for
27: end procedure

```

download a between r and d , i.e., a piece that overlaps the interval $[r, d]$ and that is sufficiently large to download a .

We start by reducing, if necessary, the interval $[r, d]$ to take into account the release date r_a and the deadline d_a of a : we want $[r, d]$ to be a subset of $[r_a, d_a]$. Then, for each piece of f_w overlapping $[r, d]$, we compute the earliest download time, $r_k = \max(r, \alpha_w^k)$ and the associated download duration u_k (lines 6–7). If the download end time ($r_k + u_k$) exceeds the current piece, we have to compute a new duration taking into account the data rate of the next piece (line 12) because the data rate cannot be modified during a download. If there are no remaining pieces in f_w , it is not possible to insert a download for a between r and d and we return \emptyset to inform the main scheduling procedure (line 10). Once we have computed the actual duration u_k , we check if the download end time exceeds d , and if it does not, we insert the task at r_k (lines 14–15). Finally, if we did not find a viable piece, we return a null-task (line 19).

4.3. On-board scheduling of acquisition downloads

Algorithm 4 On-board procedure that tries to insert acquisition a within the interval $[r, d]$ inside the download window w .

```

1: procedure TRYINSERTBETWEEN( $w, a, q, r, d$ )
2:    $r \leftarrow \max(r_a, r)$ 
3:    $d \leftarrow \min(d, d_a)$ 
4:   for  $k = 1, \dots, K_w$  do
5:     if  $\alpha_w^k < d$  and  $\beta_w^k > r$  then
6:        $r_k \leftarrow \max(r, \alpha_w^k)$ 
7:        $u_k \leftarrow \frac{q}{\gamma_w^k}$ 
8:       if  $r_k + u_k \geq \beta_w^k$  then
9:         if  $k = K_w$  then
10:          return  $\emptyset$ 
11:        end if
12:         $u_k = \max(u_k, \frac{q}{\gamma_w^{k+1}})$ 
13:      end if
14:      if  $r_k + u_k < d$  then
15:        return  $\text{task}(a, r_k, r_k + u_k)$ 
16:      end if
17:    end if
18:  end for
19:  return  $\emptyset$ 
20: end procedure

```

4.3.3 Ordering acquisitions for download

When considering a given download window w , the set of acquisitions considered for download, \mathcal{A}_w , can be partitioned into three subsets \mathcal{A}_w^A , \mathcal{A}_w^F and \mathcal{A}_w^U that respectively corresponds to acquisitions assigned to w , assigned to future windows or not assigned by the ground assignment procedure, **GrdAsg**. Since these three sets form a partition of \mathcal{A}_w , we have:

$$\begin{aligned} \mathcal{A}_w &= \mathcal{A}_w^A \cup \mathcal{A}_w^U \cup \mathcal{A}_w^F \\ \mathcal{A}_w^A \cap \mathcal{A}_w^U &= \emptyset, \quad \mathcal{A}_w^U \cap \mathcal{A}_w^F = \emptyset, \quad \mathcal{A}_w^F \cap \mathcal{A}_w^A = \emptyset \end{aligned}$$

If **GrdAsg** has not been run on the ground, all acquisitions are unassigned, i.e.:

$$\mathcal{A}_w = \mathcal{A}_w^U$$

In order to sort acquisitions, we first assign priorities to them depending on the subset of \mathcal{A}_w in which they belong. Then, for each group of priority, we sort acquisitions based on a potentially biased profit or on a ratio between this profit and the volume of acquisitions.

The notation we use for the parameters of the sorting procedure is:

$$(P, W_A, W_{UF}, R)$$

Chapter 4. Flexible scheduling of acquisition downloads

where P indicates the priority between the three subsets of acquisitions.

The possible values for P are:

- **PrtA**: All acquisitions have the same priority.
- **PrtB**: Acquisitions in \mathcal{A}_w^A have priority over other acquisitions and acquisitions in \mathcal{A}_w^U have priority over acquisitions in \mathcal{A}_w^F .
- **PrtC**: Acquisitions in \mathcal{A}_w^A have priority over other acquisitions but acquisitions in \mathcal{A}_w^U and \mathcal{A}_w^F have the same priority.

Inside a priority group, acquisitions are sorted based on modified profits. These modified profits are computed depending on the W_A and W_{UF} parameters, that are similar to the ones used in **GrdAsg** (see Section 4.4.2.1). W_A is the profit modifier for acquisitions assigned to the current download windows (in \mathcal{A}_w^A), and W_{UF} is the profit modifier for acquisitions non-assigned or assigned to future windows (in $\mathcal{A}_w^U \cup \mathcal{A}_w^F$). Multiple profit modifiers are defined in Section 4.4.

Finally, R indicates if acquisitions should be sorted using their modified profits or the ratios between their modified profits and their volumes.

Example 10. Consider six acquisitions $a_1, a_2 \in \mathcal{A}_w^A$, $a_3, a_4 \in \mathcal{A}_w^U$ and $a_5, a_6 \in \mathcal{A}_w^F$ with respective profits 7, 3, 5, 4, 6, 8 and volumes 3, 2, 3, 4, 2, 5. Let us consider non-modified profit, then the ratios between profits and the volumes are respectively $\frac{7}{3}$, 1.5, $\frac{5}{3}$, 1, 3 and 1.6. Table 4.2 shows the different ordering of a_1 to a_6 for the possible type of priorities P , taking the volume into account or not (R).

R	P	Order of acquisitions (left is best)
false	PrtA	a_6 a_1 a_5 a_3 a_4 a_2
	PrtB	a_1 a_2 a_3 a_4 a_6 a_5
	PrtC	a_1 a_2 a_6 a_5 a_3 a_4
true	PrtA	a_5 a_1 a_3 a_6 a_2 a_4
	PrtB	a_1 a_2 a_3 a_4 a_5 a_6
	PrtC	a_1 a_2 a_5 a_3 a_6 a_4

Table 4.2 – An example of ordering of acquisitions for different type of priorities.

4.4 Experiments

4.4.1 Generation of instances

4.4.1.1 Creation of acquisition scenarios

We generate acquisition plans for one day using an industrial tool which ensures that the plan is valid, i.e., that there are no overlapping acquisitions and the plan

comply with the agility of the satellite.

The generated plans only contain, for each acquisition $a \in \mathcal{A}$, its end time r_a and its duration. We generate deadlines d_a for acquisitions by adding a uniformly drawn duration between 30 minutes and 2 hours after the release dates of acquisitions. For each acquisition $a \in \mathcal{A}$, we generate profits p_a randomly using a uniform distribution between 100 and 500.

We generate non-compressed acquisition volumes based on the duration of acquisitions, assuming an acquisition rate of 9134.5 megabits-per-second, which is realistic for future Earth observation missions. We consider that the non-compressed images are encoded with 12 bits-per-pixel, that the minimum and maximum accepted compression ratios are 2 (6 bits-per-pixel) and 12 (1 bit-per-pixel) and that the compression ratios follows a beta-distribution $\text{Beta}(2, 3)$.

Let X be random variable distributed according to a beta distribution of parameters $\alpha = 2$ and $\beta = 3$, $X \sim \text{Beta}(2, 3)$. The compression ratio C is also a random variable distributed according to:

$$C = \frac{12}{1 + 5 * X}$$

Since the support of a beta distribution is $[0, 1]$, this ensures that the compression ratio C lies within the domain $[2, 12]$. Figure 4.7 shows the probability density function of the compression.

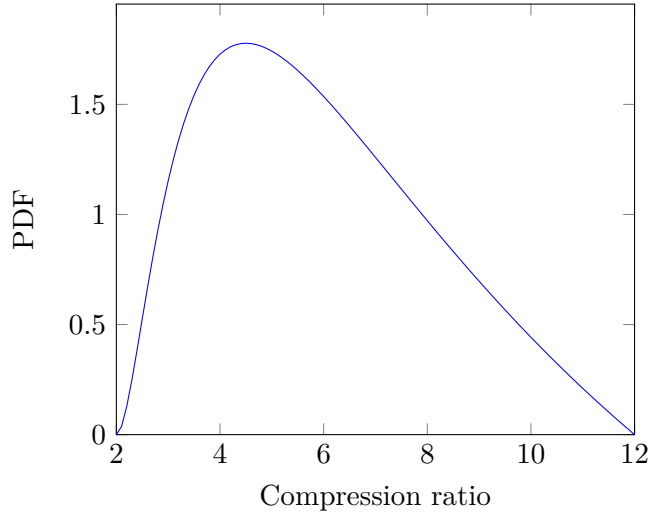


Figure 4.7 – Probability density function of the compression ratio with a 12 bits-per-pixel original encoding using a beta-distribution of parameters $\alpha = 2$ and $\beta = 3$.

With this method, the average compression ratio is 4 and corresponds to an average of 3 bits-per-pixel. The average compression ratio in considered compression systems today is 3 (2.4 bits-per-pixel), but we failed to reach it using a standard distribution while maintaining a consistent probability density function.

We consider three acquisition scenarios denoted *Normal*, *DayOnly* and *Full*.

Chapter 4. Flexible scheduling of acquisition downloads

Table 4.3 presents information regarding the considered acquisitions scenarios. Total volumes and profits are rounded to three digits. In each scenario, acquisitions are made between June 21, 2017 at 10:30 A.M. and June 22, 2017 at 10:30 A.M. The *Normal* scenario corresponds to multiple short acquisitions made at regular intervals. The *DayOnly* scenario corresponds to acquisitions made only during half of the satellite orbit (when the satellite is on the day-side of the Earth). The *Full* scenario corresponds to constant and large acquisitions. The daily acquisition volumes in these scenarios are orders of magnitude larger than the ones in existing Earth observation missions (which are around one terabits per day).

	# Acquisitions	Total volume	Total profit	Mean hourly volume
<i>Normal</i>	758	7.890 terabits	231480.356	394.49 gigabits
<i>DayOnly</i>	1241	33.288 terabits	379272.026	1331.54 gigabits
<i>Full</i>	3826	96.430 terabits	1143483.482	3857.2 gigabits

Table 4.3 – Acquisition scenarios for experiments.

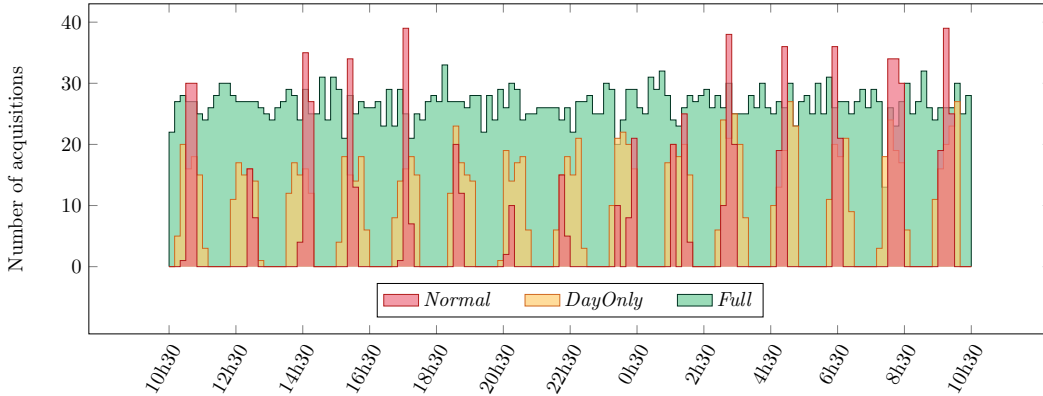


Figure 4.8 – Number of acquisitions every ten minutes for the considered acquisition plans.

Figure 4.8 shows the acquisition rates for the considered scenarios, i.e., the number of images acquired every tens of minutes. We can see that *Full* is the only scenario where images are acquired continuously and with a high rate (average of 26.6 acquisitions every ten minutes).

4.4.1.2 Computation of visibility windows

We consider seven networks containing 1, 2, 4, 7, 10, 13 and 16 stations providing respectively 5, 12, 23, 37, 64, 90 and 108 visibility windows. These are the best networks found when solving the **MaxPDT** problem over a 20 years horizon using the \mathcal{N}_{16} set of locations (see Chapter 3). For each station, we generate visibility

windows using the Systems Tool Kit software (see Chapter 1) from June 21, 2017 at 10:30 A.M. to June 22, 2017 at 2:00 P.M.

We then generate nominal data rate functions (assuming no clouds) for each visibility window according to the elevations between the stations and the satellite. We assume that the communication link can be established as soon as the satellite is visible (i.e., above 0° elevation), but that the communication is blocked until 5° elevation. After this point, the data rate increases up to 10 gigabits-per-second above 20° elevation and remains at this value until decreasing below 20° .

We used the same assumptions as in Chapter 3 for the impact of clouds. We used data from the ERA Interim database, fetching the cloud cover over the closest cell (from a station) and doing a linear interpolation over time (see Chapter 1).

Assuming a cloud cover cc_v during a visibility window v , the data rate function of this visibility window can be computed as follows:

$$f_v(t) = (1 - cc_v) * \begin{cases} 0 & \text{if } e_v(t) < 5 \\ 2.5 & \text{if } 5 \leq e_v(t) < 10 \\ 5 & \text{if } 10 \leq e_v(t) < 20 \\ 10 & \text{if } 20 \leq e_v \end{cases}$$

where $e_v(t)$ is the elevation between the station and the satellite during the visibility window v at time $t \in [t_v^{start}, t_v^{end}]$. Figure 4.9 shows the data rate as a function of elevation for various cloud covers.

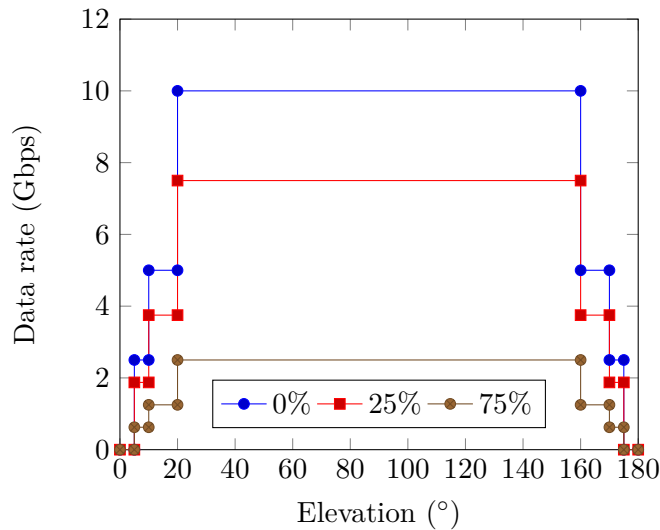


Figure 4.9 – Data rate versus elevation for different percentage of cloud cover.

