UNIVERSITE BRETAGNE LOIRE

université angers

Thèse de Doctorat

Aurélien FROGER

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université d'Angers
sous le sceau de l'Université Bretagne Loire

École doctorale : Sciences et technologies de l'information, et mathématiques

Discipline : Informatique et applications, section CNU 27, 61
Unité de recherche : Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS)

Soutenue le 14 décembre 2016

Maintenance scheduling in the electricity industry: a particular focus on a problem rising in the onshore wind industry

JURY

Rapporteurs :	**M. Jacques CARLIER**, Professeur des universités, Université de Technologie de Compiègne **M. Christian ARTIGUES**, Directeur de recherche, LAAS-CNRS
Examinateurs :	**M. Jean-Charles BILLAUT**, Professeur des universités, École Polytechnique de l'Université de Tours **M. David RIVREAU**, Professeur, Institut de Mathématiques Appliquées d'Angers
Directeur de thèse :	**M. Éric PINSON**, Professeur, Institut de Mathématiques Appliquées d'Angers
Co-encadrants de thèse :	**M. Jorge E. MENDOZA**, Maître de conférences, École Polytechnique de l'Université de Tours **M. Louis-Martin ROUSSEAU**, Professeur titulaire, École Polytechnique de Montréal **M. Michel GENDREAU**, Professeur titulaire, Ecole Polytechnique de Montréal

Acknowledgments

This thesis has become reality with the kind support of many individuals.

The path I was following some years ago was definitely not leading me pursuing PhD studies. Therefore, I would like to particularly thank Professors Eric Pinson and Jorge E. Mendoza for having believed in me when I came back for my master's degree, and for having convinced me that I could become a researcher. I am grateful to them for making me passionate for Operations Research.

I would like now to express my gratitude to my group of advisors. Without their precious support the conducted research would not have been possible. I am sincerely grateful to them for putting their trust in me during the 3 years of this thesis when it came to define the main research orientations. I thank Eric for all his invaluable comments on my work. The many discussions and debates we had have been very instructive. I thank Jorge for his encouragement and his valuable advice on both my research and my career. I am also very thankful for the help he provides me, all along this thesis, to improve my writing skills in English. I also thank Professors Michel Gendreau and Louis Martin Rousseau who provided me with the opportunity to spend one year and a half at Polytechnique Montreal. It has been an exceptional experience to work with them.

I thank very sincerely Pr. Jacques Carlier and Christian Artigues for having accepted to be "rapporteur" of my thesis. I also thank Pr. Jean-Charles Billaut and David Rivreau for having accepted to be part of the jury.

I also wish to express my gratitude to the WPred team (Christophe, Thomas, Hugo, Jérôme, François) for this enriching collaboration. I am very thankful to them for having taken the time to explain me their work with the wind industry and having kindly accepted me as a member of the team. I have greatly appreciated this experience that has enabled this work to be really connected to what is happening on the ground. This collaboration has been possible thanks to the grant program of the Natural Sciences and Engineering Research Council of Canada – this support is gratefully acknowledged. I would also like to acknowledge the financial support from Angers Loire Metropole.

I also thank Julien, Maria, Vinicius, Selene, Philippe, Diego, Shohre, Andrea, Slavic, Maëlle, Ragheb, Pirmin, and all my fellow labmates in CIRRELT for the stimulating discussions and also all the fun we have had. Thank you for having created an enjoyable environment during my stay in Montreal.

Finally, I thank my parents for their unanimous support of my choices, whatever they were.

Contents

Introduction

This thesis discusses the optimization of maintenance scheduling in the electricity industry and particularly focuses on a problem rising in the onshore wind industry. This introductory chapter aims to make the reader understand the positioning, the motivation, and the relevance of this work.

Nowadays, the electricity industry experiences major challenges and is currently in a transitional phase in various respects. The opening of the electricity market to competition, following the deregulation of the sector, has notably triggered fundamental changes. Electricity prices are no longer essentially regulated by the government (hence the terms deregulation and liberalization), but they are subject to market interactions. Given the success of this system in the aeronautics and telephone industries, this reform is promoted as a benefit for the sector. It is intended to favor innovation, to lower prices, and to lead to better services. But the fact is that the transition from the original monopoly (state) system is a slow and lengthy process. Indeed, this new system introduces new actors and redefines the role or activities of the existing ones in such a way that this raises new issues, especially on the organization and the regulation of the market. Along with this change, the sector is steadily growing. With the increasing world population, the development of countries, the overall needs for energy services for the world is predicted to be multiplied by 3.2 between 2000 and 2100 (European Commission, 2011). If we exclusively focus on the electricity industry, the demand is predicted to increase at a rate of 2% per year until 2035 (European Commission, 2011). In a market driven economy, electric companies are turned towards cost-management and profitability. Therefore, in this growth context, they tend to pledge for more and/or reliable electricity to maintain their position on the market or to gain market share. Predominantly, companies in the sector consistently aim to be able to meet the demand they committed to produce with a high degree of reliability while being cost-effective for suppliers. As a long-term development strategy, they may also choose to built more generating units or to increase the capacity of those existing if the return on investments is worth it. The different electricity generating technologies then present various investments options to those companies.

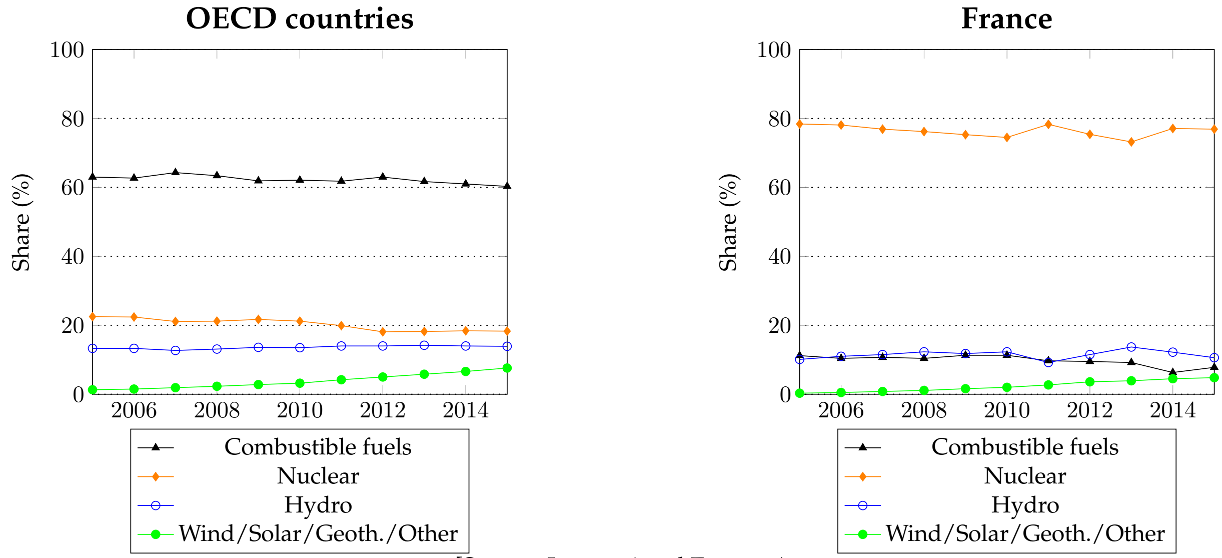
As it is widely known, electricity can be produced from fuel (e.g., oil, gasoline, uranium, gas, coal, wood) or natural forces (e.g., sun, wind, water, biomass, geothermal). The costs incurred in producing electricity and the revenue generated from its sale vary depending on the generating technologies. Take the example of nuclear power plants; the construction costs excesses by far the fuel (uranium and heavy water), operating and maintenance costs (even if these latter costs are significant). On the contrary, fossil fuel power plants (coal, oil or natural gas) are less expensive to build, but the fuel costs are more important and subject to rising or volatility. Similar to nuclear plants, hydroelectric power stations have much higher fixed costs than fossil fuel technologies but, once installed, tend to require less maintenance and last much longer than their nuclear and fossil-fuel counterpart. The remarkable advantage of hydroelectricity is the elimination of fuel costs. The same holds for wind farms and photovoltaic (PV) power stations for which the costs are essentially composed of the building costs and some maintenance costs. Nonetheless, for the amount of electricity they can generate, renewable energy technologies are quite often more expensive. One should not ignored that the renewable sources of power are usually located in the desert, the mountains, or off-shore, that is far away from the location of the large proportion of the demand. New transmission lines are therefore required to connect to the existing network, and these building costs are significant. However, in the past decade, the investment and exploitation costs of renewable energy technologies have already drastically reduced to make them competitive alternatives, especially if we also take into account tax breaks and incentives that have been set up in many countries. Nonetheless, performance and efficiency of electricity generating technologies still remain the most critical point when speaking about profitability. If one looks at the availability factor (AF) of generating units, renewable energy sounds attractive. Indeed, while gas, coal, and nuclear plants carry AF over 80%, often around 90%, wind farms AF top 95%, PV power plants AF reach over 98%, and hydro power plants AF stays above 90% every summer (but fell as low as 75% in fall and spring). However, high AF does not translate into a full capacity power generation. For instance, it is not always possible to release all the water needed to reach the maximum power. In the same way, in the middle of the night, during cloudy days, or during winter, an available PV power plant will not generate any output or a very small one. Similarly, a wind farm will not generate power if the wind speed is too low. In order to assess the effectiveness of the electricity generating technologies, the most relevant criterion is the capacity factor (CF). Figures are then clearly different. Nuclear plants have a CF usually between 80% and 90% , gas and coal plants around 50-60%, biomass around 80%, geothermal around 60%, hydro power plants around 40% and wind farms around 25-30%, while PV power stations barely reach 15-20% [1]. However, the choice of an electricity generating technology over another cannot be made solely based on this factor.

Taking a look at the evolution of the electricity production per fuel type from 2005 to 2015 for the countries part of the Organization for Economic Co-operation and Development (OECD) and for the particular case of France (see Figure 1), we observe that the share of renewable energy sources has significantly increased, whereas the share of combustible fuels (coal, gas) and nuclear have slightly reduced in the past decade. Although CF figures and the profit-driven nature of companies raise legitimate questions about renewable energy sources (as pointed out above), the unprecedented boom experienced by renewable energy is in fact explained by the increasing development of policies to reduce (if not cut) greenhouse gas emissions. Driven by climate change mitigation and adaptation measures (e.g., the tax incentives previously mentioned), the renewable energy sector is called to keep growing as producing low-carbon power or carbon-free electricity becomes the priority. The Paris Agreement – resulting from the 2015 United Nations Climate Change Conference (COP21) – is in this respect a clear evidence supporting this claim. Nuclear and/or natural gas power stations (called to replace coal power plants since they produce twice less CO_2 emissions) are then usually predicted to compensate the intermittency of renewable energy as long as energy storage devices are not fully developed and applicable on a wide scale. It is therefore clear that the transition to a low carbon society by 2050 has already a significant impact on the electricity industry, and this impact will continue to grow.

This brief overview of the energy sector points out the very important challenges faced nowadays by the electricity industry to reconcile the quest of profits – which grew out with the policy of opening up competition – with environmental sustainability.

The issue of profitability is naturally, although not exclusively, linked to the question of reliability. Indeed, electrical companies aim to avoid costly unexpected breakdowns and try to minimize the downtime that follows. Maintenance management is therefore a major economic issue as it saves some investment costs with the life extension of the generating units and prevents unnecessary downtime and excessive operational costs. Just to cite few examples, equipment maintenance management in electric power systems is concerned with decisions such as: when to stop a generation unit for maintenance, according to what criteria, and when to re-start it again. These decisions are taken under complex environments and constraints such as resource availability, demand satisfaction, and reliability thresholds. They aim to build effective and efficient business strategies based on revenue or profits maximization. Maintenance in the electricity industry is therefore an ongoing challenge. For clarification purposes, maintenance represents the actions required to ensure that a generating unit provides reliable service. Maintenance is generally split into two categories according to the nature of the actions, whether they are proactive or reactive. On the one hand, as a reactive approach, *corrective maintenance* is performed after a breakdown in order to restore the serviceability

1. U.S. Energy Information Administration - Electric generator capacity factors vary widely across the world, http://www.eia.gov/todayinenergy/detail.cfm?id=22832



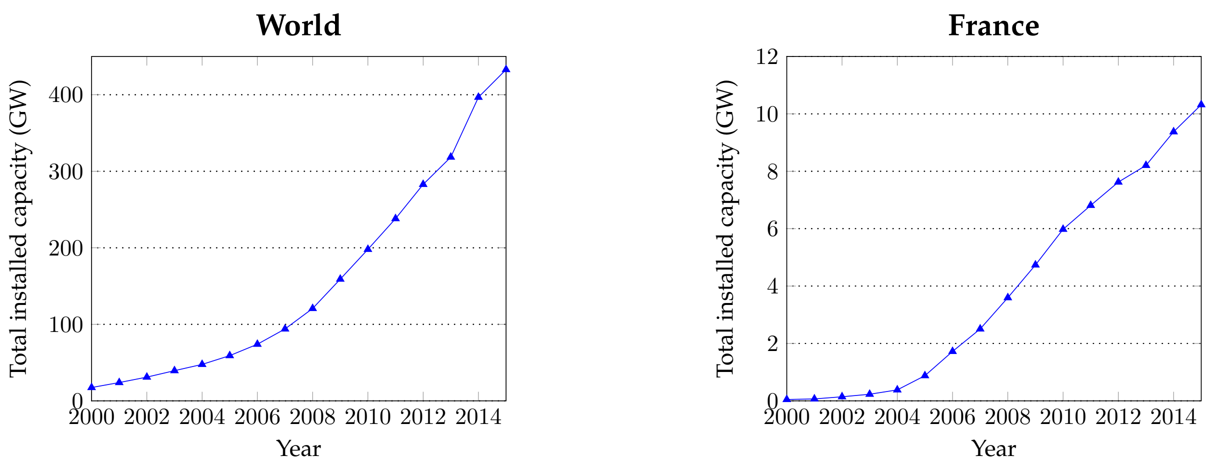
[Source: International Energy Agency
http://www.iea.org/statistics/monthlystatistics/monthlyelectricitystatistics]

Figure 1 – Electricity production per fuel type

of the generating unit. On the other hand, *preventive maintenance* intends, as a proactive approach, to reduce the probability of failure. It is performed at predetermined intervals (based on a calendar or a routine system) or according to some prescribed criteria usually coupled with real-time monitoring. In the last case, the maintenance is usually referred as *predictive maintenance* or *condition-based maintenance*. Basically, companies are interested in increasing the probability that a generating unit will function in the required manner over its design life cycle with a minimum of maintenance. The process of determining the most effective maintenance approach refers to the concept of *reliability-centered maintenance* (RCM). In today's competitive world economy, RCM translates obviously into savings. Questions such as why failures happens, what should be done when they happen, and how to predict or prevent each failure should be answered by the framework to be implemented. Two noticeable approaches can be jointly considered. On the one hand, topics such as failure prediction or maintenance policies that manage the risks of generating unit failure in the most effective way have been widely studied. The results of these studies help to identify the best time windows for maintenance operations and allow to define maintenance service contracts (e.g., preventive maintenance has to be performed every six months on each turbine). However, theses studies fall out of the scope of this thesis. We refer the reader to Doyle (2004) and Yssaad et al. (2014) for literature dealing with it. On the other hand, electrical companies are interested in precisely defining time intervals for the preventive maintenance of generating units under financial (cost minimization, profits maximization) and/or reliability (leveling, maximization of the net reserves) considerations. As a very common practice in the *Operations Research* (OR) studies focusing on the topic, the first approach is then often assumed to part of an upstream work, as it provides valuable inputs to set the maintenance time window constraints for each generating unit. Thereafter, maintenance scheduling optimization refers to this last approach.

The issue of environmental sustainability leads us to concentrate on the renewable energy industry. As it generates very little pollution and hazardous waste, it therefore provides a valuable alternative to non-renewable fossil fuels. With a 63-Gigawatts (GW) increase in the global installed capacity in 2015 (and a total of about 432 GW), wind energy is currently the world's fastest-growing source of electricity (The Global Wind Energy Council, 2016). Figure 2 illustrates the evolution of the wind installed capacity in the world and in France during the past 15 years. Boosted by the ever-increasing environment awareness and the constantly-decreasing cost of turbines, wind power is expected to account for up to 20% of the global electricity production by 2050 (The Global Wind Energy Council, 2016) (vs. 2.4% in 2015). It accounts already for around 10% of the European Union

(UE) electricity production (3.9% in France). With an eye to the future, focusing on the wind industry is therefore particularly relevant. In this context of rapid growth, the cost-effectiveness of wind energy exploitation is essential. Since the maintenance costs account for a significant part of the total cost incurred by the wind industry, there is an incentive to optimize the maintenance process in wind farms. Piloting this process in the best possible way demands a deep understanding of turbine performance. Wind farms are usually equipped with a supervisory control and data acquisition (SCADA) system. The data provided by this system may be used, along with mathematical models, to diagnose performance issues and risks of failure. This input is then used to define time windows where the preventive maintenance must be scheduled. As mentioned before, this is beyond the scope of this thesis. Nonetheless, the problem still contains a significant degree of operational flexibility. Defining a preventive maintenance plan on a short-term horizon (from a few days to two weeks) is a relevant challenge for the OR field. Contrary to the maintenance of generating units in traditional power plants that is primarily based on electricity price and/or demand satisfaction considerations, maintenance of wind turbines should be scheduled in a way that it minimizes the loss of production. Indeed, although wind turbine CF is primarily the result of the wind power intermittency and of design decisions (for a fixed wind speed, the larger are the blades the more electricity the turbine can produce), the impact of operational decisions (among them the maintenance) on its value is also non negligible due to the uncontrollable nature of the wind. Producing a maintenance plan in which no operations generate a loss of production (e.g., every task is scheduled during time periods where the wind speed is below 3.5 m.s^{-1}, which is too low to produce electricity) can almost never be achieved in practice, since human resources are a major bottleneck. In this way, focusing on the electricity production while scheduling the maintenance helps to guarantee the highest possible CF, which essentially leads to higher revenue. Finding efficient strategies for this problem is a challenge that has motivated this work.



[Source: (The Global Wind Energy Council, 2016)]

[Source: RTE France - Bilan électriques nationaux `http://www.rte-france.com/fr/article/bilans-electriques-nationaux`]

Figure 2 – Wind installed capacity between 2000 and 2015

The purpose of this thesis is first to analyze the OR studies on the planning of maintenance operations in the electricity industry and then to expand this research to a relatively unexplored area: the wind industry.

Two different fundings have enabled to produce this thesis. First, this thesis is partially funded by the research grant program of Angers Loire Metropole. This funding has been provided to address problems faced by the electricity industry on the maintenance scheduling of generating units and to investigate the impact of data uncertainty on these problems. The discussions above demonstrate a major interest to tackle these issues. Nonetheless, the topic being large, there was a need to frame the subject and to focus on a specific topic. While the knowledge and understanding of the above-mentioned main tenets were becoming increasingly clear, it turns out that we were approached by a Canadian company called WPred [2]. This company is specialized in the supply of weather forecasts and, more specifically, on power production forecasts for wind and solar energy. Wind farm operators as well as wind-turbines maintenance companies are among the main customers of WPred, which proposes a calendar application to help these companies to find the best time windows of opportunity for their maintenance operations. This application is essentially a communication tool where companies can manually fill maintenance tasks planning and work assignments with the help of a weather and production predictions feed. Although from our side we seem to distinguish the potential of OR contribution to the maintenance scheduling in the wind industry, it has become clearly evident since then. The collaboration with WPred has allowed this work to be supported by the Natural Sciences and Engineering Research Council of Canada through its grant program. This thesis has therefore been conducted between: i) the *Institut de Mathématiques Appliquées* part of the *Université Catholique de l'Ouest* in Angers (France) under the supervision of Professors Eric Pinson and Jorge E. Mendoza and ii) *Polytechnique Montréal* and the *Interuniversity research center on enterprise networks, logistics, and transportation* in Montreal (Canada) under the supervision of Professors Michel Gendreau and Louis-Martin Rousseau [3].

The contributions of this work are multifold and are reported subsequently while we define the outline of this thesis.

The first part of this thesis is devoted to the presentation of the general orientations and provides methodological details.

By now, the literature contains a sound body of work focused on improving decision making in generation unit and transmission line maintenance scheduling. Chapter 1 updates the state-of-the-art on maintenance scheduling in the electricity industry and provides a global overview of the current stream of research in this field. We study both regulated and deregulated power systems and highlight the changes that have followed the deregulation of the sector. We explore some important features such as network considerations, fuel management, and data uncertainty. We also introduce a multidimensional classification of references (problem solved, power system targeted, solution method used, among other things) to make easier to researchers working on this topic identifying the problem they are working on and what have been the main methods used for solving it. We also point out one of the motivations of our focus on the wind industry by identifying the lack in the OR literature.

To help with reading the remainder of the work, Chapter 2 provides some technical background on the solutions methods put into practice in this dissertation. More specifically, we first emphasize the difference between *constraint programming* (CP) and *linear programming* (LP). Then, we set out the *large neighborhood search* (LNS) metaheuristic framework. Afterwards, we present the wide-spread exact approach called Benders decomposition, and we point out the potential improvements to the original method that have been proposed. We also talk about its implementation. We then briefly speak about the Dantzig-Wolfe decomposition. Finally, to address the inherent uncertainty in many real-life problems, we introduce robust optimization.

The second part of this thesis is devoted to its most notable achievements as it presents the research conducted on the main optimization problem we have worked on.

Chapter 3 identifies and defines a challenging maintenance scheduling problem rising in the onshore wind industry. While the research in the field primarily focuses on condition-based maintenance strategies, we aim to address the problem on a short-term horizon considering a multi-skill workforce. The objective is to find a preventive maintenance plan that maximizes the revenue generated by the electricity production of the wind turbines while taking into account wind predictions, multiple task execution modes, and daily restrictions on the routes of the technicians. We prove that the problem is strongly NP-hard. In order to test optimization algorithms, we put together the insight on wind prediction and maintenance operations that we obtained from WPred and their cus-

2. Official website: `http://www.wpred.com`
3. one year in France followed by one year and a half in Canada and finally the last six months in France

tomers to build an instance generator. Although devising a perfect instance generator is nothing but impossible, we believe our instances represent reality with a good degree of accuracy.

At a first step, we tackle the deterministic problem introduced in Chapter 3 as we assume perfect knowledge of the wind speed during the planning horizon.

Chapter 4 proposes different *integer linear programming* (ILP) formulations of this problem. While providing an alternative way to define this latter, these mathematical models allow us to judge at a first sight the difficulty of the problem and the relevance of designing more complex strategies. Computational results indicate that, generally, the models cannot be directly used to solve realistic instances. Indeed, an ILP commercial solver is unable to solve to optimality most of the medium and large-sized instances after 3 hours, and the gap with respect to best known upper bounds is still very large. Nevertheless, we emphasize considerable differences in the effectiveness of the formulations.

After assessing the complex nature of the problem, we then explore two different approaches to tackle it.

Chapter 5 provides a heuristic solution method. More specifically, we propose a *constraint programming-based large neighborhood search* (CPLNS) approach. Starting from an initial maintenance plan, we iteratively alternate between a destroy stage (basically consisting in removing tasks from the maintenance plan) and a repair stage (basically consisting in rescheduling the tasks that have been removed) in order to improve the baseline maintenance plan. We develop multiple destruction operators either specifically conceived for the problem or adapted from the literature. The repair operator consists in solving a CP model with some fixed variables using branching strategies specially tailored for the problem. The CPLNS produces high quality solutions when the availability of the technicians is not binding. For instances where the availability of the technicians is scarce, the gap with respect to optimal solutions if known, or to the best upper bounds, grows as the number of tasks and the time horizon increase. Nonetheless, the computational results demonstrate the overall efficiency of the proposed metaheuristic. At this end of this chapter, we outline our specific work with WPred, as the CPLNS is the solution method implemented for the industrial prototype we provide them. We present the specificities taken into account in this particular case and the integration of the optimization tool in their existing product.

Chapter 6 investigates the possibility of designing an efficient exact approach that addresses the problem. We decompose our wind turbine maintenance scheduling problem into a task scheduling problem and a technician-to-task assignment sub-problem, and we solve it using a *branch-and-check* (B&C) approach. More specifically, while solving the task scheduling problem, we discard by means of cuts, all along the branch-and-bound tree, maintenance plans that cannot be performed by the technicians. In addition to the generic Benders cuts, we introduce problem-specific cuts based on several approximations to the sub-problem, and we demonstrate that they are key to speed-up the convergence of the approach. For most of the instances, the method finds an optimal solution in a short execution time. For the remaining instances where the 3-hour time limit is reached, it delivers solutions with a small gap with respect to upper bounds. The results suggest that this B&C approach significantly outperforms the direct resolution of ILP models and, in a certain context, the previously introduced metaheuristic (i.e. the CPLNS).

Chapter 7 tackles the problem with an alternative objective consisting in maximizing the availability of the turbines. We show in our experiments that this widespread objective when scheduling the maintenance of wind farms does not seem to be the most appropriate choice to address the cost-efficiency and/or profitability issues faced by wind farm operators and maintenance companies.

To meet another intended objective, Chapter 8 addresses the inherent uncertainty of the wind speed in the decision-making process for the problem introduced in Chapter 3. Indeed, operational decisions have to handle with care the quite unpredictable nature of the wind: the maintenance plan can become infeasible due to safety concerns or/and the revenue can vary widely if the wind speed deviates from its nominal value. After analyzing the possible approaches and, in particular stochastic optimization, we choose to rely on robust optimization, as it appears to be the most suitable approach in the context. We introduce a budgeted uncertainty set with additional constraints to deal with the possible spatial and time-wise correlation of the wind speed. We solve the problem using a cutting plane method built on top of the decomposition approach originally designed for the deterministic version of the problem and presented in Chapter 6. Our computational experiments demonstrate that the robust problem is more difficult to solve than its deterministic version, but we are able to compute near-optimal solutions in short solution times. We show that using robust solutions avoid feasibility issues while barely penalizing the revenue. To allow decision-makers to take the right decisions according to which kind of maintenance plan they are interested in, we study the impact of taking into account correlations or not and the impact of the budget on the quality of the robust solution. We also present an alternative robust approach in which we aim to ensure that the solution performs decently in the worst-case but very good in the nominal case. As a perspective to reduce the solution time for the robust problem, we briefly investigate the combination of a column generation process with the decomposition approach.