Table 4.4 shows the total available capacities (without taking overlaps into account) for each considered network, together with the start of the first visibility window and end of the last visibility window. The maximum gap indicated corresponds to the maximum duration without access to a station (between two visibility

Chapter 4. Flexible scheduling of acquisition downloads

windows).

# Stations	First window	Last window	Maximum gap	Total capacity
1	June 21, 13:03:36	June 22, 04:01:52	10:25:26	10.877 terabits
2	June 21, 11:00:52	June 22, 11:15:12	04:44:03	13.398 terabits
4	June 21, 11:00:52	June 22, 12:55:41	03:10:52	26.774 terabits
7	June 21, 11:01:09	June 22, 12:52:29	01:48:56	73.303 terabits
10	June 21, 10:50:46	June 22, 12:52:29	01:31:04	92.524 terabits
13	June 21, 10:45:29	June 22, 12:52:29	01:29:30	104.739 terabits
16	June 21, 10:41:53	June 22, 12:55:41	01:29:30	126.081 terabits

Table 4.4 – Details of the considered visibility window scenarios.

Figure 4.10 shows a temporal view of the visibility windows for the 16 considered stations. The width of each visibility window corresponds to its duration, while its height correspond to its download capacity (not taking cloud into account). The color of each visibility window indicates the amount of clouds above the station during the visibility window. Table B.2 in Appendix B shows which stations are used for each of the considered networks. These networks have been optimized over a 20 years horizon, which is why their performances on the single day considered in these experiments may not be optimal.

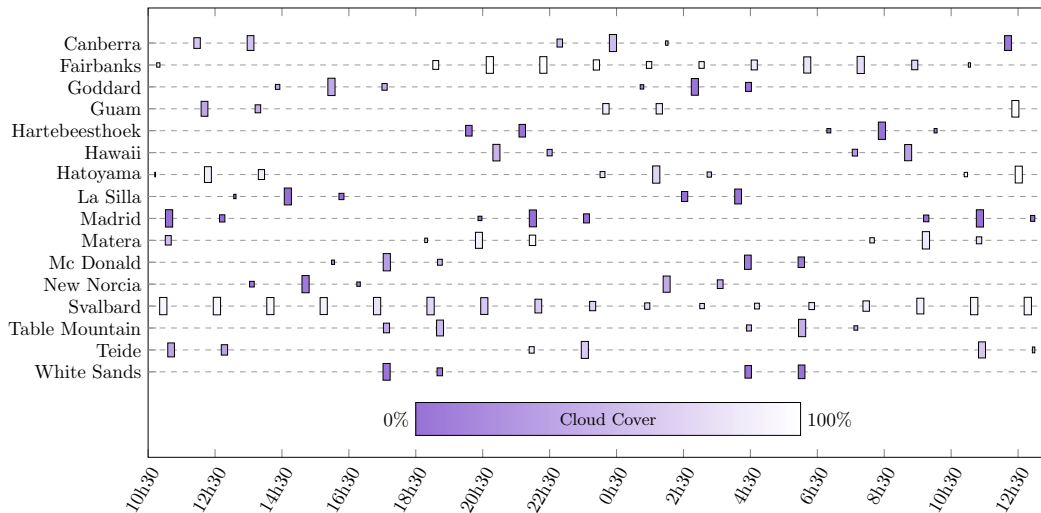


Figure 4.10 – Visibility windows for the considered stations in the various scenarios.

We assume that updated download windows and ground assignments are computed and uploaded to the satellite every 6 hours. The first upload is made on June 21, 12:00 A.M. (10 hours before the first acquisition), and the last one on June 22, 12:00 P.M., making a total of 4 ground optimization processes during the simulation.

By combining the previously described scenarios regarding acquisitions (3 scenarios) and visibility windows (7 scenarios), we obtain a total of $3 \cdot 7 = 21$ scenarios.

4.4.2 Context and considered sets of experiments

We test these scenarios using a combined ground optimization (i.e., ground optimization running both **MaxCap** and **GrdAsg**) with various sets of parameters for both the **GrdAsg** problem and the on-board heuristic. We also run experiments with a ground optimization using only **MaxCap**, with various parameters for the on-board heuristic. These two sets of experiments are respectively denoted Cf_1 and Cf_2 in the following.

4.4.2.1 Definition of policies and modifiers

Matching policies Given an acquisition a release at date r_a and due for d_a , the corresponding download must be made within the interval $[r_a, d_a]$. When assigning acquisitions to download windows in the **GrdAsg** problem, we do not know the exact download times of acquisitions, we only know that if acquisition a is assigned to download window w , the download of a will be made between t_w^{start} and t_w^{end} . Thus, we need to decide prior to the assignments which download windows can be used to download each acquisition. In order to compute the set $\mathcal{W}[a]$ of download windows to which we can assign a , we propose two matching policies.

In the **soft** policy, acquisition a can be assigned to download window w if the release date of a is before the end of w and the due date of a is after the start of w :

$$\mathcal{W}[a] = \left\{ w \in \mathcal{W} : r_a \leq t_w^{end} \text{ and } t_w^{start} \leq d_a \right\}$$

In the **hard** policy, acquisition a can be assigned to download window w only if the release date of a is before the start of w and the due date of a is after the end of w :

$$\mathcal{W}[a] = \left\{ w \in \mathcal{W} : r_a \leq t_w^{start} \text{ and } t_w^{end} \leq d_a \right\}$$

Example 11. Figure 4.11 shows multiple acquisitions a_i with their associated release and due dates, and a single download window w .

If we consider a **soft** matching policy, acquisitions a_3 and a_6 cannot be assigned to the download window w because the release date of a_3 is after the end of w and the due date of a_6 is before the start of w . Acquisitions a_1 , a_2 , a_4 and a_5 can be assigned to w even if acquisitions a_4 and a_5 are released after the start of w and a_2 and a_5 are due before the end of w .

If we consider a **hard** matching policy, only acquisition a_1 can be assigned to the download window w because it is the only acquisition that is released before the start of w and whose due date is after the end of w .

Volume modifiers Volume modifiers are used in the **GrdAsg** problem to compute estimated volume \bar{q}_a for acquisitions to take into account uncertainties regarding

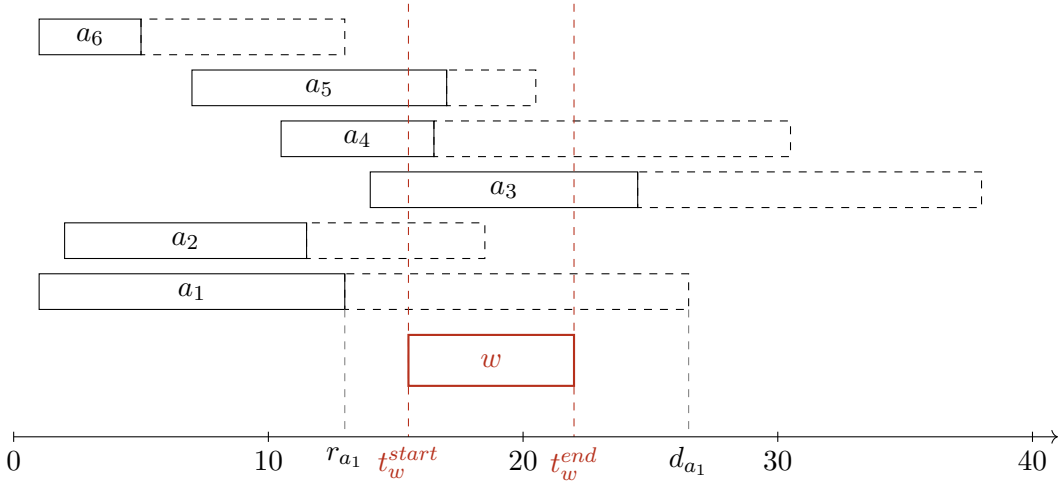


Figure 4.11 – Example of assignments of acquisitions depending on the chosen matching policy.

their sizes. We consider in these experiments two trivial volume modifiers, **VolExp** and **VolMax**, that simply use the expected q_a^{exp} and maximum q_a^{max} volumes of acquisitions.

We also consider a weighted volume modifier that computes modified volumes based on two parameters, k_1 and k_2 , and on the profit of acquisitions. We call this modifier **VolWgt** $[k_1, k_2]$, where k_1 and k_2 are parameters that represent the minimum and maximum values for this volume modifiers. Equation (4.12) gives the formal definition for **VolWgt** $[k_1, k_2]$.

$$\bar{q}_a = q_a^{exp} + \begin{cases} k * (q_a^{exp} - q_a^{min}) & \text{if } k \leq 0 \\ k * (q_a^{max} - q_a^{exp}) & \text{if } k > 0 \end{cases} \quad (4.12)$$

The value of k depends on the profit p_a of the acquisition and on the minimum and maximum profits p_{min} and p_{max} of all acquisitions in \mathcal{A} :

$$k = k_1 + (k_2 - k_1) * \frac{p_a - p_{min}}{p_{max} - p_{min}}$$

Figure 4.12 shows examples of various volume modifiers for $p_{min} = 50$ and $p_{max} = 200$.

Profit modifiers Profit modifiers are used in the **GrdAsg** problem and in the on-board algorithms to avoid downloading acquisitions at the last minute. The modified profit considered if acquisition a is downloaded during download window w is denoted \bar{p}_{aw} . We propose a non-biased profit modifier **ProfitId**, a parameterized modifier **ProfitDc** $[\mathbf{b}, \mathbf{d}]$ based on the download time of acquisitions and a modifier **ProfitCa** based on the remaining capacity to download acquisitions.

The **ProfitId** modifier generates non-biased profits, i.e., for each acquisition a

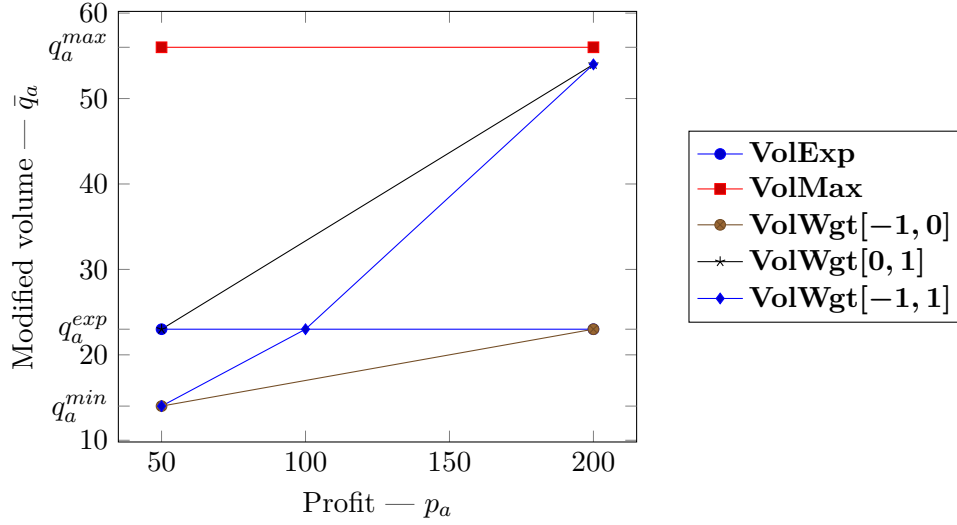


Figure 4.12 – Examples of simple volume modifiers and weighted volume modifiers with various values of k_1 and k_2 .

and download window w , $\bar{p}_{aw} = p_a$.

The **ProfitDc**[\mathbf{b}, \mathbf{d}] modifier decreases or increases linearly the profit depending on the download time of the acquisition. We consider that this time corresponds to the start of the considered download window t_w^{start} because we need to be able to modify profits before assigning exact download times to acquisitions.

We first define the freshness $\alpha(a, w)$ of acquisition a when downloaded during w as follows:

$$\begin{aligned} \alpha(a, w) : \mathcal{A} \times \mathcal{W} &\rightarrow \mathbb{R} \\ (a, w) &\rightarrow 1 - \frac{t_w^{start} - r_a}{d_a - r_a} \end{aligned}$$

In practice, $\alpha(a, w)$ is positive because it is not possible to download an acquisition using a download window that start after the deadline of the acquisition, i.e., $t_w^{start} < d_a$ if a is downloaded using w . But $\alpha(a, w)$ might be greater than one if the download window chosen to download a starts before the release date of a . In the following, we will assume that $\alpha(a, w) \in [0, 1]$.

Using this freshness function, we can define the profit modified by **ProfitDc**[\mathbf{b}, \mathbf{d}] using equation 4.13:

$$\bar{p}_{aw} = p_a * (d + (b - d) * \alpha(a, w)) \quad (4.13)$$

When using **ProfitDc**[\mathbf{b}, \mathbf{d}], the value of the modified profit \bar{p}_{aw} goes from $p_a * b$ when the acquisition is fresh (i.e., downloaded soon after its release) to $p_a * d$ when the acquisition is downloaded just before its deadline.

Figure 4.13 shows examples of profit modifiers **ProfitDc**[\mathbf{b}, \mathbf{d}] for various values of b and d . We consider in the examples a single acquisition a released at $r_a = 30$,

with a profit $p_a = 180$ and a deadline $d_a = 90$.

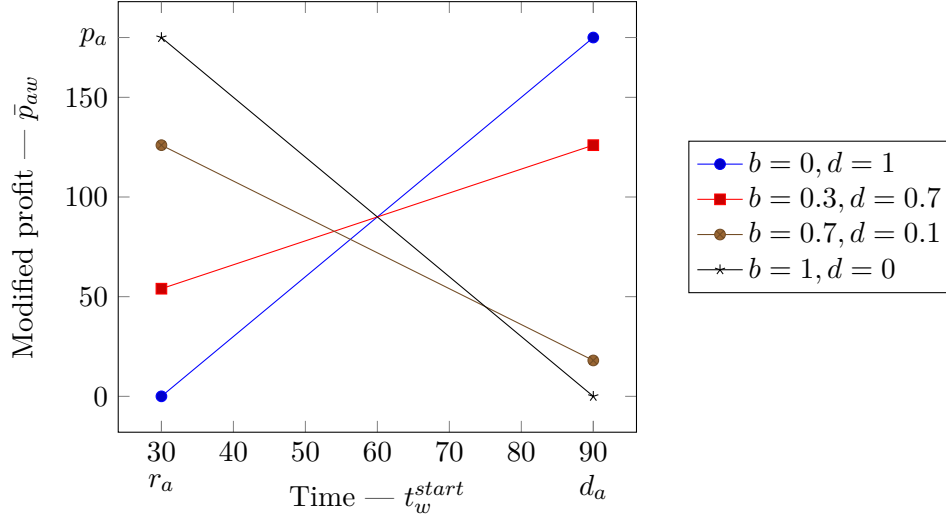


Figure 4.13 – Examples of volume modifiers **ProfitDc**[**b**, **d**] for various values of b and d for a single acquisition a .

The profit modifier **ProfitCa** reduces the profit p_a depending on the remaining capacity to download a after w (including the capacity of w). The idea of this biased profit is to prioritize acquisitions for which this remaining capacity is small. Equation (4.14) gives the formal definition for **ProfitCa**.

$$\bar{p}_{aw} = \frac{p_a}{\log_{10}(R(w, a))} \quad (4.14)$$

where $R(w, a)$ is the available capacity to download a during or after w :

$$R(w_i, a) = \sum_{\substack{w_j \in \mathcal{W}[a] \\ i \leq j}} Q_{\nu_{w_j}}(t_{w_j}^{start}, t_{w_j}^{end})$$

A logarithm is used in Equation (4.14) because what really matters is the order of magnitudes of the remaining capacity, and not really the actual capacity.

4.4.2.2 Considered sets of experiments

In the Cf_1 set of experiments, we need parameters for both the **GrdAsg** problem and the on-board sorting procedure. For the **GrdAsg** ground optimization problem, we consider:

- both the **soft** and the **hard** matching policies;
- five volume modifiers: **VolExp**, **VolMax**, **VolWgt**[$-1, 1$], **VolWgt**[$0, 1$] and **VolWgt**[$-1, 0$];
- five profit modifiers: **ProfitId**, **ProfitDc**[$1, 0$], **ProfitDc**[$0.5, 0.5$], **ProfitDc**[$0, 1$] and **ProfitCa**.

For the on-board sorting procedure we consider:

- two priorities: **PrtB** and **PrtC**;
- three profit modifiers for assigned acquisitions: **ProfitId**, **ProfitDc[1, 0]** and **ProfitCa**;
- five profit modifiers for other acquisitions: **ProfitId**, **ProfitDc[1, 0]**, **ProfitDc[0.5, 0.5]**, **ProfitDc[0, 1]** and **ProfitCa**;
- both ordering based on modified profits ($R = false$) and based on ratios between modified profits and volumes ($R = true$).

By making combinations of these parameters, we create a total of 3000 configurations for our experiments.

In the Cf_2 set of experiments, we only need parameters for the on-board sorting procedure. We consider:

- no priority since acquisitions are not assigned on the ground: **PrtA**;
- 13 profit modifiers for acquisitions: **ProfitId**, **ProfitCa** and **ProfitDc[b, 1 - b]** for $b \in \{0, 0.1, \dots, 1\}$.
- both ordering based on modified profits ($R = false$) and based on ratios between modified profits and volumes ($R = true$).

In this set of experiments, we do not distinguish between profit modifiers for assigned or non-assigned acquisitions since acquisitions are not pre-assigned on the ground (all acquisitions are in the \mathcal{A}_w^U set). By making combinations of these parameters, we create a total of 26 configurations for our experiments.

We compare ourselves to an upper bound and two reference configurations, which are approximations of what is feasible in reality. These approximations are based on a complete ground optimization (computation of download windows and assignments of acquisitions to download windows) considering a **soft** matching policy, and no on-board computation. Since the ground optimization does not create an actual schedule but only a packing, we simulated the actual schedule on-board by considering another packing problem: given a download window w , we download as much as possible from the assigned acquisitions (considering them by decreasing profit), and non downloaded acquisitions are thrown away.

In order to obtain an upper bound for our instances, we consider a configuration where everything is known on the ground (actual data rate functions and volumes of acquisitions).

We also compare ourselves to two different configurations: one where the ground optimization consider actual data rate functions (assuming perfect forecasts) but maximum volumes for acquisitions (since the schedule cannot be modified on-board), and another one that consider predicted data rate functions and maximum volumes for acquisitions. The last configuration is not realistic since, as soon as we keep uncertainty in the download plans (i.e., data rate functions), we need an on-board procedure to fix the schedule before each download window.

We refer to these configurations as Ub (upper bound), Ref^+ (actual data rate functions and maximum volumes) and Ref (predicted data rate functions and maximum volumes).

By using only a packing procedure on-board, we obtain an upper bound on what would actually be feasible in reality. Indeed, we do not consider switch between data rates in the middle of downloads, or problems with acquisitions that become available during a download window or whose deadline is in the middle of a download window. The packing assumes that it is possible to download acquisitions right from the start of the download window, but this may not be possible in reality. If we consider a download window w with a constant data rate and a single acquisition a released just before the end of w , there may not be sufficient time between the release of a and the end of w to download a . However packing a inside w is valid as long as w is large enough to download a .

4.4.3 Results and analysis

In order to evaluate the various configurations, we implement the models for the `MaxCap` and `GrdAsg` problems and the on-board algorithm in C++. We use CPLEX 12.7.1 to solve the mathematical models for `MaxCap` and `GrdAsg`. We let the solver deal with the piecewise linear equations (4.7d–4.7e) using `IloPiecewiseLinear` objects.

4.4.3.1 Global performances

Figure 4.14 shows the total earned profit (on the left) and the number of downloaded acquisitions (on the right) for the two sets of experiments and the reference configurations. For each set of experiments and scenarios (acquisition plans and numbers of stations), the value shown is the best over the given set. In each figure, the “Max” entries respectively correspond to the total profit and the number of acquisitions in the considered acquisition plans (as detailed in Table 4.3).

The first visible result on these graphics is that it seems impossible to download all acquisitions using the considered networks of optical stations. Even considering perfect information without uncertainties (Ub), some acquisitions are not downloaded. This may be due to acquisitions having a too short deadline (i.e., lower than the maximum gap between visibility windows shown in Table 4.4), possibly combined with a very low profit.

With the considered network of seven stations, we manage to download 91.16% of the acquisitions for the *Normal* acquisition plan and 88.56% for the *DayOnly* plan. The Ub configuration gives us slightly higher numbers: 92.45% and 90%.

For the *Normal* acquisition plan, both Cf_1 and Cf_2 sets have similar results, and reach their maximum profits with seven stations. The reference configurations Ref and Ref^+ seem to provide results similar to the upper bound, which are better than the ones for Cf_1 and Cf_2 . This is likely due to the simplifications made within these configurations: since the scheduling aspect of the problem is not taken into

4.4. Experiments

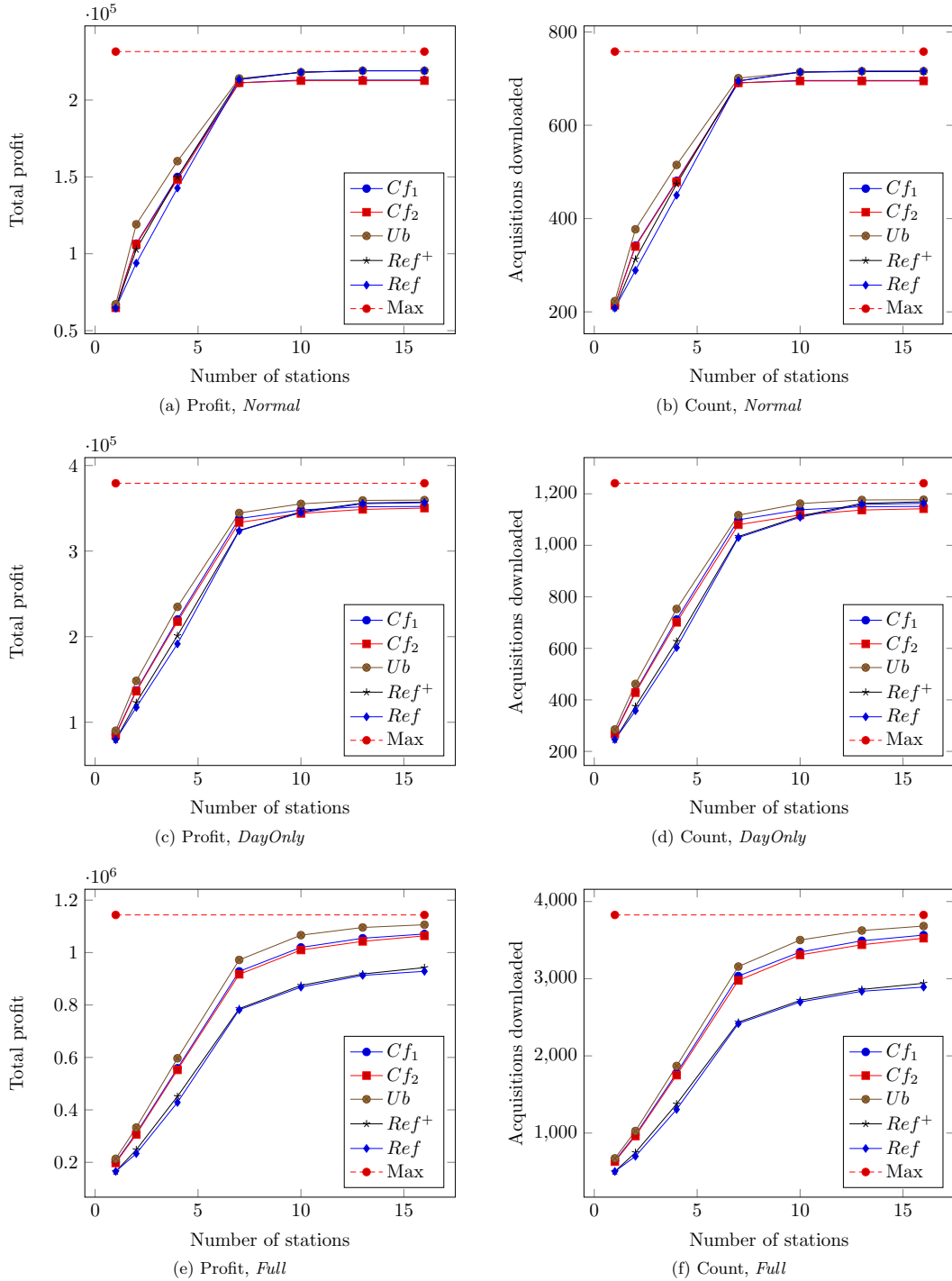


Figure 4.14 – Earned profit and number of acquisition downloaded versus number of stations for various configurations.

account (we consider a simple packing), we obtain non-feasible download schedules that are upper bounds of what is actually achievable.

Results are similar for the *DayOnly* acquisition plan, except that the profit and number of download acquisitions do not stabilize as much when using more than seven stations.

For the *Full* acquisition plan, both Cf_1 and Cf_2 also have similar results, but the difference between these two sets of experiments and the reference configurations *Ref* and *Ref+* is significant. These experiments show that, with such amount of acquisitions, having an on-board procedure is necessary to create efficient download schedules. Using Cf_1 or Cf_2 , we manage to download only 79.3% of the acquisitions using 7 stations, but we can reach 93.18% using 16 stations. It is clear that with higher acquisition volumes, the number of available visibility windows (which depends on the number of available stations) will have more impact than with smaller acquisition volumes.

Comparing the percentages of download acquisitions with the mean hourly acquisition volumes of the three scenarios, we observe that the distribution of acquisitions over time has an impact on the final results.

4.4.3.2 Impact of the matching policy

Figure 4.15 shows, for both matching policies, the number of times a set of experiments using the given policy find the best solution (between all the solutions found). The figure also shows the ratio between the best solution found with the **hard** matching policy and the **soft** matching policy.

For both the *Normal* and *DayOnly* acquisition plans, the ratio between both matching policies gets very close to one when the number of stations is higher than seven. In both scenario, the **soft** matching policy has better results than the **hard** matching one, except for very small numbers of stations for the *Normal* acquisition plans. This seems to indicate that when the plan is not that much filled, giving more freedom to the on-board heuristic, the chance of finding better solutions increases.

Results for the *Full* acquisition plan show a very different behavior. For small number of stations, the **soft** matching policy has better result, but for high number of stations, the **hard** matching policy finds better solutions more often. Furthermore, the ratio between both matching policies is higher than for the two other plans, meaning that the matching policies have much more impact when the plans contain lots of acquisitions.

4.4.3.3 Impact of the on-board parameters

Figure 4.16 compares, for each priority policy and scenario, the impact of taking into account the volumes of acquisitions when ordering them, i.e., the difference between sorting only regarding computed profits of acquisitions, or using the ratios between profits and volumes of acquisitions (see Section 4.3.3).