Finally we summarize the overall research, present the final conclusion and outline some research perspectives.

List of publications

The research reported in this thesis has been presented in many conferences and has been wrapped up on several articles either published or submitted to scientific journals on Operations Research.

Refereed publications in scientific journals

1. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016) Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695-706.

Papers submitted to scientific journals

1. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016). Solving a wind turbine maintenance scheduling problem. Submitted for possible publication to *Journal of Scheduling* (under second round of review).
2. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016). A branch-and-check approach to solve a wind turbine maintenance scheduling problem. Submitted for possible publication to *Computers & Operations Research* (under first round of review).

Conference talks

1. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016, november). *A branch-and-check approach to solve a wind turbine maintenance scheduling problem*. Institute for Operations Research and the Management Sciences - INFORMS annual meeting, Nashville (USA).
2. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016, july). *Une approche de type branch-and-check pour optimiser la planification de la maintenance dans l'industrie éolienne*. Journée du groupe de recherche en Ordonnancement Théorique et Appliqué (GOTha), Angers (France).
3. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016, may). *A branch-and-check approach to solve a wind turbine maintenance scheduling problem*. ISCO - 4th International Symposium on Combinatorial Optimization, Vietri-Sul-Mare (Italy).
4. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016, february). *Optimisation de la planification de la maintenance de parcs éoliens*. ROADEF - 17ème congrès annuel de la Société française de recherche opérationnelle et d'aide à la décision, Compiègne (France).
5. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2015, november). *Maintenance scheduling in wind farms*. GDR-RO - Journée conjointe des groupes de recherche P2LS, GT2L et GOTha sur le thème des problèmes d'optimisation intégrés, Paris (France).
6. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2015, june). *A MIP Model and Several Approaches to Schedule Maintenance in Wind Farms on a Short-term Horizon*. CORS/INFORMS2015 – Joint international meeting, Montreal (Canada).
7. Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2015, february). *Planification de la maintenance d'équipements de production d'électricité*. ROADEF - 16ème congrès annuel de la Société française de recherche opérationnelle et d'aide à la décision, Marseille (France).

I

Context, literature, and technical background

Chapter 1

Maintenance scheduling in the electricity industry: a literature review

The research reported in this chapter has been wrapped up on a journal article published in European Journal of Operational Research.

Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016) Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695-706.

The reliability of power plants and transmission lines in the electricity industry is crucial for meeting demand. Consequently, timely maintenance plays a major role reducing breakdowns and avoiding expensive production shutdowns. By now, the literature contains a sound body of work focused on improving decision making in generation unit and transmission line maintenance scheduling. The purpose of this chapter is to review that literature. We update previous surveys and provide a global overview of the problem. We study both regulated and deregulated power systems, and we explore some important features such as network considerations, fuel management, and data uncertainty.

Introduction

As the world population is expanding and a significant number of countries are experiencing (or are predicted to experience) a strong economic growth, the overall needs of energy services for the world is predicted to be multiplied by 3.2 between 2000 and 2100. If we exclusively focus on the electricity industry, the demand is predicted to increase at a rate of 2% per year until 2035 (European Commission, 2011). The rise of consumerism specially oriented on high-tech electronic devices is also an important factor. In this growth context, electric companies tend then to pledge for more and/or reliable electricity to maintain their position on the market or to acquire a more comfortable market share. Predominantly, they consistently aim to be able to meet the demand they committed to produce with a high degree of reliability while being cost-effective for suppliers. In this context, equipment maintenance management is a major economic issue as it saves some investment costs with the life extension of the generating units and prevents unnecessary downtime. Just to cite few examples, equipment maintenance management in electric power systems is concerned with decisions such as: when to stop a generating unit for maintenance, when to re-start it again, and how much resources (e.g., technicians) are to be assigned to the maintenance of a given unit during a given period. These decisions are taken under complex environments and constraints such as resource availability, demand satisfaction, and reliability thresholds.

One of the most successful contributions of OR to improve decision making in equipment maintenance management is the application of optimization techniques to solve maintenance planning and scheduling problems. In the particular case of electric power systems, these problems range from simple technician-to-equipment assignments to complex problems considering interactions between different stakeholders and uncertainty in the problem parameters. We build here on the work of Yamayee (1982); Kralj and Petrović (1988); Dahal (2004); Khalid and Ioannis (2012) to update the state-of-the-art, and we provide a global overview of the current stream of research in the field.

The chapter is organized as follows. Section 1.1 presents a brief description of the energy industry, Section 1.2 reviews maintenance scheduling problems rising in *regulated* and *deregulated* environments. Section 1.3 discusses existing solution methods for these problems. Section 1.4 concludes the chapter and outlines research perspectives in this sector.

1.1 The energy industry

The energy industry carries out three activities: production, transmission, and distribution. Traditionally the industry is organized in a centralized vertically integrated way (see Figure 1.1): a single company has a monopoly of the entire system in its area of operation. However, the government regulates the situation directly or indirectly: no entity must not take advantage of the end consumer. Therefore, the term *regulated monopoly utilities* is also used. With the deregulation of the electricity industry from the end of the 1990s, competition has been replacing monopolies in most places.

The deregulation (or liberalization) of the power industry has opened up the electricity market to competition. Several companies can now produce or distribute energy; it is, however, more difficult to introduce competition for the transmission management. Energy prices are no longer regulated by the government (hence the terms deregulation and liberalization) but are subject to market interactions. Regulations remain (sometimes the term *restricted power system* is used), but monopolies are no longer acceptable. Given the success of this system in the aeronautics, gas, and telephone industries, this reform is promoted as a benefit for the sector. It is intended to favor innovation, to lower prices, and to lead to better service. This new system introduces challenges such as the organization of the electricity market, the price-setting mechanism, and the coordination of the various actors.

Indeed, the introduction of market players leads to the emergence of new actors or redefines the role or activities of existing actors. An *independent system operator* (ISO) is responsible for the reliability and security of the system. It dispatches all or part of the energy transactions and can decrease loads on the network to avoid congestion. The ISO is the leading entity in a power market, and it must be fair. It manages the interactions between three key entities: *generation companies* (GENCOs), *transportation companies* (TRANSCOs), and *distribution companies* (DISCOs). When a single TRANSCO owns the entire transmission network, the ISO operates the transmission lines. The TRANSCO is then paid for the use of its lines and the maintenance of its network (Shahidehpour et al., 2002). *Retail energy service companies* (RETAILCOs) act as intermediaries between GENCOs and consumers by buying energy from the former to sell to the latter. Other actors exist, but their roles are relatively minor.

Energy transactions of different natures can take place in this new market structure. In a *power exchange model*, GENCOs and RETAILCOs negotiate bilateral contracts defining prices and quantities independently of the ISO. However, the availability of the transmission lines must be checked with the system operator to maintain security. This decentralized approach is opposed to the centralized approach (hereafter referred to as the *pool-based model*) where market participants share extensive information (e.g., energy offer, start-up costs, generation costs, ramp-rate for each generator) with the ISO, which is responsible for ensuring the social and economic welfare of the market while keeping the system safe. The ISO received two kinds of bids: producers' bids consist of energy blocks and their selling prices, and buyers' bids consist of energy blocks and their buying prices. The power price is determined by the balance between supply and demand using a market clearing process. Several markets, such as day-ahead, intra-day, real-time or a combination, can be encountered. Although they are different, both pool-based and power exchange models can coexist. Moreover, a transmission market deals with the purchase and sale of transmission rights. For a more detailed explanation of all these specificities, the reader is referred to Shahidehpour et al. (2002). Figure 1.2 summarizes the various interactions between the actors. It is however difficult to define a typical organization because several structures are possible.

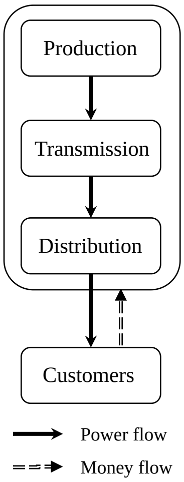


Figure 1.1 – Interactions in a vertically regulated utility

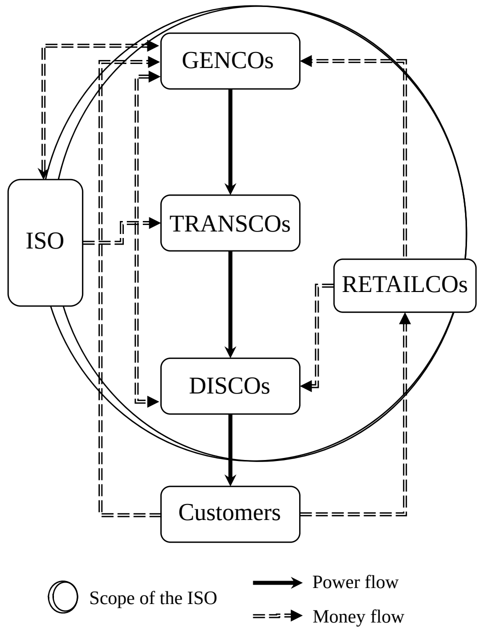


Figure 1.2 – Interactions between market players under deregulation

Liberalization modifies and sometimes complicates power industry issues. GENCOs, TRANSCOs, and DISCOs mainly serve their own interests, which may call into question the stability of energy production and/or energy distribution. Regulations are therefore required.

After this brief presentation of the electricity industry, we discuss, in the next section, optimization problems in maintenance scheduling of generating units and transmission lines rising in both regulated and deregulated power systems. We focus on network constraints, on data uncertainty, and on fuel consumption and supply management. To provide a global overview, we classify in Section 1.5 the references according to the problem they solve, the power system they target and the features they deal with.

1.2 Maintenance in the electricity industry

Maintenance represents the actions required to ensure that a product provides reliable service. Maintenance can be split into two categories: corrective and preventive. Corrective maintenance is performed after a breakdown. Preventive maintenance, as a proactive approach, intends to reduce the probability of failure. It is performed at predetermined intervals (based on a calendar or a routine system) or according to some prescribed criteria usually coupled with real-time monitoring. In the last case, the maintenance is usually referred as *predictive maintenance* or *condition-based maintenance*. Basically, companies are interested in increasing the probability that a generating unit will function in the required manner over its design life cycle with a minimum of maintenance. The process of determining the most effective maintenance approach refers to the concept of RCM. In today's competitive world economy, RCM translates obviously into savings. Questions such as why failures happens, what should be done when it happens, and how to predict or prevent each failure should be answered by the framework to be implemented. Two approaches are noticeable and can be jointly considered. On the one hand, topics that will manage the risks of generating unit failure in the most effective way, such as failure prediction or maintenance policies, have been widely studied. The results of these studies help to identify the best time windows for maintenance while taking into account unexpected breakdowns, and allow to define maintenance service contracts (e.g., preventive maintenance has to be performed every six months on each turbine). However, as mentioned in the introduction, theses studies fall out of the scope of this thesis.

On the other hand, electrical companies are interested in precisely defining time intervals for the preventive maintenance of generating units under financial (cost minimization, profits maximization) and/or reliability (leveling, maximization of the net reserves) considerations. As a very common practice in the OR studies focused on the topic, the first approach is then often assumed to part of an upstream work as it provides valuable inputs to set the maintenance time window constraints for each generating unit. Thereafter, maintenance scheduling optimization refers to this last approach.

Maintenance in the electricity industry concerns generating units and transmission lines; the horizon can be long-term or short-term.

1.2.1 Maintenance scheduling of generating units

The maintenance scheduling of generating units has been widely studied. On its basic version, the maintenance scheduling problem consists in defining when to stop the generating units for preventive maintenance in order to maintain the system reliability and to reduce the general operational costs. We refer to it as the *generation maintenance scheduling* (GMS) problem. Additional constraints include, but are not limited to:

— maintenance tasks: maintenance window (possible time for maintenance), sequence, incompatibility, spacing, and overlapping of tasks.

— generating units: highest/lowest production levels, ramp-rate [1].

— manpower: availability for each period, requirements by maintenance tasks.

— resources: availability for each period, requirements, consumption by maintenance tasks.

— network: transmission-line capacity (see Section 1.2.2), voltage.

— demand: fully satisfied or not, meeting of demand, energy-not-served (ENS) threshold.

— reliability: minimum reserve required by period, risk levels, ENS.

This optimization problem is generally NP-hard and may be nonlinear and nonconvex. Notice that the problem is generally solved at a strategic level as it mainly considers a long-term horizon.

Moreover, the power production is strongly impacted by maintenance decisions. To include load constraints, especially demand satisfaction, in the GMS problem, it can be necessary to simultaneously decide the production levels of the generating units and the maintenance scheduling. The solutions obtained can then be used as guidelines for *unit commitment (UC)* with a short time horizon. UC aims to schedule generating units level to meet forecasted load and reserve requirements.

In the next two sections we discuss the GMS problem in regulated and deregulated power systems.

Regulated power systems

Although the deregulation of the electricity industry has expanded very rapidly in the world, monopolies still operate in some regions. In a vertically integrated utility, the maintenance is scheduled in a centralized way, and all the information is available (costs, network, etc.). The various studies can be classified according to the nature of the considered objective function:

— **Reliability-based**

References: (Baskar et al., 2003; Canto and Rubio-Romero, 2013; Chen and Toyoda, 1991; Dahal and McDonald, 1997; Dahal et al., 1999; Dahal and Chakpitak, 2007; Ekpenyong et al., 2012; El-Amin et al., 2000; Fetanat and Shafipour, 2011; Foong et al., 2007; Mohanta et al., 2004, 2007; Reihani et al., 2012;

1. Output gap limitation between two successive time periods for a generating unit

Schlünz and van Vuuren, 2013; Suresh and Kumarappan, 2013; Volkanovski and Mavko, 2008; Wang and Handschin, 2000; Yare and Venayagamoorthy, 2010).

For reliability considerations, the main optimization criterion is the leveling of the net reserves along the planning horizon. For a given time period t, the net reserves correspond to the maximal power that can be produced by the available generating units not in maintenance during t minus the estimated demand during t. The most common approach is to minimize the sum of squares of the net reserves by period (Dahal and McDonald, 1997; Dahal et al., 1999; Dahal and Chakpitak, 2007; Ekpenyong et al., 2012; Foong et al., 2007; Mohanta et al., 2004, 2007; Reihani et al., 2012; Schlünz and van Vuuren, 2013; Yare and Venayagamoorthy, 2010). Nonetheless, alternative criteria have been studied. Chen and Toyoda (1991) maximized the reserve margin when they tackled isolated power systems, whereas they leveled the *reserve margin* for each area when they consider a multi-area system. Suresh and Kumarappan (2013) and Wang and Handschin (2000) leveled the reserve margin by minimizing an objective function based on the deviation between a reserve rate and its average (i.e. the average reserve rate over the planning horizon). Suresh and Kumarappan (2013) defined the reserve rate as the ratio of the net reserve to the sum of the generation capacity plus the predicted maximum load while Wang and Handschin (2000) defined the reserve rate as as the ratio of the difference between the sum of the generation capacity and the maximum predicted load to the maximum predicted load. El-Amin et al. (2000) considered the deviation between the reserve by period and the average reserve along the horizon. Baskar et al. (2003) considered the square of this deviation, and they studied the impact of a crew constraint (manpower availability at each period) on the results. Canto and Rubio-Romero (2013) maximized the sum by period of the ratio of the net power reserves to the gross power reserves. They introduced geographical, seasonal, and coordination constraints for a problem with wind farm turbines and thermal and hydroelectric power plants. Volkanovski and Mavko (2008) minimized the annual value of the loss of load expectation (LOLE) [2] taking into account the forecasted outage rate of the generating units. Finally, Fetanat and Shafipour (2011) defined an objective function such that the generating units have to be maintained as promptly as possible to reduce the expenses related to damaged machines.

To our knowledge, only Ekpenyong et al. (2012) and Suresh and Kumarappan (2013) gave outlines to schedule the power production. Ekpenyong et al. (2012) also considered ramp-rate constraints for the generating units. Otherwise, the remaining studies ensure for each period that the generating capacities that are not in maintenance are sufficient to cover the demand plus sometimes a reserve constraint.

— **Cost-based**

References: (Abirami et al., 2014; Al-Khamis et al., 1992; Anghinolfi et al., 2012; Baskar et al., 2003; Brandt et al., 2013; Buljubasic and Gavranovic, 2012; Burke and Smith, 2000; Canto, 2008; Charest and Ferland, 1993; Chattopadhyay, 1998; Digalakis and Margaritis, 2002; Ekpenyong et al., 2012; El-Amin et al., 2000; El-Sharkh et al., 2003; El-Sharkh, 2014; Fattahi et al., 2014; Fourcade et al., 1997; Frost and Dechter, 1998; Fu et al., 2007; Gardi and Nouioua, 2011; Godskesen et al., 2013; Gorge et al., 2012; Jost and Savourey, 2013; Khemmoudj et al., 2006; Kralj and Petrovic, 1995; Leou, 2006; Lusby et al., 2013; Marwali and Shahidehpour, 1998, 1999a, 2000a; Mollahassani-pour et al., 2014; Mytakidis and Vlachos, 2008; Rozenknop et al., 2013; Saraiva et al., 2011; Satoh and Nara, 1991; Silva and Morozowski, 1995; Silva, 2000; Yellen and Al-Khamis, 1992).

The other common objective is to minimize the general operational costs. These are production costs (e.g., fuel consumption), maintenance costs (e.g., loss of profit), and sometimes unit start-up costs (Canto, 2008). The production costs depend on the power output of the generating units, so it is necessary to compute an approximate schedule for their production level. An economic dispatch problem is usually solved with an objective of satisfying the demand at a minimum cost. The units with the lowest marginal costs are used to meet the system requirements; the other units produce only during the peak time periods. Some of these studies

2. expected time during which the demand is greater than the available capacity of the power system

took also into account the transmission network when scheduling the maintenance (see Section 1.2.2). Other authors included a fined-grained fuel management (see Section 1.2.4).

— **Reliability-and-cost-based**

References: (Huang, 1997; Kralj and Petrovic, 1995; Muñoz-Moro and Ramos, 1999).

The literature reports on some studies dealing with objective functions that are based on both reliability and cost. The goal is to find the best trade-off between these two conflicting objectives. Muñoz-Moro and Ramos (1999) proposed a lexicographic ordering of the two objectives considering first the operational costs and then the reserve margins. A parameter controls the increase of the cost when optimizing the reliability of the system.

Deregulated power systems

Deregulation changes the maintenance scheduling problem. The GENCOs and TRANSCOs are now usually responsible for maintaining their equipment. The ISO ensures the smooth running of the system in terms of reliability and security. Risk is managed by guaranteeing sufficient reserves of energy for each period to meet uncertainties in, for example, the demand or the generator deterioration. The different actors may have conflicting interests: GENCOs and TRANSCOs want to maximize their profits, whereas the ISO is concerned with demand satisfaction and congestion avoidance. For example, GENCOs tend to perform maintenance when the energy price is at its lowest, which may make it difficult to meet the demand. Thus, in an iterative way (see Figure 1.3), the GENCOs and TRANSCOs submit their preferred maintenance schedules to the ISO, which verifies the acceptable behavior of the system on the basis of all the market-player information. If the ISO is not satisfied, it will request modifications (e.g., the rescheduling of one or more maintenance tasks). Notice that the coordination procedure may vary from one system to another.

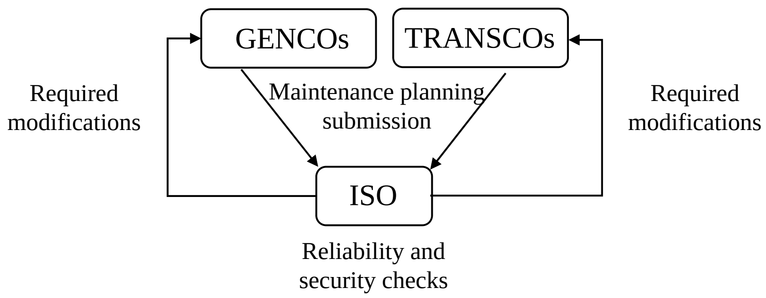


Figure 1.3 – Coordination procedure in a deregulated power system

Since the opening up to competition, the GMS problem in deregulated power systems has been widely studied (Badri and Niazi, 2012; Barot and Bhattacharya, 2008; Billinton and Abdulwhab, 2003; Bisanovic et al., 2011; Bozorgi et al., 2016; Chattopadhyay, 2004b,a; Conejo et al., 2005; Elyas et al., 2013; Eshraghnia et al., 2006; Feng et al., 2009; Fu et al., 2007; Geetha and Shanti Swarup, 2009; Han et al., 2011; Kim et al., 2005; Latify et al., 2013; Marwali and Shahidehpour, 1999a,b, 2000a,b; Min et al., 2013; Shabanzadeh and Fattahi, 2015; Wu et al., 2008; Zhan et al., 2014). The problem has globally the same constraints as for the vertically integrated case. Among some specificites, Bisanovic et al. (2011) introduced bilateral contracts (defined through prices and power quantities to supply) in their model. Chattopadhyay (2004b), Latify et al. (2013), Marwali and Shahidehpour (1999b), and Marwali and Shahidehpour (2000a) considered fuel based constraints. Indeed, restrictions on the maximum fuel supplied each week, month, and year may arise due to contractual agreements with fuel suppliers (see Section 1.2.4).

As noted for regulated power systems, it may be necessary to compute an approximate schedule for the production of the different generating units. Indeed, it allows for the more accurate estimation of the revenue and operating costs and can be required, among others, by the introduction of network constraints.

In some studies (Badri and Niazi, 2012; Fu et al., 2007; Marwali and Shahidehpour, 1999a,b, 2000a,b), cost minimization is still the objective. These studies generally deal with a security-constrained GMS problem operating under a pool-based model. Nonetheless, profit maximization is a much more common objective (Barot and Bhattacharya, 2008; Bisanovic et al., 2011; Bozorgi et al., 2016; Chattopadhyay, 2004b,a; Conejo et al., 2005; Elyas et al., 2013; Eshraghnia et al., 2006; Feng et al., 2009; Geetha and Shanti Swarup, 2009; Kim et al., 2005; Latify et al., 2013; Min et al., 2013; Shabanzadeh and Fattahi, 2015; Wu et al., 2008; Zhan et al., 2014). Note that these two objectives are different because profits depend on both costs and revenue. A very recent study by Dahal et al. (2015) explains how maintenance costs can be precisely modeled in deregulated markets considering, among others: failures, contractual compensations, rescheduling, and market opportunities. The authors demonstrated the importance and the impact of market related costs in maintenance schedules.

Under deregulation, the GENCOs have limited information about the system. The coordination of the decisions and information exchange between the GENCOs and the ISO are therefore very important. The interactions between these two latter actors have received intensive interest (Barot and Bhattacharya, 2008; Conejo et al., 2005; Elyas et al., 2013; Eshraghnia et al., 2006; Geetha and Shanti Swarup, 2009; Han et al., 2011; Min et al., 2013; Zhan et al., 2014). When the GMS problem is considered from the point of view of the ISO, one generally works with reliability-based objectives: maximizing the reserve throughout the horizon (Eshraghnia et al., 2006), maximizing the sum of the ratio of the net reserves to the gross reserves by period (Conejo et al., 2005), minimizing the standard deviation of this last ratio (Zhan et al., 2014), or minimizing a risk penalty factor related to an adequate level of reliability (Geetha and Shanti Swarup, 2009). More scarcely, one may consider cost minimization as in (Barot and Bhattacharya, 2008).

Many authors presented iterative coordination methods (between the GENCOs and ISO) based on rescheduling signals. Eshraghnia et al. (2006) suggested a coordination procedure where at every iteration the ISO indicates permissible and impermissible maximum power for the maintenance of the generating units in each period. In the same way, Barot and Bhattacharya (2008) coordinated the decisions through corrective signals sent by the ISO to the GENCOs, indicating the maximal capacities that can be in maintenance during critical time periods. These signals are calculated according to the responsibility of each GENCO for not supplying the load. These capacity-based signals can be replaced by penalties and/or incentive signals. Conejo et al. (2005), Geetha and Shanti Swarup (2009), and Min et al. (2013) penalized the scheduling of maintenance tasks during peak time periods or when the reliability of the system is uncertain. The objective function associated with the GENCO problem is modified at each iteration to represent the ISO's recommendations. In Conejo et al. (2005), GENCOs that adjust their maintenance plans are paid to offset their losses compared to their initial plans; the cost is paid by the customers. Latify et al. (2013) studied the maintenance of generating units that use natural gas as fuel. They introduced a complex coordination scheme between different entities (among them the ISO) as the GMS problem affects and is affected by gas price uncertainty, gas availability variations, and gas network constraints. Penalties based on reliability indices are sent from the ISO to the GENCOs in order to make the latter shift the maintenance of their units when the demand is low.

Among the alternative approaches, Han et al. (2011) proposed an ISO coordination procedure to adjust the individual generator-maintenance schedules according to the preferences of each GENCO, while guaranteeing system reliability. Elyas et al. (2013) gave the ISO the responsibility for maintenance scheduling. They took into account consumer satisfaction by maximizing the annual social welfare and also considered the profit-seeking GENCOs. They suggested a maintenance bidding approach to model the coordination mechanism. Finally, Zhan et al. (2014) analyzed the relationship between the ISO and the GENCOs using a multiobjective approach.

1.2.2 Transmission maintenance scheduling and network considerations

Along with generation maintenance, the maintenance of the transmission lines must be scheduled. It is necessary to ensure that taking a line out for maintenance does not impact the network reliability and security. This problem, usually called the *transmission maintenance scheduling* (TMS) problem, has received less attention in the literature than the GMS problem (Abirami et al., 2014; Fu et al., 2007; Geetha and Shanti Swarup, 2009; Langdon and Treleaven, 1997; Lv et al., 2012; Marwali and Shahidehpour, 1998, 1999b, 2000b). The TMS constraints are globally the same as those for GMS (e.g., time windows for maintenance tasks, resource requirements, demand satisfaction). These are also constraints on the line capacity and the problem may include voltage considerations. The network can be modeled as either a transportation model (Abirami et al., 2014; Marwali and Shahidehpour, 1998, 1999b) or a more complex but more realistic DC power flow model [3] (Fu et al., 2007; Geetha and Shanti Swarup, 2009; Langdon and Treleaven, 1997; Lv et al., 2012; Marwali and Shahidehpour, 2000b).