For the *Normal* acquisition plan, there is not a clear difference between the two approaches. This may be explained by the small amount of data that needs to be

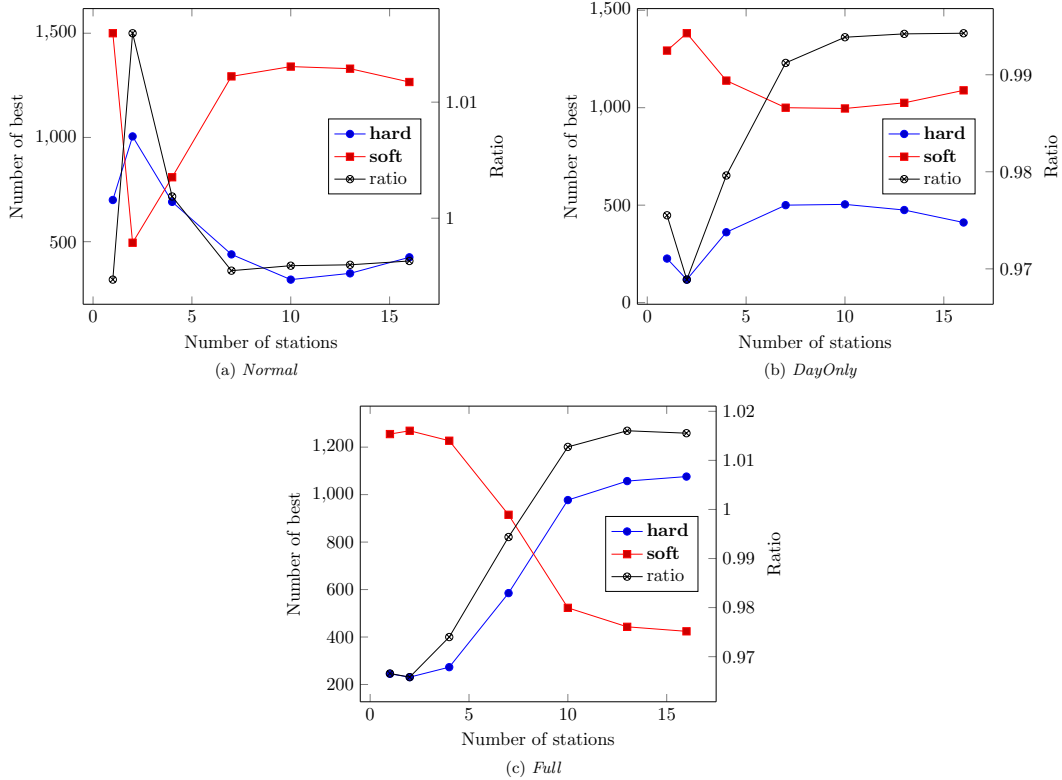


Figure 4.15 – Number of configurations for which each matching policy has found the best profit and comparison between the profit found.

downloaded compared to the available download capacity. This is confirmed by the results for the *DayOnly* acquisition plan with larger ground networks: the difference between the two approaches reduces as the number of stations gets larger.

However, these results must be compared with the actual differences in profits between the two approaches. When not taking acquisition volumes into account, the average benefit is less than 0.03% and the maximum benefit is smaller than 0.7% for **PrtA** and **PrtB**, and only 3% for **PrtC**. In comparison, the average benefit when taking acquisition volumes into account is about 3% for **PrtC** and **PrtB**, and as much as 10% for **PrtC**.

Results for the *Full* scenario are less ambiguous. It is clear that taking volumes of acquisitions into account when ordering them raises better results.

Figure 4.17 compares the total earned profit and the mean age of acquisitions (the time between the release date of an acquisition and its actual download) for various number of stations and on-board profit modifiers, for configurations inside Cf_2 .

Each curve (color) consider a given number of stations, and each point corresponds to different values (b, d) for the **ProfitDc** $[b, d]$ weight modifier: from $(0, 1)$ for the bottom-left point to $(1, 0)$ for the top-right point. Figure 4.17 also shows

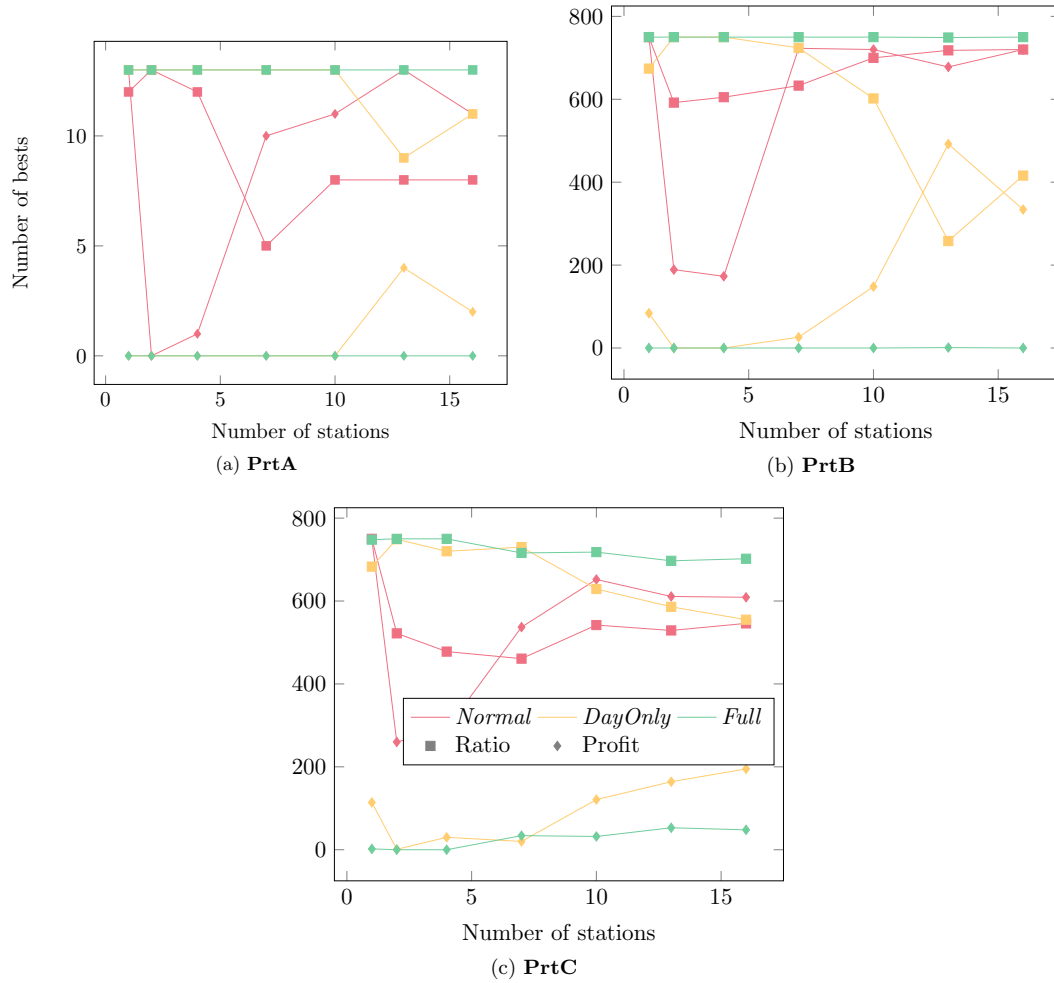


Figure 4.16 – Number of configurations for which sorting using the ratio between weight and volume or only volume find the “best” solution.

the total earned profit and the mean age of acquisitions for the configuration using **ProfitCa**.

In most cases, there exists a configuration using **ProfitDc**[b, d] that dominates the corresponding configuration using **ProfitCa**, making the **ProfitCa** profit modifier mostly uninteresting.

For the *Normal* acquisition plan, the values of b and d seem to have few impacts on the mean age of acquisitions, and the pair (1, 0) seems to be the most appropriate one.

For the *Full* acquisition plan, the impact of b and d is much more visible. Increasing d in favor of b seems to reduce the mean age of acquisitions and the profit, while reducing d in favor of b seems to increase both objectives. By customizing b and d , we can tune our on-board heuristics to boost one objective or the other.

Results for the *DayOnly* acquisition plan are in-between results from the two

other scenarios. The impact of b and d on the mean age of acquisition is less visible here than for the *Full* plan.

These results show that for realistic scenario (e.g., *DayOnly*), the mean age of acquisitions is lower than 30 minutes.

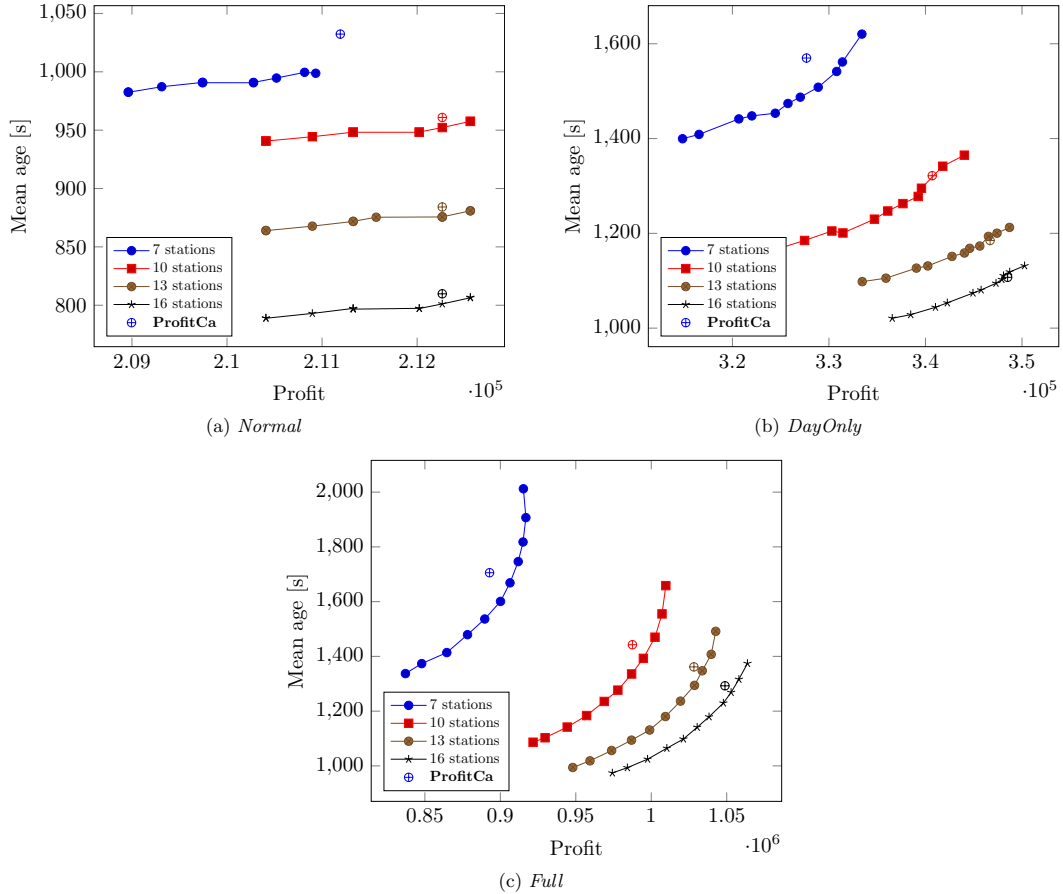


Figure 4.17 – Total profit vs. mean acquisition age for various number of stations.

Summary

In this chapter, we presented the problem of planning and scheduling the downloads of images acquired by a low-Earth orbiting satellites in the context of free-space optical communications. We explained that modern concepts and technologies such as fixed-quality (variable bit-rate) on-board compression and free-space optical communications bring multiple uncertainties into the problem. To face these uncertainties, we proposed a multi-step procedure: computation of download windows and possible flexible assignments of acquisitions to download windows on the ground, followed by a configurable greedy on-board scheduling heuristic. Experiments showed that the use of on-board processing has a positive impact on the number of ac-

Chapter 4. Flexible scheduling of acquisition downloads

quisitions downloaded, and that some parameters impact the performances of the on-board scheduling procedure. Pre-assigning acquisitions on the ground seems to have few impacts on the final outcome of the system, but might be interesting from an operational point-of-view to prioritize the download of acquisitions during specific download windows. Since the on-board procedure can completely modify download plans, using a **soft** matching policy seems to be the best approach here. Finally, profit modifiers allow us to prioritize age of acquisitions over profits, especially for plans with lots of acquisitions.

Conclusion

In this thesis, we defined and presented two problems related to the development of free-space optical communications in the context of Earth observation missions.

In Chapter 3, we presented the optical ground station network optimization problem. We showed that under some assumptions, this problem can be reduced to two interleaved selection problems: the selection of ground stations and the selection of download points. We proposed two hierarchical approaches to solve it. The first one uses an exhaustive enumeration to evaluate all possible networks while the second one is based on a branch and bound algorithm to reduce the search space. We proposed a dynamic programming algorithm to solve the download points selection problem and use it in both enumeration approaches. To improve this algorithm, we designed a dominance rule tailored to real instances. Experiments showed that, even if the dynamic programming algorithm had a theoretical exponential solving time, its performance were near-linear on real instances. The exhaustive enumerations of subsets of locations prove to be efficient to solve instances with small number of possible locations for ground stations, but due to the exponential number of possible networks, this approach was not suited for large instances. However, the branch and bound approach managed to find optimal or very efficient networks, even on very large instances.

Additional research efforts are needed to investigate the complexity of the download points selection problem, and in consequence, the overall problem, for real instances where conflicts within visibility windows can be represented as interval graph with possible overlaps between slots. In this thesis, we chose to reduce each visibility window to a single download point, thus transforming overlaps into conflicts between download points. A finer approach could be considered where visibility windows could be split into multiple download points in order to allow the algorithm to partially use two overlapping visibility windows. Results showed that when looking for large networks of optical ground stations, the percentage of data transferred quickly reach 100%, and thus cannot be improved by adding new stations. It would be interesting to consider other objectives or multi-objectives approaches to differentiate between two networks with a 100% percentage of data transferred, such as the minimization of the age of acquisitions, i.e., the time between acquisitions and downloads. Variations or extensions of the problem could also be considered. In order to avoid temporal variations over the year, optimization over rolling horizons could be interesting, or maximization of the minimum monthly percentage of data transferred. Finally, for future space missions, optimization for multiple satellites at once could be investigated. In such case, a bi-objective approach would be preferred, in order to maximize both the percentage of data transferred from each satellite but also the fairness between them.

Conclusion

In Chapter 4, we presented the problem of scheduling downloads of acquisitions using free-space optical communications in the context of Earth observation missions. This problem mixes two type of uncertainties: uncertainties regarding the volumes of acquisitions after compression, and uncertainties regarding the communication channel due to the use of free-space optical communications and the presence of clouds. We proposed a flexible approach, where download windows are computed on the ground using cloud forecasts and expected volumes for images, and final schedules are made on-board using a configurable greedy heuristics. Experiments showed that pre-assigning acquisitions to download windows on the ground has few impact on the final outcome of the system. These experiments also showed that with so many uncertainties, using an on-board procedure was necessary to obtain efficient schedules, especially with heavy acquisition plans. Finally, experiments showed that by configuring the on-board greedy algorithm, it was possible to prioritize the maximization of the total profit or the minimization of the age of acquisitions. However, these experiments also showed that some of the parameters for the on-board heuristic had few impacts on the final schedules.