The TMS problem can be addressed independently from the GMS problem (Langdon and Treleaven, 1997; Lv et al., 2012; Marwali and Shahidehpour, 2000b). In this approach, the state of the network appears as a constraint during the resolution of the GMS problem. Although this approach is especially valid for regulated systems it may also apply to deregulated power systems. For instance, Marwali and Shahidehpour (2000b) looked for a trade-off between maintenance costs and loss of revenue over a short-term horizon.

When the TMS problem is tackled jointly with the GMS problem (Abirami et al., 2014; Fu et al., 2007; Geetha and Shanti Swarup, 2009; Marwali and Shahidehpour, 1998, 1999b), it becomes more complex. The maintenance must take into account economic considerations while minimizing the unsatisfied demand. The GMS and TMS problems have been solved generally on a monthly or weekly basis. Marwali and Shahidehpour (1998) coordinated maintenance decisions over a long-term horizon. Marwali and Shahidehpour (1999b) included fuel and emission constraints in the problem, considering local transmission lines within a GENCO. Fu et al. (2007) proposed an optimal coordination approach between generation and transmission outages, mid-term maintenance outage and hourly security constrained generation scheduling. The technique can be used by a company in a monopoly position or by an ISO. Abirami et al. (2014) solved the integrated maintenance scheduling problem on hourly basis and consider partial maintenance for transmission lines. When deregulated power systems are based on the power exchange model, the GENCOs and TRANSCOs are profit-oriented and do not have global information about the state of the system. As explained earlier, the ISO has to coordinate the submitted schedules; the cheapest transmission lines and generators might be overloaded. To our knowledge, only one study applies to this case; Geetha and Shanti Swarup (2009) solved the problem for every actor (ISO, GENCOs, and TRANSCOs) and coordinated the decisions through penalties.

Network considerations and especially coordination between GMS and TMS problems are important to maintain the security as well as the efficiency of the global system. If the TMS problem is not solved jointly with the GMS problem, network constraints can be introduced when the latter problem is solved (Badri and Niazi, 2012; Barot and Bhattacharya, 2008; Chattopadhyay, 1998; Chen and Toyoda, 1991; El-Sharkh et al., 2003; Leou, 2006; Marwali and Shahidehpour, 2000a, 1999a; Silva and Morozowski, 1995; Silva, 2000; Wu et al., 2008). These constraints can implicitly include the maintenance tasks planned for the network. Maintenance and unit commitment decisions must never exceed the line capacities. To our knowledge, Chen and Toyoda (1991) were the first to consider these constraints in a multi-area problem, but they did not handle unexpected breakdowns. The transportation model is widely used, except by Silva (2000) who modeled a DC power flow.

1.2.3 Management of uncertainty

Uncertainty in the GMS and TMS problems can be significant and it requires specific management. Indeed, the load curve as well as the fuel and energy prices may be difficult to estimate pre-

3. Linearization of an AC power flow model

cisely. Furthermore, corrective maintenance, following unexpected breakdowns, has a real impact on the production. Uncertainty has therefore to be handled with care.

Reserve constraints can help to deal with these risks. Reliability objectives, as discussed earlier, can also be used. However, using only deterministic strategies may be inappropriate in the case of large disturbances.

To explicitly consider unexpected breakdowns, researchers generally associate a *forced outage rate* (FOR) with generating units or transmission lines. The FOR represents the probability that the equipment will not be available for service when required. It impacts the quantity of energy that can be supplied. Thus, it prevents unsuitable maintenance schedules when load constraints are considered. One can also (artificially) reduce the capacity of the generators and solve a deterministic problem, but the not-supplied energy can be overestimated. However, the use of stochastic reliability indices, such as the expected energy not served (EENS) (Baskar et al., 2003; Chattopadhyay, 2004b; Geetha and Shanti Swarup, 2009; Lv et al., 2012; Marwali and Shahidehpour, 1999a,b; Silva and Morozowski, 1995; Yellen and Al-Khamis, 1992) and the *loss of load probability* (LOLP) (Billinton and Abdulwhab, 2003; Han et al., 2011; Mohanta et al., 2004, 2007; Reihani et al., 2012; Suresh and Kumarappan, 2013; Volkanovski and Mavko, 2008), is better suited to tackle problems in this context. The EENS is minimized or a threshold for acceptability is defined. Satisfying EENS within a specific threshold results in an acceptable LOLP. Mohanta et al. (2007), Reihani et al. (2012), and Suresh and Kumarappan (2013) considered the LOLP reliability index in a stochastic levelized risk method. The first step of this technique is to build an outage capacity probability table (COPT) by associating a probability to every generation capacity level taking into account the forced outage rate of each generating unit. The table can be computed with a convolution algorithm. The second step consists in finding the system's risk characteristic coefficient, defined as the change of the generating unit's outage capacity in MW when the system's risk changes by a certain factor. Each unit capacity is then replaced by an effective load carrying capacity based on the system's risk characteristic coefficient and the FOR. It represents the actual capacity of the units which is used for meeting the load demand. Similarly, the load of each interval is replaced by a value, called the equivalent load, which takes into account the peak variation during this interval and the system's risk characteristic coefficient. Leveling the risk may be finally realized by minimizing the sum of the squares of the reserves in the planning period. The reserve in each interval is obtained by subtracting the effective load carrying capacity of the available generating units and the equivalent load. In Mohanta et al. (2007), the risk associated with the resulting plan is evaluated by giving a confidence interval for the LOLP. Suresh and Kumarappan (2013) presented a coordination scheme to minimize the LOLP and the deviations of annual supply reserve ratio by considering the system's risk characteristic coefficient as a control parameter of their method. We refer the reader to the previous given references for more details. Several other approaches have been proposed. As a fairly closed approach, Volkanovski and Mavko (2008) chose to minimize directly the annual value of the LOLE. Billinton and Abdulwhab (2003) discussed a health levelization technique over a short-term horizon. Incorporated in a probabilistic framework, the objective is to maximize the health/security of the system, defined as the probability that the available reserves are greater than the required reserves. Chattopadhyay (2004b) simulated random outages using the Monte Carlo technique and proposed a stochastic optimization framework based on a game theory model to tackle the GMS problem. Geetha and Shanti Swarup (2009), Lv et al. (2012), Marwali and Shahidehpour (1999a), Marwali and Shahidehpour (1999b), Marwali and Shahidehpour (2000a), and Silva and Morozowski (1995) proposed a probabilistic approach that takes FOR into account in a problem incorporating network constraints. Last but not least, Feng et al. (2009) analyzed the impact of unexpected unit failures on the GMS solution and especially on: maintenance time periods, producer benefits, maintenance costs, and the costs of repairing or replacing some generating units. A modified superposed power law process models the unit failure rate. Its parameters are determined via the Gauss–Newton algorithm.

Fuzzy logic theory is also used to handle some data uncertainties in (Huang, 1997; Dahal et al., 1999; El-Sharkh et al., 2003; Mohanta et al., 2004; Bozorgi et al., 2016). It allows the representation and the use of linguistic knowledge. This concept is mainly used through fuzzy fitness function in a *genetic algorithm* (GA). Huang (1997) used triangular and trapezoidal membership functions for the multiple objectives (reserve margin, production cost) and for the soft constraints (manpower, time windows, geographical constraint). They used a GA to tune the membership functions. Dahal et al. (1999) introduced a fuzzy evaluation function combining the reliability objective function, the manpower constraint and a penalty factor associated with the inflexible demand satisfaction constraint. Based on their experience, they used triangular and trapezoidal membership functions to define the fuzzy sets. El-Sharkh et al. (2003) simulated the demand and the cost uncertainties using triangular membership functions.Mohanta et al. (2004) studied the incorporation of an uncertainty on the forced outage rate of the generating units by evaluating the reliability of power plants using fuzzy theory. They used a fuzzy LOLP to assess the quality of the method. Bozorgi et al. (2016) used fuzzy number to address the uncertainty on the production costs, on the electricity price, and on the demand. The major point when using a fuzzy logic approach is the intensive need of expert knowledge and substantial amounts of data.

The power demand may also be uncertain. Its stochastic nature can be explicitly considered. A set of scenarios that model alternative demands is used in (Anghinolfi et al., 2012; Brandt et al., 2013; Buljubasic and Gavranovic, 2012; Canto, 2008; Canto and Rubio-Romero, 2013; Gardi and Nouioua, 2011; Godskesen et al., 2013; Gorge et al., 2012; Jost and Savourey, 2013; Lusby et al., 2013; Rozenknop et al., 2013). The maintenance decisions ensure that the demand is met in all the scenarios. Ekpenyong et al. (2012) presented an effective method, called model predictive control, that detects demand disturbances and makes appropriate corrections.

When the objective is profit-based, it may be necessary to take into account the volatility of market prices. Wu et al. (2008) used a stochastic model based on an hourly price-based unit commitment. The hourly electricity and fuel prices are modeled as a set of scenarios determined via a Monte-Carlo method. Shabanzadeh and Fattahi (2015) proposed a robust approach to solve a maintenance scheduling problem defined from the viewpoint of GENCOs. They assumed the prices take their values into known intervals. The goal is to take a risk-averse decision by maximizing the profits of the GENCO in the worst-case scenario for the prices. To avoid overconservatism, the deviations from the nominal prices are bounded by a parameter called the uncertainty budget. The author applied duality theory to reformulate the problem into a deterministic problem with some additional variables and constraints (see Section 2.5 for more details on robust optimization).

The uncertainty can also affect the available quantity of fuel. In a complex coordination scheme with several actors, Latify et al. (2013) considered multiple scenarios for the availability of gas (used as the primary fuel in this study) and solved the extensive form of the stochastic program.

1.2.4 Fuel management and maintenance scheduling

Thermal production represents around 80% of the total global electricity production [4]. Fuel is fundamental for the effective functioning of these plants and refueling can impact the GMS problem. Nonetheless, only few studies include refueling considerations.

In some cases, refueling can be done continuously without significantly affecting electricity production (Al-Khamis et al., 1992; Chattopadhyay, 1998, 2004b; Latify et al., 2013; Marwali and Shahidehpour, 1999b, 2000a; Muñoz-Moro and Ramos, 1999), but sometimes (e.g., for nuclear reactors) it can occur only when the generators are offline (Anghinolfi et al., 2012; Brandt et al., 2013; Buljubasic and Gavranovic, 2012; Fourcade et al., 1997; Gardi and Nouioua, 2011; Godskesen et al., 2013; Gorge et al., 2012; Khemmoudj et al., 2006; Jost and Savourey, 2013; Lusby et al., 2013; Rozenknop et al., 2013). The introduction of fuel management into the GMS problem increases its complexity but also makes it more realistic. Badri and Niazi (2012) limited the fuel consumption for every generating unit during each time period. If fuel shortages occur, electricity can be purchased externally. Wu et al. (2008) limited the fuel allocations by group of generating units. These groups depend on predetermined contracts with suppliers. Al-Khamis et al. (1992) and Latify et al. (2013) considered fuel availability

4. International Energy Agency - http://www.iea.org/statistics/monthlystatistics/monthlyelectricitystatistics/

constraints. Muñoz-Moro and Ramos (1999) were concerned with the maximum fuel storage capacities of thermal plants. Marwali and Shahidehpour (1999b, 2000a) solved a fuel dispatch problem with multiple suppliers. The fuel consumption is limited by week, month, and year and is linked with the output level of the generators. Marwali and Shahidehpour (2000a) discussed the coordination of long-term and short-term maintenance decisions. Chattopadhyay (1998) discussed coal supply management with different transport modes from the mine to the power stations. Chattopadhyay (2004b) introduced fuel contracts with suppliers (with fixed fuel prices and volumes) in a GMS model and describes a successive linearization scheme to approximate the fuel consumption as a linear function. Finally, Fourcade et al. (1997) and Khemmoudj et al. (2006) planned production shutdowns to carry out refueling and maintenance operations (the fuel quantity to supply is known in advance).

A challenge submitted jointly by EURO [5] and ROADEF [6] in collaboration with EDF [7] has renewed interest in this latter problem. It presents a large-scale energy management problem with many constraints (Porcheron et al., 2010). The time horizon is long and fine-grained (up to 277 weeks with 7 or 21 timesteps per week). Two types of production units are considered. Non–nuclear plants can refuel continuously whereas nuclear plants must be shut down when fuel is supplied. In contrast to previous studies (Fourcade et al., 1997; Khemmoudj et al., 2006), the amount of fuel that is supplied for every nuclear plant is left as a decision variable. Furthermore, the production levels for the plants have to be planned under demand uncertainty modeled by a set of scenarios (up to 500). The objective is to plan the production and refueling while minimizing the production costs of non–nuclear plants and the refueling costs of nuclear plants. The problem has been proved to be NP-hard (Godskesen et al., 2013). Almost twenty teams participated in this challenge, and some of them (Anghinolfi et al., 2012; Brandt et al., 2013; Buljubasic and Gavranovic, 2012; Gardi and Nouioua, 2011; Godskesen et al., 2013; Jost and Savourey, 2013; Lusby et al., 2013; Rozenknop et al., 2013) published their results. Gorge et al. (2012) also considered this problem, but they introduced with some simplifications. Notice that the problem is often decomposed into several components: planning of the refueling, computation of the refuel amounts, and planning of the production.

1.2.5 Benchmarks

Publicly accessible data to test optimization algorithms for maintenance scheduling in electricity systems are rather scarce. Probably the most classical instance is the IEEE [8] Reliability Test System (IEEE-RTS) published in 1979 (Reliability Test System Task Force of the Application of Probability Methods Subcommittee, 1979) and released in 1996 (Reliability Test System Task Force of the Application of Probability Methods Subcommittee, 1999). The IEEE-RTS includes data on the network, the generating units, the demand, and the costs. This benchmark has been used in several articles (Badri and Niazi, 2012; Billinton and Abdulwhab, 2003; Elyas et al., 2013; Eshraghnia et al., 2006; Fattahi et al., 2014; Feng et al., 2009; Geetha and Shanti Swarup, 2009; Marwali and Shahidehpour, 1998, 1999a,b, 2000a; Mollahassani-pour et al., 2014; Schlünz and van Vuuren, 2013; Suresh and Kumarappan, 2013; Zhan et al., 2014). Another commonly used set of instances contributed by the IEEE [9] represent portions of the North American electricity system; this set served as a benchmark in (Abirami et al., 2014; El-Sharkh et al., 2003; El-Sharkh, 2014; Fu et al., 2007; Marwali and Shahidehpour, 2000b). In addition, an instance with 21 generating units described by Yamayee et al. (1983) regularly serves as a test case (Baskar et al., 2003; Dahal et al., 1999; Dahal and Chakpitak, 2007; Ekpenyong et al., 2012; Schlünz and van Vuuren, 2013; Suresh and Kumarappan, 2013; Yare and Venayagamoorthy, 2010). Data associated with real cases are often used to validate proposed techniques. However, to our knowledge, the only publicly available data is that published of the EURO-ROADEF-EDF challenge (Porcheron et al., 2010).

5. Association of European Operational Research Societies
6. Société française de recherche opérationnelle et d'aide à la décision
7. Electricité de France
8. Institute of Electrical and Electronics Engineers
9. University of Washington Electrical Engineering, Power systems case archive, http://www.ee.washington.edu/research/pstca/

1.2.6 Electricity generating technologies

It is noteworthy that the scope of the OR studies on maintenance scheduling optimization cover essentially the traditional power plants (nuclear and fossil-fuels). This may be partially explained by the strong predominance of these latter in the world. Consequently, renewable energies are barely mentioned in the OR literature about maintenance scheduling optimization. In the rare studies they are taking into consideration, it is always with other electricity generating technologies and, to our knowledge, only in the case of a regulated power system where the problem can be solved considering a large part (or the totality) of the network (Canto and Rubio-Romero, 2013). This situation is quite surprising, but maintenance scheduling optimization is not overlooked for renewable electricity generating technologies. It is just drawing the attention of another field of research. For the sake of conciseness, we just focus here on the wind industry. Exploring some existing studies, maintenance optimization for wind turbines has recently started to received attention in literature related to electrical and reliability engineering (we refer the reader to Ding et al. (2013) for a survey). This stream of research primarily focused on RCM. Indeed, studies are essentially interested in the definition of maintenance policies according to failure models or/and condition monitoring. For instance, Carlos et al. (2013) proposed a stochastic model to find a maintenance strategy minimizing the operational costs and maximizing the annual electricity production while taking into account meteorological conditions, degradation, and failures. The underlying methodology uses the Nelder-Mead method. The results of the optimization process is a maintenance time interval for each turbine. The study considers each turbine in isolation which is very common as studies usually choose to focus at most on a single wind farm. However, these studies do not build detailed maintenance plans that can be used on a daily or weekly basis, since human and material resource management is generally overlooked. To our knowledge, only few studies have responded to this issue. Kovács et al. (2011) considered fine-grained resource management while scheduling on a one-day horizon maintenance operations for onshore wind turbines. These authors aimed to minimize lost production due to maintenance and failures. They solved an ILP formulation of the problem with a commercial solver. With regard to offshore wind farms, Irawan et al. (2016) optimized a maintenance routing and scheduling problem minimizing labor, travel and penalty costs. They proposed a solution method based on Dantzig-Wolfe decomposition in which all the feasible routes for each vessel are generated a priori.

1.3 Solution methods

Various heuristic and exact approaches have been proposed for the GMS and/or TMS problems. The solution techniques mainly focus on metaheuristics and mathematical programming. This section provides details about all these techniques and discusses their applicability to the problems defined in the previous sections. To provide a global overview, Table 1.1 classifies the references according to the solution method they apply.

1.3.1 Mathematical programming approaches

Mathematical programming methods are essentially based on dynamic programming, pure mixed-integer programming, branch-and-bound, Lagrangian relaxation, and Benders decomposition.

Until the 90s, dynamic programming was often used for solving the GMS problem because of its sequential decision process. For instance, Huang (1997) combined dynamic programming with fuzzy logic in a multiobjective problem. However, the "curse of dimensionality" limits the application of this method (Yamayee, 1982).

Alternatively, many mixed integer programming models are proposed for the GMS and TMS problems. Objectives and constraints widely vary from one study to another; a clear indicator of the difficulty to point out a general model. Only linear models are likely to be handled by commercial solvers and only small or medium-sized instances can be efficiently solved. Badri and Niazi (2012), Barot and Bhattacharya (2008), Bisanovic et al. (2011), Canto and Rubio-Romero (2013), Chen and Toyoda (1991), Conejo et al. (2005), Fourcade et al. (1997), Latify et al. (2013), Mollahassani-pour et al.

(2014), Muñoz-Moro and Ramos (1999), Shabanzadeh and Fattahi (2015), and Wu et al. (2008) formulated *mixed-integer linear programming* (MILP) models and solved them combining branch-and-bound algorithms with simplex or interior-point methods (see Section 2.1 for more details). Among studies based on multiobjective optimization, Kralj and Petrovic (1995) designed a customized branch-and-bound, and Muñoz-Moro and Ramos (1999) proposed a two-stage goal programming approach in which each problem is solved by branch-and-bound.

The direct use of mixed integer programming is sometimes unsuitable as the computational time grows prohibitively with problem size. To overcome this drawback, decomposition techniques can be applied. The most studied technique is Benders decomposition. For more details, the reader is invited to consult Section 2.3. This decomposition technique applies particularly well to the GMS problem because of the problem's intrinsic two-stage structure. The master problem is generally concerned only with the constraints regarding the scheduling of the maintenance tasks as well as the resources requirement if needed. Load and network constraints, as well as fuel management, are moved into the sub-problems. Benders decomposition is applied for both regulated (Al-Khamis et al., 1992; Canto, 2008; Fu et al., 2007; Marwali and Shahidehpour, 1998, 1999a; Silva and Morozowski, 1995; Silva, 2000; Yellen and Al-Khamis, 1992) and deregulated (Fu et al., 2007; Geetha and Shanti Swarup, 2009; Marwali and Shahidehpour, 1999b, 2000a) power systems. Lv et al. (2012) introduced modified Benders feasibility cuts. They also define an index of critical lines –related to the system reliability– to reduce the computational complexity and the solution time. Marwali and Shahidehpour (2000a) coordinated long-term and short-term generation decisions with a dynamic scheduling algorithm. They used a Benders decomposition to define the maintenance decisions and an augmented Lagrangian relaxation to solve the underlying unit commitment. The same authors proposed also a deterministic (Marwali and Shahidehpour, 1998) and a probabilistic (Marwali and Shahidehpour, 1999a) Benders based approach to jointly solve the GMS and TMS problems.

Mathematical programming may be coupled with heuristic approaches. For instance, LP has been used in combination with local search (Gardi and Nouioua, 2011), genetic algorithms (Feng et al., 2009), and customized heuristics (Jost and Savourey, 2013). Examples of the latter include the work of Rozenknop et al. (2013) who combined column generation with customized heuristics, and that of Gorge et al. (2012) who applied a technique based on semidefinite programming, followed by a randomized rounding procedure.

It is worth noting that mathematical programming does not seem to be better adapted to one or other of problems discussed in the previous sections. However, all the methods presented in this part are mainly suitable when only linear objectives and constraints are considered in the GMS and/or TMS problems. Therefore, they cannot be used without any adaptation with the main reliability criterion (i.e. the minimization of the square of the reserves) or with quadratic cost functions.

1.3.2 Heuristics and metaheuristics

Due to the NP-hardness of the maintenance problems faced in power systems and the size of real-word instances, heuristic techniques have been largely developed. These methods also allow for more flexibility to deal with non linear or very complex constraints and/or objectives.

Genetic algorithms

GAs are widely used for solving the GMS problem due, mainly, to the structure of the decisions that are made. These algorithms are inspired by natural evolution. A population of abstract representations of solutions, called individuals, evolves through an iterative process toward better solutions. Solutions are usually encoded to facilitate the application of the several techniques used in GAs such as selection, mutation and crossover. In the particular case of the GMS problem, different coding methods have been considered. In the binary representation, an individual has $G \times T$ genes (where G is the number of generating units and T is the number of time periods) and the value of gene (g, t) is set to 1 if and only if unit g starts maintenance at the beginning of time period t. In the integer representation, a individual has G genes and the value of gene g corresponds to the maintenance starting period of generating unit g. Binary or gray encoding of the latter values may also be used. Dahal and McDonald (1997) showed that integer-coding is the most efficient encoding strategy since it generates the smallest search space. Moreover, the integer coding reduces the probability of infeasibility during the process, and it avoids the overhead necessary to code and decode a solution. Baskar et al. (2003) revisited the binary for integer representation, the real encoding (using real values instead of integer values), and the classic integer encoding. For their experiments, integer coding proves to be the best independently of the problem size. Wang and Handschin (2000) and Reihani et al. (2012) reached the same conclusions about the efficiency of integer-coding. Despite these findings, binary coding is still used in some approaches (Eshraghnia et al., 2006; Leou, 2006; Mohanta et al., 2004, 2007; Suresh and Kumarappan, 2013). Finally, Volkanovski and Mavko (2008) introduced a completely different encoding strategy that uses real values between 0 and 1 obtained by dividing the maintenance starting time value by the number of time periods in the time horizon.

Needless to say, contrary to solution encoding, the fitness function associated with every individual is problem dependent. Usually, it includes penalties associated with the violations of some constraints. Fuzzy models may be developed to handle data uncertainties (see Section 1.2.3) through the use of fuzzy evaluation functions.

Being population-based metaheuristics, GAs require initial solutions to built the initial population. The literature reports on randomly generated solutions (Dahal et al., 1999; Huang, 1997; Leou, 2006; Mohanta et al., 2004, 2007; Reihani et al., 2012) and heuristically generated solutions (Burke and Smith, 2000; Dahal and Chakpitak, 2007; Volkanovski and Mavko, 2008). The latter are often built using constructive methods that are mainly based on the following process: ranking generating units in order of decreasing capacity, and iteratively schedule the maintenance of the units when the demand is at its lowest level, while satisfying the constraints of the model. GAs try to improve solutions by applying the classic operators such as tournament or roulette wheel selection, one or two-point crossovers, random mutation and a replacement policy based usually on elitism. To reduce the probability of trapping in local optima, Dahal and Chakpitak (2007); Mohanta et al. (2004, 2007) accepted non-improving solutions at each iteration according to the probabilistic acceptance criterion of *simulated annealing* (SA). Langdon and Treleaven (1997) designed a GA to schedule the maintenance of the transmission lines combined with some greedy heuristics. Some GAs include a local search algorithm for improving the quality of the solutions. Neighborhoods are usually defined by changing the maintenance starting time of a randomly selected generator. Burke and Smith (2000) tested the combination of a genetic algorithm with several local search methods: a basic hill climbing technique, a SA, and a *tabu search* (TS). Hybridization with TS proved to be the most efficient approach in their experiments. El-Sharkh et al. (2003) maintained feasibility using a hill climbing technique during the solution process. Leou (2006) executed SA for each individual solution of the population. Reihani et al. (2012) designed an hybrid algorithm based on extremal optimization and a GA.

GAs are mostly used in regulated power systems; they allow dealing with the non linearity of the main reliability-based objectives.

Particle swarm optimisation

Recent studies (Ekpenyong et al., 2012; Suresh and Kumarappan, 2013; Yare and Venayagamoorthy, 2010) applied *particle swarm optimization* (PSO) to the GMS problem. PSO is another population-based metaheuristic which bares many similarities with GAs. It simulates the social behaviour of birds within a flock, or even fishes within a school evolving by information exchange. The population is composed of particles moving in the search space of the optimization problem. The position of a particle represents a candidate solution. Each particle is guided according to the best solution (fitness) it has achieved so far, and according to the current best particle, or particles if multiple swarms are considered. On its original version, PSO handles only continuous variables; however, Ekpenyong et al. (2012) introduced a penalty function to deal with the discrete nature of the variables involved in the GMS problem. Suresh and Kumarappan (2013) chose to test a particular binary version of PSO. They defined the particle according to the capacity outage probability table. To improve the effectiveness of the algorithm, they applied crossover and mutation operators to some particles during the iterative process. Yare and Venayagamoorthy (2010) considered only the latter operator in a multiple swarms-modified discrete PSO where information sharing is enhanced by using multiple populations. Non linear reliability-based objectives (Suresh and Kumarappan, 2013; Yare and Venayagamoorthy, 2010) and cost-based objectives (Ekpenyong et al., 2012) of regulated power systems have been tackled using PSO.

Other populations-based methods

Apart from GAs and PSO, other population-based techniques have been explored in the literature. The first three studies deal with regulated power systems and are concerned with the maintenance and generation costs.