Additional research efforts would be required to analyze in more depth the impact of the various parameters of the on-board heuristics and the complexities of the considered problems. In this thesis, we compared our approaches to various methods that solve the studied problem only on the ground. These methods generate upper bounds based on transformations of the problem into a packing of acquisitions into download windows. In order to obtain finer bounds, it would be interesting to consider more precise scheduling algorithms on the ground. In this work, we chose to completely re-compute the schedule prior to each download window, meaning that past decisions made on-board are not remembered for future schedules. Since the real volumes of acquisitions are revealed intermittently, the schedule could be repaired incrementally after the release of each acquisition, and thus the consumption of the algorithm could be spread over time. We also assumed that the acquisition plan was built on the ground and never modified. It would be interesting to consider procedures that create both the acquisition plan and the download plan at the same time, and update them on-board as uncertainties are resolved. This would likely increase the efficiency of the overall system and avoid wasting resources to acquire images that cannot be downloaded. Our work assumed a single optical communication channel, but future technologies could make use of wavelength division multiplexing (WDM). In this context, multiple channels would have to be considered, but their number would depend on the elevation angle between the satellite and the ground stations, and thus would change during visibility windows. Finally, more realistic simulations could be made in order to validate the proposed approaches, but this would require a much more complex simulation platform.

The advantages of free-space optical communications for image telemetry in Earth observation missions were already well-established. This thesis explored two problems arising from their utilization and showed that both of these could be

tackled efficiently by considering new approaches with the proposed optimization methods. In particular, experiments in this thesis confirmed the needs for on-board procedures for future satellite missions in order to increase the performance of Earth observation systems.

Considered networks

A.1 List of ground locations

#	Name	Latitude (°)	Longitude (°)	\mathcal{N}_{11}	\mathcal{N}_{16}	\mathcal{N}_{48}
1	Adelaide	-34.86	138.57			X
2	Arecibo	18.34	-66.75			X
3	Athens	38.00	23.70	X		
4	Aussaguel	43.43	1.50	X		
5	Bangalore	13.03	77.51			X
6	Bordeaux	44.98	-0.57			X
7	Canberra	-35.40	149.98		X	X
8	Dubai	25.27	55.27			X
9	Esrangle	67.88	21.06			X
10	Fairbanks	64.84	-147.73		X	X
11	Gelsdorf	50.60	7.01	X		
12	Goddard	39.00	-76.85		X	X
13	Guam	13.59	144.84		X	X
14	Hammaguira	30.88	-3.06			X
15	Hartebeesthoek	-25.89	27.71	X	X	X
16	Hatoyama	36.00	139.35		X	X
17	Hawaii	21.52	-158.00		X	X
18	Inuvik	69.32	-133.54	X		X
19	Kerguelen	-49.35	70.26	X		
20	Kiruna	67.86	20.96	X		X
21	Kourou	5.25	-52.80	X		
22	La Silla	-29.26	-70.74		X	X
23	Lustbuhel	47.07	15.49			X
24	Madrid	40.43	-4.25		X	X
25	Magdalena	33.98	-107.19			X

Appendix A. Considered networks

26	Mas Palomas	27.76	-15.63	X		X
27	Matera	40.65	16.70		X	X
28	Mauritius	-20.17	57.52			X
29	Mc Donald	30.67	-104.02		X	X
30	Mountain View	37.39	-122.08			X
31	Mt Lemmon	32.44	-110.79			X
32	New Norcia	-30.97	116.21		X	X
33	Okinawa	26.50	127.90			X
34	Paris	48.84	2.28	X		
35	Perth	-31.80	115.89			X
36	Pic du Midi	43.00	0.10			X
37	Pratica Di Mare	41.70	12.50	X		
38	Pretoria	-25.76	28.24			X
39	Punta Arenas	-53.01	-70.85			X
40	Quantico	38.49	-77.32			X
41	Quilicura	-33.36	-70.73			X
42	Santiago	-33.49	-70.70			X
43	Sicily	37.75	13.91			X
44	Sierra Nevada	37.07	-119.40			X
45	Singapore	1.35	103.99			X
46	Sintra	38.87	-9.28			X
47	South Point	19.01	-155.66			X
48	Svalbard	78.23	15.38		X	X
49	Table Mountain	34.38	-117.68		X	X
50	Tampa	27.97	-82.48			X
51	Teide	28.30	-16.51		X	X
52	Toulouse	43.56	1.48			X
53	Troll	-72.01	2.54			X
54	White Sands	32.51	-106.61		X	X
55	Yatharagga	-29.04	115.35			X

Table A.1 – List of the possible locations for ground stations.

A.2 Map of ground locations

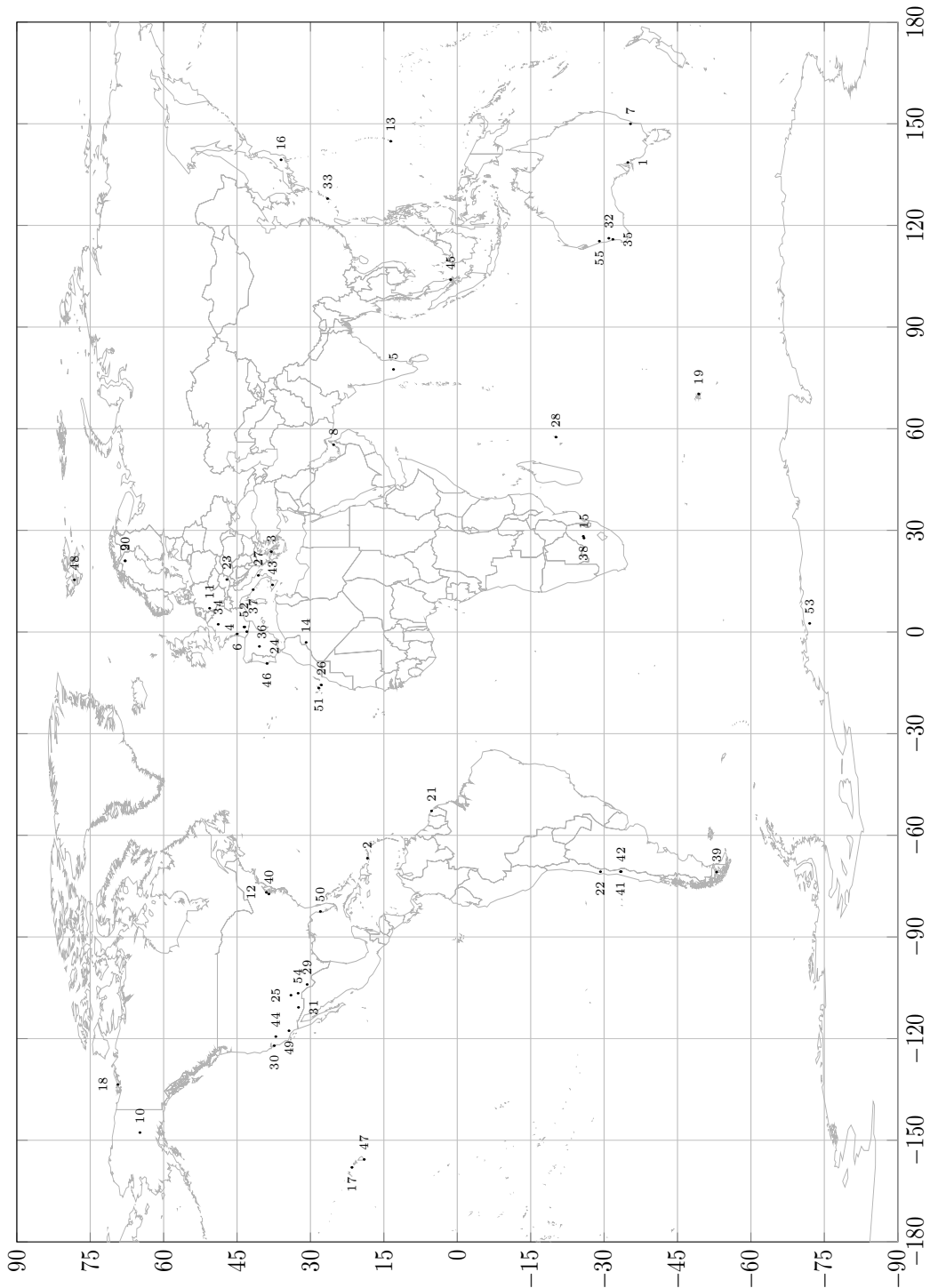


Figure A.1 – Map of the possible locations for ground stations.

Chosen locations in MaxPDT for the 20 year horizon

B.1 Chosen locations from \mathcal{N}_{11}

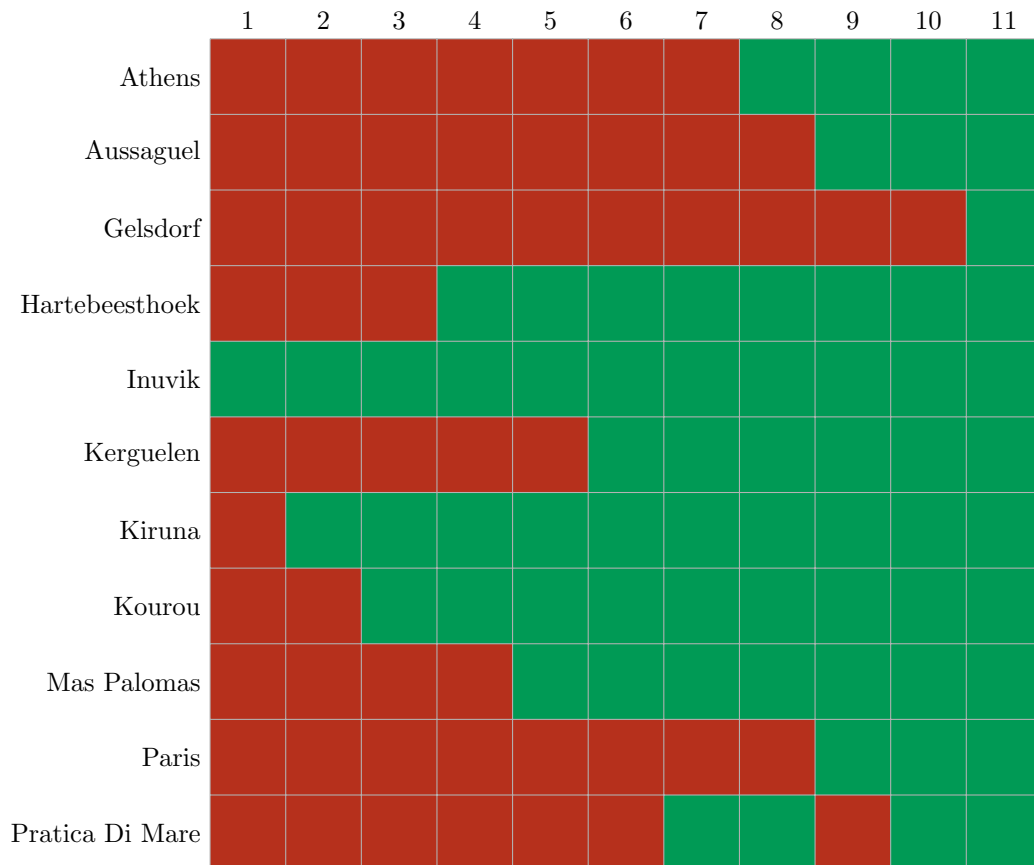


Figure B.1 – Chosen locations for $K = 1, \dots, 11$ using \mathcal{N}_{11} .

B.2 Chosen locations from \mathcal{N}_{16}

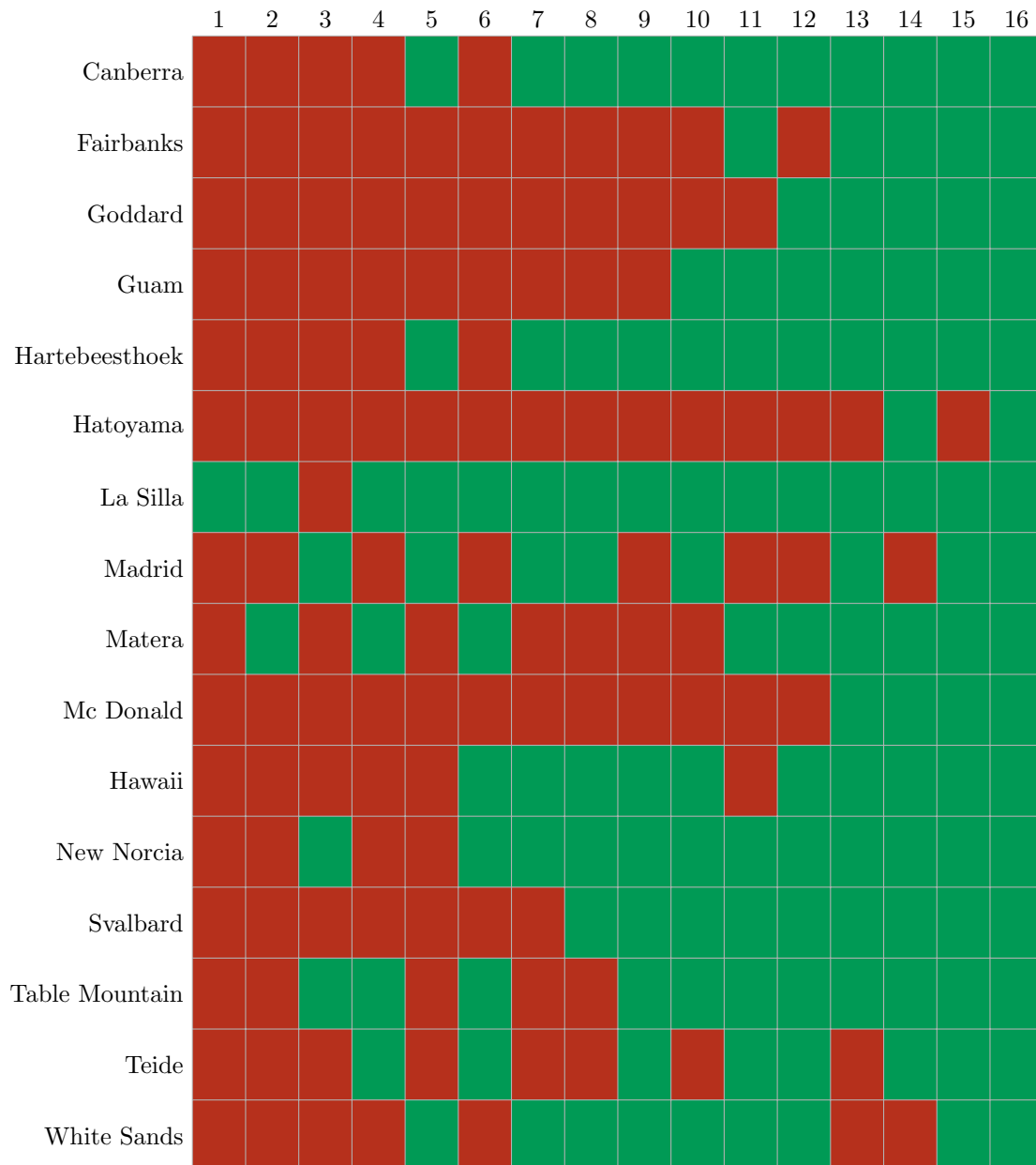


Figure B.2 – Chosen locations for $K = 1, \dots, 16$ using \mathcal{N}_{16} .