Digalakis and Margaritis (2002) applied a parallel co-operating cultural algorithm to solve the GMS problem. Contrary to GAs, this method considers multiple populations but uses similar operators (selection, crossover, mutation). The exchange of individuals between the populations allows to guide the algorithm towards the promising areas of the search space. The authors used local search to improve the quality of the solutions after the initialization and the application of the genetic operators.

El-Sharkh (2014) tested a clonal selection algorithm (CSA) to schedule the maintenance in a regulated power system. CSA imitates the mechanisms of the adaptive immune system. From a population of individuals randomly generated, all infeasible individuals are repaired using a hill climbing algorithm. The best individuals are then cloned and a mutation operator is applied to these copies. The hill climbing technique is used again to repair them. According to the outages planning, an economic load dispatch problem is solved within the process. This algorithm is also used for deregulated power systems in (Elyas et al., 2013).

Recently, Abirami et al. (2014) presented a teaching-learning based optimization algorithm (TLBO) for solving the GMS problem. This technique is inspired by the transfer of knowledge between teacher and students in the classroom. TLBO is a population-based iterative learning algorithm. However, it does not use genetic operations like selection, crossover, and mutation, but tries to improve individuals based on their interaction with the teacher and the communication with the other individuals. The population is initialized by randomly setting the starting period for the maintenance of the generators and the transmission lines. The best individual is deemed as the teacher. Other individuals are modified to move towards the teacher.

Finally, Zhan et al. (2014) proposed an approach based on a novel evolutionary algorithm called group search optimizer inspired by animal searching behaviour. They computed a set of Pareto-optimal solutions for a multiobjective problem based on reliability, costs, and profits.

Simulated annealing

SA is a stochastic metaheuristic inspired by the annealing process in metallurgy. At each iteration, SA randomly generates a neighbor of the current solution. The algorithm accepts non-improving solutions according to some probability, allowing the search to escape local optima. These probability decrease as the number of iterations increases. SA has been used as the main approach to solve the GMS problem in (Burke and Smith, 2000; Dahal and Chakpitak, 2007; Fattahi et al., 2014; Han et al., 2011; Saraiva et al., 2011). Fattahi et al. (2014) used in addition customized heuristics to solve the underlying UC. Saraiva et al. (2011) built new solutions by randomly selecting a generating unit and setting a new random maintenance starting time for it. At each iteration, Han et al. (2011) selected first the unit with a weighted roulette wheel where weights are fixed according to the LOLP associated with each generator, and then change randomly its maintenance starting time. SA has been also combined with other techniques. For instance, Anghinolfi et al. (2012) hybridized SA and LP; Godskesen et al. (2013) coupled a SA and CP; and Burke and Smith (2000), Dahal and Chakpitak (2007), Mohanta et al. (2004), Mohanta et al. (2007), and Leou (2006) embedded a SA into a GA. SA has been primarily used to solve problems arising in the regulated power systems.

Tabu search

TS is a local search-based metaheuristic that avoids revisiting solutions by recording the recent history of the search in a short-time memory called tabu list. To escape local optimum, non improving solutions are accepted during the algorithm. To our knowledge, TS has only been applied to solve the GMS problem in regulated power systems. Burke and Smith (2000) found better computational results when TS is used as a local search embedded in a GA. The neighborhoods are those described in the Section 1.3.2. El-Amin et al. (2000) applied TS by randomly selecting a generating unit and modifying its maintenance starting time.

Ant colony optimization

Ant colony optimization (ACO), a constructive metaheuristic inspired by the behavior of ant colonies, has been also applied to solve the GMS problem in regulated power systems with a cost-based objective (Foong et al., 2007; Fattahi et al., 2014; Mytakidis and Vlachos, 2008). The algorithm builds solutions incrementally by selecting a maintenance starting period for each generating unit. The selection process is based on the combination of pheromone level, related to the number of times the component has been selected, and some greedy heuristics. Fattahi et al. (2014) combined ACO with some sub-algorithms to deal with a complex problem based on operational hours. According to the outages fixed in the previous phase, UC is solved by a customized heuristic which proves to be better in their experiments than a GA.

1.3.3 Constraint programming

CP is particularly useful for highly constrained problems. It is a powerful and flexible tool which makes expressing complex constraints relatively easy. It applies particularly well to the ROADEF challenge where maintenance tasks faced a variety of constraints such as overlapping, spacing, and incompatibility (Buljubasic and Gavranovic, 2012; Godskesen et al., 2013; Brandt et al., 2013). Brandt et al. (2013) coupled CP to a greedy heuristic to solve the unit commitment and the refueling problems. Buljubasic and Gavranovic (2012) presented a heuristic approach combining a constraint satisfaction problem to both a local search based on the marginal cost and a constructive optimization algorithm. Godskesen et al. (2013) used it with a local search and a greedy heuristic in a 3-phase algorithm. For a quite similar problem, Khemmoudj et al. (2006) proposed an approach combining CP and local search.

CP is less suitable when the main objective of the problem is to find near-optimal solutions. Therefore, it has been scarcely used for solving the GMS problem. Frost and Dechter (1998) iteratively applied CP to solve a cost bound problem; learning constraints are added to improve the efficiency of the algorithm.

1.3.4 Game theory

Game theory-based approaches have been explored on the problem arising in deregulated power systems (Bozorgi et al., 2016; Chattopadhyay, 2004a; Kim et al., 2005; Min et al., 2013). These approaches are especially suitable since every GENCO tries to predict its competitors actions so as to stay one step ahead. The strategies adopted by the GENCOs are defined by a Nash equilibrium of the game. Since they took into account data uncertainty, Bozorgi et al. (2016) solved a dynamic non-cooperative fuzzy game.

1.4 Conclusion and perspectives

The GMS and TMS problems are the two main maintenance scheduling problems in the electricity industry. The constraints are related to the maintenance tasks (time windows, incompatibility, sequence), the resource requirements, the reliability, and the demand satisfaction. Sometimes, e.g., for nuclear power plants, fuel consumption management is required. The GMS and TMS problems can be solved jointly or network constraints can be introduced into the former. Production planning is often incorporated into GMS, especially over a short-term horizon. This results in a complex problem that is generally NP-hard.

Maintenance scheduling is a major challenge in the electricity industry, especially since the liberalization of the electricity market. The objectives of regulated power systems are based mainly on the reliability (leveling, maximization of net reserves) and the costs (minimization of the operational costs). These objectives are not necessarily suitable for deregulated systems. It may be more appropriate to maximize the profits of the GENCOs and to coordinate the decisions of the various actors. The objectives of regulated systems remain relevant to the ISO –the actor that must ensure system reliability and security– but may conflict with the goals of the other actors (GENCOs, TRANSCOs, DISCOs). A multiobjective optimization is thus a future solution framework to propose solution taken into account the conflicting interests of the different actors.

Many solution methods have been proposed for the GMS and/or TMS problems. They include heuristics, metaheuristics (GA, PSO, SA, TS, ACO, hybrid approaches, mathematical programming (dynamic programming, MILP, branch-and-bound, Benders decomposition), CP, and game theory. As the problem complexity increases making frontal resolution impracticable, the use of decomposition techniques become more and more relevant.

To the best of our knowledge, some problems have not yet been investigated. These include load uncertainty and price volatility when the TMS problem is solved jointly with the GMS problem or where coordination is needed between the GENCOs and the ISO. Taking into account numerous unexpected breakdowns, short-term rescheduling could be investigated. Apart from that, the growing renewable energy industry and the stochastic nature of the associated power sources have an increasing impact on the planning problems in power systems. For instance, maintenance decisions in the solar and wind industries are quite specific, since the consequence of shutting down an equipment on the power production depends on an uncontrollable factor: the weather. Moreover, due to safety concerns, the weather has a direct impact on the possible concrete realization of the maintenance operations. Furthermore, since solar and wind power forecasting can only be established in a very short term horizon, explicitly handling these uncertainties, as well as those associated with the demand, may lead to substantial energy and cost savings. However, these new issues have not been investigated upfront by the OR community, despite research opportunities coming from recent environmental pressure measures. In conclusion, future research will have to improve the handling of uncertainty, the coordination of decisions as regards the different actors of the power systems as well as to answer the new challenge power systems face, especially with the growing renewable industry. Part II of this thesis responds to this need as we address a maintenance scheduling problem of onshore wind farms.

1.5 Classification of the bibliographical references

Table.1.1 classifies each reference according to:

— the problem it solves: GMS, TMS (a checkmark in both GMS and TMS columns means that the two problems are addressed in the chapter), GMS with network constraints (GMS+N)

— the power system it targets: regulated (Rg) or/and deregulated (Dg)

— the objective function it handles: reliability-based (R), cost-based (C), profits-based (P).

— some features it deals with: FOR, stochastic demand (D), refueling management (F).

— the solution method it proposes.

It is not however exhaustive to cover the wide range of features that can occur for the maintenance scheduling in electricity industry. Some classifications may also be debated since some studies do not fit well into the boxes, but we aim at being the most consistent.

Table 1.1 – Classification of the bibliographical references

Reference	Problem solved			Power system		Objectives			Features			Solution method
	GMS	TMS	GMS+N	Rg	Dg	R	C	P	FOR	D	F	
Abirami et al. (2014)	✓	✓		✓			✓					teaching learning based optimization algorithm
Al-Khamis et al. (1992)	✓			✓			✓				✓	Benders decomposition
Anghinolfi et al. (2012)	✓			✓			✓			✓	✓	matheuristic combining SA and LP
Badri and Niazi (2012)			✓		✓		✓				✓	MILP
Barot and Bhattacharya (2008)			✓		✓		✓	✓				MILP
Baskar et al. (2003)	✓			✓		✓	✓		✓			GA with modified operators
Billinton and Abdulwhab (2003)	✓				✓	✓			✓			stochastic heuristic
Bisanovic et al. (2011)	✓				✓			✓				MILP (interior-point method)
Bozorgi et al. (2016)	✓				✓			✓				dynamic non-cooperative fuzzy game
Brandt et al. (2013)	✓			✓			✓			✓	✓	CP combined with heuristic
Buljubasic and Gavranovic (2012)	✓			✓			✓			✓	✓	CP combined with local search and heuristic
Burke and Smith (2000)	✓			✓			✓					Hybrid GA/SA, GA/TS
Canto (2008)	✓			✓			✓			✓		Benders decomposition
Canto and Rubio-Romero (2013)	✓			✓		✓				✓		MILP
Charest and Ferland (1993)	✓			✓			✓					heuristic based on Lagrangian relaxation
Chattopadhyay (1998)			✓	✓			✓		✓		✓	LP combined with a rule-based heuristic
Chattopadhyay (2004a)	✓				✓			✓				MILNP based on game theory
Chattopadhyay (2004b)	✓				✓	✓		✓	✓		✓	stochastic framework
Chen and Toyoda (1991)			✓	✓		✓						Decomposition technique based on LP and heuristics
Conejo et al. (2005)	✓				✓	✓		✓				MILP
Dahal and McDonald (1997)	✓			✓		✓						GA
Dahal et al. (1999)	✓			✓		✓						GA combined with fuzzy logic
Dahal and Chakpitak (2007)	✓			✓		✓						Hybrid GA/SA
Digalakis and Margaritis (2002)	✓			✓			✓					multipopulation cultural algorithm
Ekpenyong et al. (2012)	✓			✓		✓	✓			✓		PSO
El-Amin et al. (2000)	✓			✓		✓	✓					TS
El-Sharkh et al. (2003)			✓	✓			✓			✓		GA combined with fuzzy logic
El-Sharkh (2014)	✓			✓			✓					clonal selection algorithm
Elyas et al. (2013)	✓				✓	✓		✓				clonal selection algorithm
Eshraghnia et al. (2006)	✓				✓	✓		✓				GA
Fattahi et al. (2014)	✓			✓			✓					ACO, SA, MILNP
Feng et al. (2009)	✓				✓	✓		✓	✓			decomposition technique based on GA and LP
Fetanat and Shafipour (2011)	✓			✓		✓						ACO
Foong et al. (2007)	✓			✓		✓						ACO