B.3 Chosen locations from \mathcal{N}_{48}

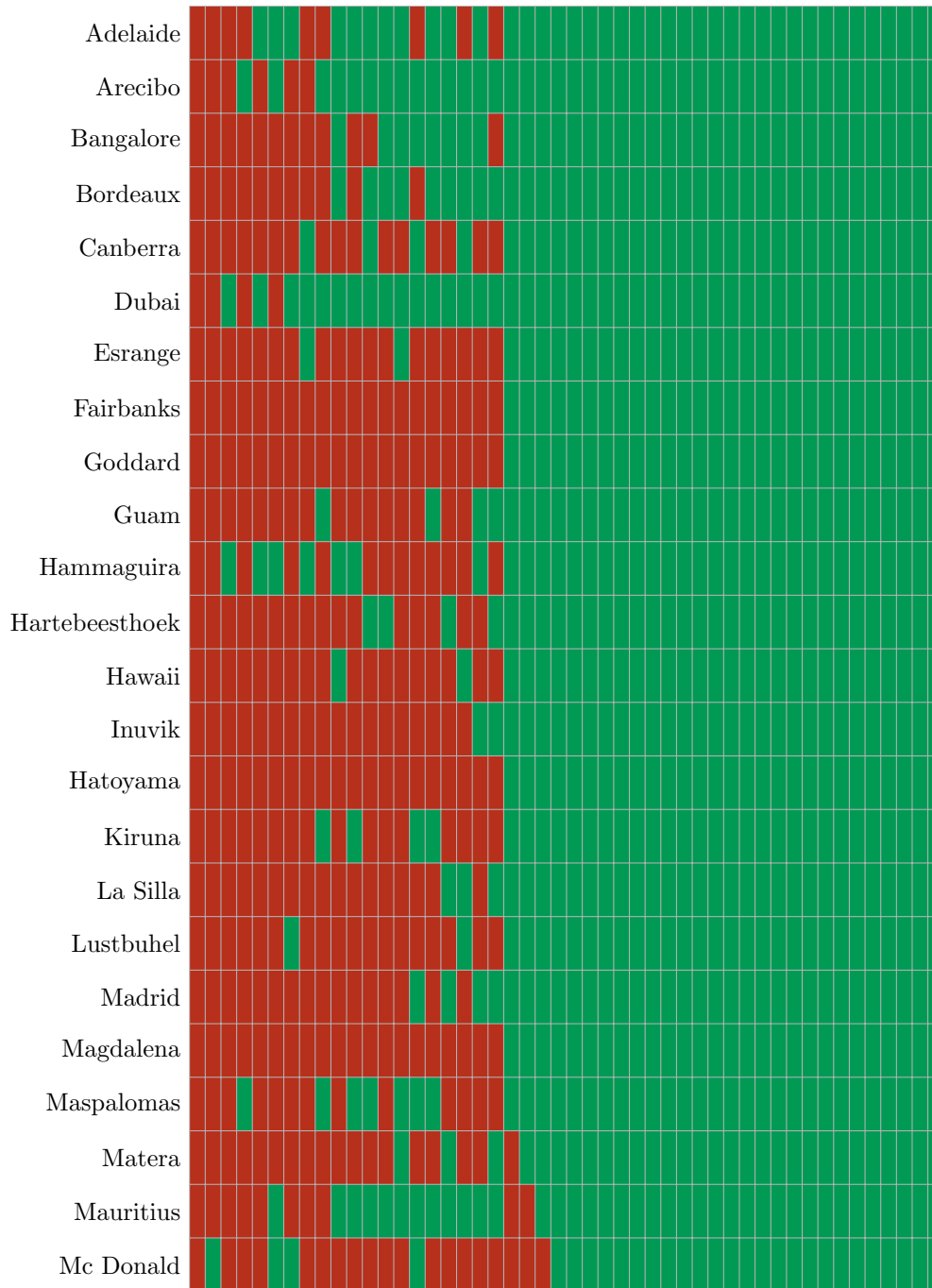


Figure B.3 – Chosen locations for $K = 1, \dots, 48$ using \mathcal{N}_{48} (Part 1).

Appendix B. Chosen locations in MaxPDT for the 20 year horizon



Figure B.4 – Chosen locations for $K = 1, \dots, 48$ using \mathcal{N}_{48} (Part 2).

Résumé en français

Optimisation pour les communications optiques dans les systèmes d'observation de la Terre

Depuis de nombreuses années, les satellites sont utilisés pour observer et surveiller la Terre depuis l'espace. Historiquement dédiées à des applications militaires, les missions d'observation de la Terre sont de plus en plus exploitées dans des applications commerciales. Afin de satisfaire les besoins des utilisateurs, de nombreux satellites vont être mis en orbite dans les années à venir et vont acquérir quotidiennement des centaines d'images très haute résolution qu'il va falloir redescendre sur Terre. Bien que ces images soient fortement compressées à bord, les volumes de données à transférer vers la Terre seront tels que de nouvelles technologies de communication doivent être envisagées. Parmi-elles, les communications optiques en espace libre (free-space optical communications, FSO) sont étudiées pour remplacer les radiocommunications actuelles qui deviennent de plus en plus limitées.

Les communications optiques en espace libre transmettent des données en utilisant des rayons lumineux, à l'instar de la fibre optique utilisée dans les réseaux informatiques, et peuvent ainsi fournir des débits de transfert grandement supérieurs aux débits atteignables par les radiocommunications. Des communications optiques ont déjà été utilisées dans des applications commerciales afin de relayer les données des satellites Sentinel en basse orbite, appartenant au programme Copernicus, via le système EDRS (European Data-Relay System) de relais géostationnaires.

Bien que les apports des communications optiques directement depuis l'espace vers la Terre aient déjà été démontrés dans le passé, de nombreux problèmes restent à résoudre avant de pouvoir considérer leur utilisation dans des applications industrielles. En effet, afin de pouvoir transférer des données depuis un satellite vers la Terre, les communications optiques doivent traverser l'atmosphère terrestre. L'impact des turbulences atmosphériques sur les communications optiques a déjà été profondément étudié, et des techniques d'atténuation de ces turbulences sont en cours de développement. En revanche, certaines perturbations de l'atmosphère ne peuvent être atténuées directement. En particulier, l'énergie qu'il faudrait déployer pour permettre aux communications optiques de traverser les nuages est largement supérieure à l'énergie disponible à bord des satellites d'observation. Afin d'envisager l'utilisation de communications optiques dans des applications industrielles, de nouveaux systèmes et de nouvelles procédures doivent être définis.

Appendix C. Résumé en français

Le but de cette thèse est d'étudier l'impact, sur le dimensionnement et l'exploitation d'un système spatial, de l'utilisation de communications optiques en espace libre. Plus précisément, nous nous intéressons aux problématiques liées à l'utilisation de cette technologie pour de la télémétrie image dans les systèmes d'observation de la Terre, c'est-à-dire la transmission (le vidage) d'images haute-résolution depuis un satellite d'observation vers des stations sol de réception. Dans ce contexte de communications optiques et de prise en compte des perturbations atmosphériques, nous étudions deux problèmes d'optimisation : d'une part, un problème de dimensionnement de réseaux de stations de réception ; d'autre part un problème de planification des vidages d'images depuis un satellite en orbite basse.

Nous nous intéressons tout d'abord à la création de réseaux de stations sol optiques efficaces. En effet, les nuages bloquant entièrement les communications optiques, l'utilisation d'une unique station de réception ne permettrait pas d'avoir des vidages réguliers depuis les satellites. Les satellites d'observation ayant à leurs bords des ressources mémoires limitées, cela impliquerait des pertes régulières d'images ou une sous-utilisation du système. Afin de palier à cela, l'utilisation d'un ensemble de stations optiques de réception, efficacement réparties autour du globe, est actuellement considérée.

L'objectif considéré consiste à minimiser la quantité de données perdues (i.e., ne pouvant être descendues vers une station sol) ce qui revient à maximiser le pourcentage de données transférées. A partir d'une analyse métier des contraintes d'un système composé d'un satellite défilant en basse orbite, nous proposons une formalisation sous forme de programme mathématique (programme linéaire en variables mixtes) de ce problème et montrons sa difficulté en termes de complexité. Afin de prendre en compte les variations saisonnières et annuelles de la couverture nuageuse, l'optimisation de réseaux de stations de réception doit être faite sur des horizons temporels très étendus. Pour faire face à la dimension du problème considéré et sous certaines hypothèses métiers, nous proposons une modélisation du problème en deux problèmes de sélection interdépendants : un premier problème de sélection des stations de vidage parmi un ensemble donné de stations possibles, et un problème de sélection des contacts (fenêtres de visibilité) entre le satellite et les stations. Ces deux problèmes sont résolus via une approche hiérarchique définie ci-après. Concernant le problème de sélection de stations, nous considérons deux méthodes : la première méthode repose sur une énumération exhaustive de tous les réseaux possibles de stations, alors que la deuxième est une approche de type séparation et évaluation (branch-and-bound). Pour résoudre le problème de sélection des contacts de vidage à l'intérieur de ces deux méthodes, nous proposons une méthode basée sur un algorithme de programmation dynamique. Afin d'éviter l'explosion combinatoire dans le choix des contacts de vidage, nous définissons une règle de dominance pour l'algorithme de programmation dynamique qui s'avère très efficace en pratique sur les instances réalistes.

Afin d'évaluer les performances de ces méthodes, nous proposons un ensemble d'instances de test en considérant des données réalistes à la fois sur la localisation des

stations, sur les taux de couvertures nuageuses (sur une période de 20 ans), sur les plans d'observation de la Terre, et sur les conditions de visibilité entre le satellite et les stations de réception. Ces instances sont étendues en augmentant les volumes de données à transférer pour évaluer le passage à l'échelle des méthodes proposées. Les résultats des expérimentations montrent que, pour résoudre le problème de sélection des contacts de vidage sur des instances réelles, l'algorithme de programmation dynamique est très efficace. Cependant sa complexité théorique est exponentielle, et elle est atteinte sur certains scénarios générés aléatoirement. Pour la résolution du problème complet, l'approche hiérarchique basée sur l'énumération exhaustive des réseaux de stations ne réussit pas à passer à l'échelle sur les instances de grandes tailles alors que celle basée sur une procédure de type séparation et évaluation réussit à résoudre de manière optimale certaines instances contenant jusqu'à cinquante stations possibles sur un horizon de plusieurs années.

Dans un second temps, nous nous intéressons au problème de la planification des vidages d'acquisitions depuis un satellite basse-orbite via l'utilisation de communications optiques. Nous nous plaçons dans un contexte de missions d'observation où les images acquises par les satellites sont compressées à bord avec un taux variable, ce qui empêche de connaître à l'avance leurs tailles (i.e., au sol). De plus, la présence de nuages au dessus des stations durant les fenêtres de visibilité étant incertaine, des prédictions météorologiques doivent être utilisées. Afin de palier aux incertitudes liées au taux de compression variable et à la présence de nuages, nous considérons de nouvelles procédures de planification différées, effectuées à bord du satellite.

L'approche que nous proposons dans cette thèse est une approche flexible dans laquelle le calcul des fenêtres de visibilité à utiliser est réalisé au sol, avec comme objectif la maximisation de la capacité de vidage, alors que le calcul de l'affectation et de l'ordonnancement des vidages d'acquisitions à l'intérieur des fenêtres est réalisé à bord du satellite, avec pour objectif la maximisation du profit lié aux vidages réussis. Pour déterminer les fenêtres de visibilité permettant de maximiser la capacité de vidage, nous proposons une modélisation basée sur la programmation linéaire en nombre entiers avec des contraintes intégrant des fonctions linéaires par morceaux pour représenter les volumes de données pouvant être transférées. Pour les calculs effectués à bord du satellite, nous proposons des heuristiques gloutonnes permettant de résoudre les problèmes d'affectation et d'ordonnancement considérés avec les ressources disponibles à bord. Différentes instances de test sont générées sur la base de scénarios réalistes que nous étendons afin d'évaluer le comportement des méthodes de résolution proposées. La comparaison entre notre approche flexible et des approches plus standards, où les plans de vidage sont calculés au sol pour être ensuite exécutés à bord du satellite, montre que, avec autant d'incertitudes, il est nécessaire d'avoir des procédures réactives à bord pour ne pas sous-utiliser le système en recourant à l'utilisation de marges.

Les approches proposées et les résultats obtenus dans cette thèse reposent sur un ensemble d'hypothèses métiers et sur des données industrielles qu'il est néces-

Appendix C. Résumé en français

saire d'affiner, comme par exemple, l'étude de contextes multi-satellites, la prise en compte des coûts d'ouverture de stations de réception, l'intégration plus fine de contraintes de chevauchement des fenêtres de visibilité ou encore la prise en compte de communications optiques multi-canaux.

Cette thèse met en avant les bénéfices que peut apporter l'utilisation de communications optiques en espace libre pour le transfert d'images haute-résolution dans les systèmes d'observation de la Terre. La création de réseaux de stations de réception efficacement réparties autour du globe, ainsi que l'utilisation de procédures effectuées à bord des satellites semblent être les éléments clés qui permettront aux communications optiques de trouver leur place au sein des applications industrielles de demain.

Bibliography

- [Agnèse 1995] J.-C. Agnèse and E. Bensana. *Exact and approximate methods for the daily management of an earth observation satellite*. In Proceedings of the 5th ESA Workshop on Artificial Intelligence and Knowledge Based Systems for Space, 1995. (Cited in page 29.)
- [Albers 1999] S. Albers. *Better bounds for online scheduling*. SIAM Journal on Computing, vol. 29, no. 2, pages 459–473, 1999. (Cited in page 32.)
- [Albers 2003] S. Albers. *Online algorithms: a survey*. Mathematical Programming, vol. 97, no. 1-2, pages 3–26, 2003. (Cited in page 31.)
- [Alliss 2012] R. J. Alliss and B. Felton. *The mitigation of atmospheric impacts on free-space optical communications*. In Proceedings of the International Conference on Space Optical Systems and Applications (ICSOS), October 2012. (Cited in pages 23 and 26.)
- [Atamtürk 2007] A. Atamtürk and M. Zhang. *Two-stage robust network flow and design under demand uncertainty*. Operations Research, vol. 55, no. 4, pages 662–673, 2007. (Cited in page 33.)
- [Augenstein 2014] S. Augenstein. *Optimal scheduling of earth-imaging satellites with human collaboration via directed acyclic graphs*. In Proceedings of the AAAI Spring Symposium on the Intersection of Robust Intelligence and Trust in Autonomous Systems, 2014. (Cited in page 29.)
- [Beale 1969] E. M. L. Beale and J. A. Tomlin. *Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables*. Operational Research, vol. 69, pages 447–454, 1969. (Cited in page 79.)
- [Ben-Tal 2000] A. Ben-Tal and A. Nemirovski. *Robust solutions of linear programming problems contaminated with uncertain data*. Mathematical programming, vol. 88, no. 3, pages 411–424, 2000. (Cited in page 32.)
- [Benoist 2004] T. Benoist and B. Rottembourg. *Upper bounds for revenue maximization in a satellite scheduling problem*. Quarterly Journal of the Belgian, French and Italian Operations Research Societies, vol. 2, no. 3, pages 235–249, 2004. (Cited in page 29.)
- [Bensana 1996] E. Bensana, G. Verfaillie, J. C. Agnèse, N. Bataille and D. Blumstein. *Exact and inexact methods for daily management of earth observation satellite*. In Proceedings of the 4th International Symposium on Space Mission Operations and Ground Data Systems (SpaceOps), 1996. (Cited in page 29.)

Bibliography

- [Bensana 1999] E. Bensana, G. Verfaillie, C. Michelon-Edery and N. Bataille. *Dealing with uncertainty when managing an earth observation satellite*. European Space Agency Publications (ESA SP), vol. 440, pages 205–210, 1999. (Cited in page 30.)
- [Beraldi 2002a] P. Beraldi and A. Ruszczyński. *A branch and bound method for stochastic integer problems under probabilistic constraints*. Optimization Methods and Software, vol. 17, no. 3, pages 359–382, 2002. (Cited in page 34.)
- [Beraldi 2002b] P. Beraldi and A. Ruszczyński. *The probabilistic set-covering problem*. Operations Research, vol. 50, no. 6, pages 956–967, 2002. (Cited in page 34.)
- [Bertsimas 2004] D. Bertsimas and M. Sim. *The price of robustness*. Operations research, vol. 52, no. 1, pages 35–53, 2004. (Cited in page 32.)
- [Bianchessi 2007] N. Bianchessi, J.-F. Cordeau, J. Desrosiers, G. Laporte and V. Raymond. *A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites*. European Journal of Operational Research (EJOR), vol. 177, no. 2, pages 750–762, 2007. (Cited in page 29.)
- [Bianchessi 2008] N. Bianchessi and G. Righini. *Planning and scheduling algorithms for the COSMO-SkyMed constellation*. Aerospace Science and Technology, vol. 12, no. 7, pages 535–544, 2008. (Cited in page 30.)
- [Birge 2011] J. R. Birge and F. Louveaux. Introduction to stochastic programming. Springer Science & Business Media, 2011. (Cited in page 33.)
- [Boche-Sauvan 2014] L. Boche-Sauvan, B. Cabon, M.-J. Huguet and E. Hébrard. *Heuristic methods for test sequencing in telecommunication satellites*. In Proceedings of the 10th International Conference on Modelling, Optimization and Simulation (MOSIM), 2014. (Cited in page 21.)
- [Böckenhauer 2014] H.-J. Böckenhauer, D. Komm, R. Královič and P. Rossmanith. *The online knapsack problem: advice and randomization*. Theoretical Computer Science, vol. 527, pages 61–72, 2014. (Cited in page 32.)
- [Camarero 2012] R. Camarero, X. Delaunay and C. Thiebaut. *Fixed-quality/variable bit-rate on-board image compression for future CNES missions*. In Proceedings SPIE 8514, Satellite Data Compression, Communications, and Processing VIII, 2012. (Cited in page 72.)
- [Camino 2014] J.-T. Camino, S. Mourgues, C. Artigues and L. Houssin. *A greedy approach combined with graph coloring for non-uniform beam layouts under antenna constraints in multibeam satellite systems*. In Proceedings of the

-
- 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014. (Cited in page 21.)
- [Camino 2016] J.-T. Camino, C. Artigues, L. Houssin and S. Mourgues. *Mixed-integer linear programming for multibeam satellite systems design: application to the beam layout optimization*. In Proceedings of the IEEE Systems Conference (SysCon), 2016. (Cited in page 21.)
- [Capelle 2017] M. Capelle, M.-J. Huguet, N. Jozefowicz and X. Olive. *A hierarchical approach for the selection of optical ground stations maximizing the data transfer from low-earth observation satellites*. In Proceedings of the IEEE International Conference on Communications (IEEE-ICC), May 2017. (Cited in pages 2 and 38.)
- [Capelle 2018a] M. Capelle, M.-J. Huguet, N. Jozefowicz and X. Olive. *Ground stations networks for free-space optical communications: maximizing the data transfer*. Electronic Notes in Discrete Mathematics (ENDM), vol. 64, pages 255–264, February 2018. (Cited in pages 2 and 38.)
- [Capelle 2018b] M. Capelle, M.-J. Huguet, N. Jozefowicz and X. Olive. *Optimizing ground station networks for free space optical communications: maximizing the data transfer*. Submitted for Networks (3rd revision), 2018. (Cited in pages 2 and 38.)
- [Chen 2015] E. Chen, H. Mei, C. Zhang and C. Chang. *The link availability analysis of GEO satellite-to-ground laser communication*. In Proceedings SPIE 9619, 2015 International Conference on Optical Instruments and Technology: Optoelectronic Devices and Optical Signal Processing, August 2015. (Cited in page 29.)
- [Coffman Jr 1980] E. G. Coffman Jr, K. So, M. Hofri and A. C. Yao. *A stochastic model of bin-packing*. Information and Control, vol. 44, no. 2, pages 105–115, 1980. (Cited in page 33.)
- [Cordeau 2005] J.-F. Cordeau and G. Laporte. *Maximizing the value of an earth observation satellite orbit*. Journal of the Operational Research Society, vol. 56, no. 8, pages 962–968, 2005. (Cited in page 29.)
- [Couble 2017] Y. Couble, E. Chaput, T. Deleu, C. Baudoin, J.-B. Dupé, C. Bés and A.-L. Beylot. *Interference-aware frame optimization for the return link of a multi-beam satellite*. In Proceedings of the IEEE International Conference on Communications (IEEE-ICC), May 2017. (Cited in page 21.)
- [Dean 2004] B. C. Dean, M. X. Goemans and J. Vondrck. *Approximating the stochastic knapsack problem: the benefit of adaptivity*. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004. (Cited in page 33.)

Bibliography

- [Dee 2011] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut and F. Vitart. *The ERA-Interim reanalysis: Configuration and performance of the data assimilation system*. Quarterly Journal of the Royal Meteorological Society, vol. 137, no. 656, pages 553–597, 2011. (Cited in pages 16 and 61.)
- [del Portillo 2017] I. del Portillo, M. Sanchez, B. Cameron and E. Crawley. *Optimal location of optical ground stations to serve leo spacecraft*. In Proceedings of the IEEE Aerospace Conference, 2017. (Cited in page 27.)
- [Franz 2000] J. Franz, V. K. Jain and V. K. Jain. *Optical communications: Components and systems*. CRC press, 2000. (Cited in pages vii and 11.)
- [Fuchs 2015] C. Fuchs and F. Moll. *Ground station network optimization for space-to-ground optical communication links*. IEEE/OSA Journal of Optical Communications and Networking, vol. 7, pages 1148–1159, December 2015. (Cited in pages 24 and 28.)
- [Gabrel 2003] V. Gabrel and C. Murat. *Mathematical programming for Earth observation satellite mission planning*. Operations Research in Space and Air, pages 103–122, 2003. (Cited in page 29.)
- [Gabrel 2010] V. Gabrel, C. Murat, N. Remli and M. Lacroix. *Recourse problem of the 2-stage robust location transportation problem*. Electronic Notes in Discrete Mathematics (ENDM), vol. 36, pages 167–174, 2010. (Cited in page 33.)
- [Gabrel 2013] V. Gabrel, C. Murat and L. Wu. *New models for the robust shortest path problem: complexity, resolution and generalization*. Annals of Operations Research, vol. 207, no. 1, pages 97–120, 2013. (Cited in page 32.)
- [Gabrel 2014] V. Gabrel, C. Murat and A. Thiele. *Recent advances in robust optimization: an overview*. European Journal of Operational Research (EJOR), vol. 235, no. 3, pages 471–483, 2014. (Cited in page 32.)
- [Gelaro 2017] R. Gelaro, W. McCarty, M. J. Suárez, R. Todling, A. Molod, L. Takacs, C. A. Randles, A. Darmenov, M. G. Bosilovich, R. Reichle, K. Wargan, L. Coy, R. Cullather, C. Draper, S. Akella, V. Buchard, A. Conaty, A. M. da Silva, W. Gu, G.-K. Kim, R. Koster, R. Lucchesi, D. Merkova, J. E. Nielsen, G. Partyka, S. Pawson, W. Putman, M. Rienecker, S. D. Schubert, M. Sienkiewicz and B. Zhao. *The Modern-Era*

-
- Retrospective Analysis for research and applications, version 2 (MERRA-2)*. Journal of Climate, vol. 30, no. 14, pages 5419–5454, 2017. (Cited in page 17.)
- [Giggenbach 2015] D. Giggenbach, E. Lutz, J. Poliak, R. Maa-Calvo and C. Fuchs. *A High-Throughput Satellite System for Serving whole Europe with Fast Internet Service, Employing Optical Feeder Links*. In Proceedings of Broadband Coverage in Germany, 9th ITG Symposium, April 2015. (Cited in page 28.)
- [Globus 2003] A. Globus, J. Crawford, J. Lohn and A. Pryor. *Scheduling earth observing satellites with evolutionary algorithms*. In Proceedings of the International Conference on Space Mission Challenges for Information Technology, 2003. (Cited in page 29.)
- [Globus 2004] A. Globus, J. Crawford, J. Lohn and A. Pryor. *A comparison of techniques for scheduling earth observing satellites*. In Proceedings of the AAAI Conference, 2004. (Cited in page 29.)
- [Graham 1966] R. L. Graham. *Bounds for certain multiprocessing anomalies*. Bell Labs Technical Journal, vol. 45, no. 9, pages 1563–1581, 1966. (Cited in page 32.)
- [Gregory 2013] M. Gregory, D. Troendle, G. Muehlnikel, F. Heine, R. Meyer, M. Lutzer and R. Czichy. *Three years coherent space to ground links: performance results and outlook for the optical ground station equipped with adaptive optics*. In Proceedings SPIE 8610, Free-Space Laser Communication and Atmospheric Propagation XXV, 2013. (Cited in page 10.)
- [Grishechkin 2012a] B. Grishechkin, A. Braun and M. Wickler. *Clouds handling for planning of optical space missions*. In Proceedings of the 12th International Conference on Space Operations (SpaceOps), 2012. (Cited in page 18.)
- [Grishechkin 2012b] B. Grishechkin, A. Braun and M. Wickler. *Optimization of Positioning of Ground Stations for Space Optical Missions*. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July 2012. (Cited in pages 16, 25 and 27.)
- [Grötschel 2001] M. Grötschel, S. O. Krumke and J. Rambau. *Online optimization of complex transportation systems*. Online Optimization of Large Scale Systems, pages 705–729, 2001. (Cited in page 32.)
- [Guérin 2010a] A. Guérin, F. Lacoste, A. Laurens, J. Berthon, C. Periard and D. Grimal. *Optimisation of an optical ground station network*. In Proceedings of the 5th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TTC, ESA-ESTEC), 2010. (Cited in pages 16, 25 and 26.)

Bibliography

- [Gu erin 2010b] A. Gu erin, G. Lesthievant and J.-L. Issler. *Evaluation of new technological concepts for high data rate payload telemetry*. In Proceedings of the 5th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TTC, ESA-ESTEC), 2010. (Cited in pages 10, 24, 26 and 61.)
- [Han 2009] X. Han and K. Makino. *Online minimization knapsack problem*. Approximation and Online Algorithms, pages 182–193, 2009. (Cited in page 32.)
- [Hauschildt 2014] H. Hauschildt, F. Garat, H. Greus, K. Kably, J.-P. Lejault, H. L. Moeller, A. Murrell, J. Perdignes, D. Witting, B. Theelen, M. Wiegand and A. Hegyi. *European Data Relay System—one year to go!* In Proceedings of the International Conference on Space Optical Systems and Applications (ICSOS), 2014. (Cited in page 6.)
- [Hebrard 2017] E. Hebrard, M.-J. Huguet, D. Veysseire, L. B. Sauvan and B. Cabon. *Constraint programming for planning test campaigns of communications satellites*. Constraints, vol. 22, no. 1, pages 73–89, 2017. (Cited in page 21.)
- [Hsiao 1992] J. Y. Hsiao, C. Y. Tang and R. S. Chang. *An efficient algorithm for finding a maximum weight 2-independent set on interval graphs*. Information Processing Letters, vol. 43, no. 5, pages 229–235, October 1992. (Cited in page 47.)
- [IOAG 2012] IOAG. *Optical Link Study Group final report*. Technical Report IOAG.T.OLSG.2012.V1, Interagency Operations Advisory Group (IOAG), Optical Link Study Group (OLSG), June 2012. (Cited in pages 9, 16, 26 and 61.)
- [Kaushal 2016] H. Kaushal and G. Kaddoum. *Optical communication in space: challenges and mitigation techniques*. IEEE Communications Surveys & Tutorials, vol. 19, pages 57–96, 2016. (Cited in page 11.)
- [Keha 2004] A. B. Keha, I. R. de Farias Jr and G. L. Nemhauser. *Models for representing piecewise linear cost functions*. Operations Research Letters, vol. 32, no. 1, pages 44–48, 2004. (Cited in page 79.)
- [Kellerer 2004] H. Kellerer, U. Pferschy and D. Pisinger. *Knapsack problems*. Springer-Verlag Berlin Heidelberg, 2004. (Cited in page 82.)
- [Kernec 2014] A. L. Kernec, M. Dervin and M. Sotom. *Method and device for optical transmission at adaptive effective rates*, December 11 2014. US Patent App. 14/297,469. (Cited in page 14.)
- [Kiatmanaroj 2012] K. Kiatmanaroj, C. Artigues, L. Houssin and E. Corbel. *Greedy algorithms for time-frequency allocation in a SDMA satellite communication*

- system*. In Proceedings of the 9th International Conference on Modeling, Optimization and Simulation (MOSIM), 2012. (Cited in page 21.)
- [Kiatmanaroj 2016] K. Kiatmanaroj, C. Artigues and L. Houssin. *On scheduling models for the frequency interval assignment problem with cumulative interferences*. Engineering Optimization, vol. 48, no. 5, pages 740–755, 2016. (Cited in page 21.)
- [Lacoste 2011] F. Lacoste, A. Guérin, A. Laurens, G. Azema, C. Periard and D. Grimal. *FSO ground network optimization and analysis considering the influence of clouds*. In Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP), 2011. (Cited in pages 14, 25, 26 and 61.)
- [Liao 2005] D.-Y. Liao and Y.-T. Yang. *Satellite imaging order scheduling with stochastic weather condition forecast*. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics,, 2005. (Cited in page 30.)
- [Liao 2007] 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, pages 794–802, 2007. (Cited in page 30.)
- [Link 2005] R. Link, M. E. Craddock and R. J. Alliss. *Mitigating the impact of clouds on optical communications*. In Proceedings of the 2005 IEEE Aerospace Conference, March 2005. (Cited in pages 16, 22 and 23.)
- [Maillard 2014] A. Maillard, C. Pralet, G. Verfaillie, J. Jaubert and T. Desmousseaux. *Building flexible downloads plans for agile Earth-observing satellites*. In Proceedings of the 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), June 2014. (Cited in pages 30, 31 and 72.)
- [Maillard 2015a] A. Maillard, C. Pralet, J. Jaubert, I. Sebbag, F. Fontanari and J. l’Hermite. *Ground and on-board decision-making on satellite data downloads*. In Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS), June 2015. (Cited in pages 30, 31 and 72.)
- [Maillard 2015b] A. Maillard, G. Verfaillie, C. Pralet, J. Jaubert, I. Sebbag and F. Fontanari. *Postponing decision-making to deal with resource uncertainty on Earth-observation satellites*. In Proceedings of the 9th International Workshop on Planning and Scheduling for Space (IWSPSS), 2015. (Cited in page 30.)
- [Maillet 2011] C. Maillet, G. Verfaillie and B. Cabon. *Constraint programming for optimising satellite validation plans*. In Proceedings of the 7th Inter-

Bibliography

- national Workshop on Planning and Scheduling for Space (IWPSS), June 2011. (Cited in page 21.)
- [Morton 1998] D. P. Morton and R. K. Wood. *On a stochastic knapsack problem and generalizations*. Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search, pages 149–168, 1998. (Cited in page 33.)
- [Oddi 2003] A. Oddi, N. Policella, A. Cesta and G. Cortellessa. *Generating high quality schedules for a spacecraft memory downlink problem*. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, 2003. (Cited in page 30.)
- [Pemberton 2002] J. C. Pemberton and L. Greenwald. *On the need for dynamic scheduling of imaging satellites*. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, vol. 34, no. 1, pages 165–171, 2002. (Cited in page 29.)
- [Poulenard 2013] S. Poulenard, B. Roy, M. Hanna, F. Lacoste, H. L. Gléau and A. Rissons. *Optical ground station network optimization and performances for high data rate geosatellite-to-ground telemetry*. In Proceedings of the 6th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TTC, ESA-ESOC), 2013. (Cited in pages 16 and 28.)
- [Poulenard 2014] S. Poulenard, M. Ruellan, B. Roy, J. Riedi, F. Parol and A. Rissons. *High altitude clouds impacts on the design of optical feeder link and optical ground station network for future broadband satellite services*. In Proceedings SPIE 8971, Free-Space Laser Communications and Atmospheric Propagation XXVI, March 2014. (Cited in pages 12 and 28.)
- [Poulenard 2015] S. Poulenard, M. Crosnier and A. Rissons. *Ground segment design for broadband geostationary satellite with optical feeder link*. IEEE/OSA Journal of Optical Communications and Networking, vol. 7, pages 325–336, April 2015. (Cited in page 28.)
- [Pralet 2008] C. Pralet and G. Verfaillie. *Decision upon observations and data downloads by an autonomous earth surveillance satellite*. In Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS), 2008. (Cited in page 30.)
- [Pralet 2011] C. Pralet, G. Verfaillie, X. Olive, S. Rainjonneau and I. Sebbag. *Planning for an ocean global surveillance mission*. In Proceedings of the 7th International Workshop on Planning and Scheduling for Space (IWPSS), June 2011. (Cited in page 29.)
- [Pralet 2012] C. Pralet, G. Verfaillie, X. Olive, S. Rainjonneau and I. Sebbag. *Allocation of downlink windows for a constellation of satellites*. In Proceedings

- of the 11th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), 2012. (Cited in page 30.)
- [Pralet 2014] C. Pralet, G. Verfaillie, A. Maillard, E. Hebrard, N. Jozefowicz, M.-J. Huguet, T. Desmousseaux, P. Blank-Paques and J. Jaubert. *Satellite data download management with uncertainty about the generated volumes*. In Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS), June 2014. (Cited in pages 30, 31 and 72.)
- [Rossow 1999] W. B. Rossow and R. A. Schiffer. *Advances in understanding clouds from ISCCP*. Bulletin of the American Meteorological Society, vol. 80, no. 11, pages 2261–2288, 1999. (Cited in page 17.)
- [Roy 2010] B. Roy. *Robustness in operational research and decision aiding: A multi-faceted issue*. European Journal of Operational Research (EJOR), vol. 200, no. 3, pages 629–638, 2010. (Cited in page 32.)
- [Roynette 2016] E. Roynette, B. Cabon, C. Pralet and V. Vidal. *Designing spacecraft command loops using two-dimension vehicle routing*. In Proceedings of the 13th International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming (CPAIOR), 2016. (Cited in page 21.)
- [Schultz 1996] R. Schultz, L. Stougie and M. H. Vlerk. *Two-stage stochastic integer programming: a survey*. Statistica Neerlandica, vol. 50, no. 3, pages 404–416, 1996. (Cited in page 33.)
- [Schulz 2012] K.-J. Schulz and J. Rush. *Results of the Optical Link Study Group*. In Proceedings of the 12th International Conference on Space Operations (SpaceOps), 2012. (Cited in page 26.)
- [Simonin 2012] G. Simonin, C. Artigues, E. Hebrard and P. Lopez. *Scheduling scientific experiments on the Rosetta/Philae mission*. In Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP), 2012. (Cited in page 22.)
- [Sodnik 2012] Z. Sodnik and M. Sans. *Extending EDRS to laser communication from space to ground*. In Proceedings of the International Conference on Space Optical Systems and Applications (ICSOS), October 2012. (Cited in page 12.)
- [Soyster 1973] A. L. Soyster. *Convex programming with set-inclusive constraints and applications to inexact linear programming*. Operations research, vol. 21, no. 5, pages 1154–1157, 1973. (Cited in page 32.)
- [Stampfl 1970] R. A. Stampfl and A. E. Jones. *Tracking and data relay satellites*. IEEE Transactions on Aerospace and Electronic Systems, no. 3, pages 276–289, 1970. (Cited in page 6.)

Bibliography

- [Stubenrauch 2013] C. J. Stubenrauch, W. B. Rossow, S. Kinne, S. Ackerman, G. Cesana, H. Chepfer, L. Di Girolamo, B. Getzewich, A. Guignard, A. Heindinger, B. C. Maddux, W. P. Menzel, P. Minnis, C. Pearl, S. Platnick, C. Poulsen, J. Riedi, S. Sun-Mack, A. Walther, D. Winker, S. Zeng and G. Zhao. *Assessment of global cloud datasets from satellites: project and database initiated by the GEWEX radiation panel*. Bulletin of the American Meteorological Society, vol. 94, no. 7, pages 1031–1049, 2013. (Cited in page 16.)
- [Takayama 2010] Y. Takayama, M. Toyoshima and N. Kura. *Estimation of accessible probability in a low-Earth orbit satellite to ground laser communications*. Radioengineering, vol. 19, no. 2, pages 249–253, June 2010. (Cited in pages 23 and 24.)
- [Tangpattanakul 2013] P. Tangpattanakul, N. Jozefowicz and P. Lopez. *Biased random key genetic algorithm with hybrid decoding for multi-objective optimization*. In Proceedings of the 2013 Federated Conference on Computer Science and Information Systems (FedCSIS), 2013. (Cited in page 29.)
- [Tangpattanakul 2015] 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 (EJOR), vol. 245, no. 2, pages 542–554, 2015. (Cited in page 29.)
- [Verfaillie 2011] G. Verfaillie, G. Infantes, M. Lemaître, N. Théret and T. Natolot. *On-board decision-making on data downloads*. In Proceedings of the 7th International Workshop on Planning and Scheduling for Space (IWSPSS), June 2011. (Cited in page 30.)
- [Verfaillie 2012] G. Verfaillie, X. Olive, C. Pralet, S. Rainjonneau and I. Sebbag. *Planning acquisitions for an ocean global surveillance mission*. In Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), 2012. (Cited in page 29.)
- [Wang 2016] 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, pages 1–13, 2016. (Cited in page 30.)
- [Wielicki 1996] B. A. Wielicki, B. R. Barkstrom, E. F. Harrison, R. B. Lee III, G. L. Smith and J. E. Cooper. *Clouds and the Earth’s Radiant Energy System (CERES): An earth observing system experiment*. Bulletin of the American Meteorological Society, vol. 77, no. 5, pages 853–868, 1996. (Cited in page 17.)
- [Wojcik 2005] G. S. Wojcik, H. L. Szymczak, R. J. Alliss and R. P. Link. *Deep-space to ground laser communications in a cloudy world*. In Proceedings

- SPIE 5892, Free-Space Laser Communications V, August 2005. (Cited in page 23.)
- [Yu 1998] G. Yu and J. Yang. *On the robust shortest path problem*. Computers & Operations Research, vol. 25, no. 6, pages 457–468, 1998. (Cited in page 32.)
- [Yu 2000] C.-S. Yu and H.-L. Li. *A robust optimization model for stochastic logistic problems*. International Journal of Production Economics, vol. 64, no. 1-3, pages 385–397, 2000. (Cited in page 33.)
- [Zech 2015] H. Zech, F. Heine, D. Tröndle, S. Seel, M. Motzigemba, R. Meyer and S. Philipp-May. *LCT for EDRS: LEO to GEO optical communications at 1.8 Gbps between Alphasat and Sentinel-1A*. In Proceedings SPIE 9647, Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications, 2015. (Cited in page 9.)

Abstract: For many years, satellites have been used to explore, analyse and monitor Earth from space. Nowadays, low-earth orbiting satellites are taking a steadily increasing number of pictures, with higher and higher precision, thus generating huge volumes of data that needs to be downloaded. In order to manage such amount of data, new communication technologies are investigated. In this context, free-space optical communications are seen as a key alternative to current radio-frequency ones. With their orders of magnitude higher data rates, free-space optical communications could be able to download daily acquisition volumes in a matter of minutes. Unfortunately, optical communications are strongly impacted by atmospheric turbulence and clouds. In order to cope with the latter, it is necessary to design efficient distributed networks of optical ground stations. Furthermore, since weather forecasts are imperfect, custom operational procedures should be established to schedule the downloads of acquisitions using optical communications to take into account uncertainties regarding clouds. In this thesis, we investigate the use of free-space optical communications for image telemetry. We first consider the problem of designing efficient networks of optical ground stations taking into account the impact of clouds. We solve this problem using a combination of a dynamic programming algorithm and a branch-and-bound algorithm. Then, we look at the operational problem of planning downloads of acquisitions under download channel and image volume uncertainties. We propose a flexible approach where the decision process is shared between ground and on-board systems.

Keywords: Combinatorial Optimization, Spatial Imagery, Free-Space Optical Communications

Résumé : Depuis plusieurs années, les satellites sont utilisés pour explorer, surveiller et photographier la Terre depuis l'espace. Ces satellites prennent un nombre croissant d'images avec une résolution de plus en plus élevée, générant ainsi de gros volumes de données qui doivent être envoyés vers la Terre. Afin de transférer de telles quantités de données, de nouvelles technologies de communication sont étudiées. Parmi elles, les liaisons optiques en espace libre sont considérées aujourd'hui comme une alternative intéressante aux technologies actuelles basées sur les radio-fréquences. Avec des débits bien supérieurs à ceux actuels, les liaisons optiques sont capables de transférer d'important volumes de données journaliers en à peine quelques minutes. Cependant, les liaisons optiques sont fortement perturbées par les turbulences atmosphériques et, en particulier, les nuages. Afin de limiter l'impact de la couverture nuageuse, il est nécessaire de créer des réseaux de stations sol optiques efficacement réparties dans le monde. De plus, les prévisions météorologiques n'étant pas exactes, des procédures opérationnelles doivent être établies afin de pouvoir planifier les transferts d'images depuis le satellite vers la Terre, en prenant en compte les incertitudes liées aux nuages et aux prévisions de couverture nuageuse. Dans cette thèse, nous nous intéressons à l'utilisation de communications optiques pour la télémétrie image. Dans un premier temps, nous considérons le problème de la création de réseaux terrestres de stations optiques prenant en compte l'impact des nuages. Nous résolvons ce problème en combinant un algorithme de programmation dynamique avec un algorithme de type branch-and-bound. Dans un second temps, nous nous intéressons au problème de planification des transferts d'images d'un point de vue opérationnel, en prenant en compte les incertitudes sur les volumes des images acquises (dues à la compression bord) et celles liées aux communications optiques. Nous proposons une approche flexible où le processus de décision est partagé entre le système sol (centre mission) et le système à bord du satellite.

Mots-clés : Optimisation Combinatoire, Observation Spatiale, Communications Optiques