Table 1.1 – Classification of the bibliographical references

Reference	Problem solved			Power system		Objectives			Features			Solution method
	GMS	TMS	GMS+N	Rg	Dg	R	C	P	FOR	D	F	
Fourcade et al. (1997)	✓			✓			✓				✓	MILP
Frost and Dechter (1998)	✓			✓			✓					CP
Fu et al. (2007)	✓	✓			✓	✓	✓		✓			Benders decomposition and hybrid subgradient method with Dantzig-Wolfe decomposition
Gardi and Nouioua (2011)	✓			✓			✓			✓	✓	local search
Geetha and Shanti Swarup (2009)	✓	✓			✓	✓		✓	✓			Benders decomposition and Lagrangian relaxation
Godskesen et al. (2013)	✓			✓			✓			✓	✓	CP combined with SA
Gorge et al. (2012)	✓			✓			✓			✓	✓	semidefinite relaxation
Han et al. (2011)	✓				✓	✓			✓			SA
Huang (1997)	✓			✓		✓	✓					fuzzy dynamic programming combined with a GA to set the parameters
Jost and Savourey (2013)	✓			✓			✓			✓	✓	MILP combined with heuristics
Khemmoudj et al. (2006)	✓			✓			✓				✓	CP combined with local search
Kim et al. (2005)	✓				✓			✓				game theory
Kralj and Petrovic (1995)	✓			✓		✓	✓					extension of branch-and-bound technique (multiobjective)
Latify et al. (2013)	✓			✓		✓		✓			✓	stochastic LP (extensive form)
Langdon and Treleaven (1997)		✓		✓			✓					GA combined with a heuristic
Leou (2006)			✓	✓			✓					Hybrid GA/SA
Lusby et al. (2013)	✓			✓			✓			✓	✓	Benders decomposition combined with heuristics
Lv et al. (2012)		✓		✓			✓		✓			Benders decomposition
Marwali and Shahidehpour (1998)	✓	✓		✓			✓					Benders decomposition
Marwali and Shahidehpour (1999a)			✓	✓	✓		✓		✓			Benders decomposition
Marwali and Shahidehpour (1999b)	✓	✓			✓		✓		✓		✓	Benders decomposition
Marwali and Shahidehpour (2000a)			✓	✓	✓		✓		✓		✓	Benders decomposition
Marwali and Shahidehpour (2000b)		✓			✓		✓	✓				Benders decomposition
Min et al. (2013)	✓				✓	✓		✓				game theory
Mohanta et al. (2004)	✓			✓		✓			✓			Hybrid GA/SA combined with fuzzy logic
Mohanta et al. (2007)	✓			✓		✓			✓			Hybrid GA/SA
Mollahassani-pour et al. (2014)	✓			✓			✓					MILP
Muñoz-Moro and Ramos (1999)	✓			✓		✓	✓		✓		✓	goal programming based on MILP (multiobjective)
Mytakidis and Vlachos (2008)	✓			✓			✓					ACO
Reihani et al. (2012)	✓			✓		✓			✓			GA combined with LS (extremal optimization)
Rozenknop et al. (2013)	✓			✓			✓			✓	✓	column generation combined with heuristics
Saraiva et al. (2011)	✓			✓			✓					SA

Table 1.1 – Classification of the bibliographical references

Reference	Problem solved			Power system		Objectives			Features			Solution method
	GMS	TMS	GMS+N	Rg	Dg	R	C	P	FOR	D	F	
Satoh and Nara (1991)	✓			✓			✓					SA
Schlünz and van Vuuren (2013)	✓			✓		✓						SA and local search (hybridisation)
Shabanzadeh and Fattahi (2015)	✓				✓	✓		✓			✓	robust linear optimization (reformulation)
Silva (2000)			✓	✓			✓					Benders decomposition
Silva and Morozowski (1995)			✓	✓			✓		✓			Benders decomposition
Suresh and Kumarappan (2013)	✓			✓		✓			✓			hybrid improved binary PSO / GA
Volkanovski and Mavko (2008)	✓			✓		✓			✓			GA
Wang and Handschin (2000)	✓			✓		✓						GA
Wu et al. (2008)			✓		✓			✓				Lagrangian relaxation combined with Dantzig-Wolf technique
Yare and Venayagamoorthy (2010)	✓			✓		✓						multiple swarms-modified PSO
Yellen and Al-Khamis (1992)	✓			✓			✓		✓			Benders decomposition
Zhan et al. (2014)	✓				✓	✓		✓				GSO (multiobjective)

Chapter 2

Technical background

This chapter provides some technical background on the solutions methods put into practice in this dissertation. Section 2.1 first briefly introduces two important programming paradigms, namely constraint programming (CP) and linear programming (LP), and discusses their differences. Then, Section 2.2 presents the large neighborhood search metaheuristic framework. Next, Section 2.3 introduces the wide-spread exact approach known as Benders decomposition. It discusses its implementation and it point out some strategies that have been proposed in the literature for potentially improve the original method. Section 2.4 then briefly describes the Dantzig-Wolfe decomposition. Finally, Section 2.5 introduces robust optimization as a modeling framework widely-used to address the uncertainty inherent to real-life problems.

2.1 Constraint programming vs Linear programming

Nowadays, two different programming paradigms, namely LP and CP are widely used techniques in OR. These two paradigms may differ, among others, by the nature of the decision variables that can be used, the constraints that can be considered, and the resolution techniques that can be applied. Naturally, there usually exist several possible ways to model the same problem. The choice of one model over another one is then essentially made based on efficiency concerns. For the sake of completeness and clarity, we propose here a brief overview of LP and CP:

— LP is concerned with maximizing or minimizing a linear function over a convex polyhedron specified by linear equality and inequality constraints. The objective and the constraints are algebraically expressed in terms of the parameters and the (continuous) decision variables. Without lost of generality, in the remainder of this section, unless it is explicitly said we refer to the optimization problem at hand as a maximization problem. There essentially exist two different kinds of methods to solve a LP model: *simplex* methods and *interior point* methods (also known as barrier methods). Simplex methods explore in sequence the vertices of the polytope (by moving along the edges of the polytope) in non-decreasing values of the objective function until they prove that an optimum has been reached. In contrast to simplex algorithms, interior-point methods construct a sequence of feasible points moving through the interior of the polytope until they converge to an optimum on the boundary of this latter. Both methods are iterative, but, while simplex methods may require a large number of inexpensive iterations, interior point methods may converge in a small amount of expensive iterations. Regarding the efficiency of each method, the worst-case time complexity for simplex methods is exponential, while it is polynomial for interior point methods. However, the average-case analysis of simplex methods shows a polynomial time convergence that makes them competitive against interior point methods. Nonetheless, for very large scale and sparse problems, interior point methods seem to work better.

  When LP models contain some decision variables that are restricted to be integers (those variables are referred to as integer variables), we are then speaking about MILP or ILP if the model contains only integer variables. Solving MILP and ILP models is (theoretically) difficult as MILP (and therefore ILP) lie in the class of NP-hard problems. Nonetheless, *branch-and-bound* algorithms, *cutting plane* algorithms or an hybridization of these two algorithms are known to be successful – to a certain extent– in solving these models.

Branch-and-bound algorithms are enumeration techniques combining the partitioning of the search space (known as branching) with efficient bounding strategies. LP-based techniques start with the resolution of the continuous relaxation of the current model (the integrity constraints on the integer variables are relaxed yielding a LP model). Branching rules intend to exclude the LP solution by partitioning the problem into two or more sub-problems (the root node of the tree is associated with the original problem and a node is created for each new sub-problem). Branching can be performed on a single disjunction (a single variable at a time) or on a general disjunction (linear inequality on a group of variables). While the latter generally leads to a smaller size tree, solving the sub-problems may be more time-consuming, since new rows are introduced (which increases the size of the basis). Efficiency of branch-and-bound algorithms partially relies on finding the right disjunctions to branch on as this can limit the enumeration. The efficient exploration of the search tree is also impacted by the quality of the upper and lower bounds on the value of the objective function that are computed for each node throughout the search. At each node, the upper bound relies on solving the LP relaxation of the sub-problem, while the lower bound is usually based on heuristics. A comparison of these bounds allows to prune (i.e., remove) nodes that will never lead to an optimal solution (i.e., the lower bound is larger than the upper bound). Moreover, preprocessing techniques, such as fixings and variable aggregations, are often very useful since they usually lead to a smaller problem with a tighter relaxation.

Cutting plane algorithms work differently. They are based on the successive resolution of the LP relaxation of the model, strengthened at each iteration by the addition of a special type of valid inequalities known as cuts (or cutting planes). A valid inequality is a linear inequality that is redundant to the original model (i.e., it does not discard any feasible solution). It is called a cut if it separates the current LP solution (i.e. if adding this cut to the model eliminates the LP solution). The algorithm ends as soon as the LP solution is a solution to the original model or when the LP model becomes infeasible.

The hybridization of branch-and-bound and cutting plane algorithms yields what is commonly known as *branch-and-cut* algorithms. The elimination of the LP solution is either performed by branching or by the addition of valid cuts. These algorithms are known to be more efficient than considering the two previous methods separately.

For more details on linear and integer programming, we refer to (Schrijver, 1998; Wright, 1997).

— Unlike LP whose origins lie in the 1940s, the principles of CP were established in the late 1970s. CP consists in modeling the problem by a set of variables taking their values in a finite set or in an interval defined by real numbers and linked to each other by mathematical or/and symbolic constraints. Indeed, contrary to LP, CP allows logical constraints, non-linear constraints, and global constraints (a constraint that encapsulate a subset of other constraints) in addition to algebraic constraints. CP is therefore an attractive tool which makes expressing complex constraints relatively easy. The resolution of a CP model combines an enumeration phase with domain reductions strategies based on logical reasoning and constraint propagation techniques. A solution is found as soon as the domain of every variable contains a single value (i.e., all the variables are instantiated).

  Constraint propagation intends to limit the search space. It is based on filtering algorithms. A filtering algorithm is defined for a specific constraint and aims to reduce the domain of the involved variables by removing values that cannot be feasible for that constraint. During the propagation, filtering algorithms are applied in cascade again and again until the search reaches a *fix point* or meets certain predefined criteria (e.g., number of calls, activation only after the instantiation of one variable). A fix point is reached when applying the filtering algorithms in cascade to a given set of constraints does not lead to changes in the domain of any of the variables involved in those constraints. Filtering algorithms associated with global constraints can therefore take advantage of the knowledge of multiple constraints and then achieve stronger domain pruning.

Propagation itself is usually not enough to determine a solution. In that case, the domain of a non instantiated-variable is split into several sub-domains, each sub-domain being handled in a different sub-problem (also referred to as a node). This partitioning relies on heuristics (also called branching strategies) that define the variable on which the branching needs to be performed and the decision to apply for splitting its domain. Solving a CP model relies then on exploring a search tree. Constraint propagation is applied (by calling the filtering algorithms associated with the constraints of the model) to each sub-problem newly created. One still needs to define the order in which the nodes are explored. For memory consumption concerns, the most common exploration strategy is the depth first search (also known as chronological backtracking) since it restricts the number of active nodes. However, the first decisions taken using this strategy are very critical since the whole sub-tree needs to be explore before having the possibility to refute these decisions. To overcome this drawback, Harvey and Ginsberg (1995) introduced *Limited Discrepancy Search* (LDS). This technique relies on the intuition that branching strategies are efficient to lead to a solution or otherwise may only fail by making few mistakes (called *discrepancies*). LDS therefore restricts the search to paths which do not diverge more than k times from the choices recommended by the branching strategy (i.e., k is an upper bound on the allowed number of discrepancies). This yields a heuristic search. In order to define a complete search, one can always consider an iterative procedure based on increasing values of k.

There exist several ways to improve the exploration of the search space. Nogood recording is one of them. A nogood is a set of assignments and branching constraints that are not consistent with any solution (Rossi et al., 2006) and thus implies new constraints that can be propagated as the other constraints of the model. Nogood can help to record the past experience of the search. Alternatively, since the search can be stuck in local optima or in a sub-tree with no solution, restart policies can be implemented to allow diversification. The idea is too restart the search from the root node to explore another part of the search space. Branching heuristics need then to be randomized in order to explore different parts of the search space. Nogoods can also be added to this effect as in (Lecoutre et al., 2007).

CP is mostly applied to constraint satisfaction problems. Finding optimal solutions using CP is harder as it does not benefit from good bounds on the objective value. However, CP can be very powerful for highly constrained problems when compared to MILP.

Combinations of LP and CP have also been developed. For instance, reduced costs provided by the LP relaxation are used for domain reduction by Focacci et al. (1999). Hooker and Osorio (1999) introduced *mixed logical-linear programming* which combines LP relaxation and constraint propagation techniques. Jain and Grossmann (2001) proposed a set of hybrid algorithms based either on decomposition methods or branch-and-bound algorithms. One of the most successful hybrid method is *logic-based Benders decomposition* (LBBD) introduced by Hooker (2000). In that technique, the problem is decomposed into a master problem and a set of sub-problems and the method iterates between solving these problems. New constraints known as cuts are added to the master problem based on the sub-problem results. Thorsteinsson (2001) presented a closely related framework called B&C which requires solving the master problem only once (as opposed to the optimal resolution of the restricted master problem before solving the sub-problems). In this framework, the sub-problems are then checked all along the branch-and-bound tree. LBBD and B&C are very flexible in terms of the paradigm used to solve the sub-problems and thus in terms of the nature of the constraints that can be considered. If considering CP, cuts can be derived from nogoods or constraint propagation. Finally, CP can also be used to solve the column generation sub-problem in the Dantzig Wolfe decomposition (Gualandi and Malucelli, 2009).

CP is also commonly used in local search techniques as it can efficiently explore the neighborhood of a solution while considering complex constraints. The combination of mathematical programming techniques and (meta)heuristics (often called matheuristic) has also given satisfactory results.

2.2 Large Neighborhood Search

Large neighborhood search (LNS) is a metaheuristic originally proposed by Shaw (1998) for a vehicle routing problem. Due to its nature, LNS can be applied to a wide range of optimization problems. In LNS, the current solution is successively partially destroyed and repaired in order to improve its quality. It can be observe that the LNS is a local search technique as the destroy and repair operators implicitly define the neighborhood of the solution to which they are applied. Notice also that the method can be alternatively perceived as a combination of fixing and optimization phases. Indeed, destroying the solution is equivalent to fixing some parts of the solution while repairing the solution means re-optimizing it.

Algorithm 1 presents the general structure of the method. The algorithm enters an iterative process. It starts with an initial solution that may be provided by any heuristics. At each iteration, it partially destroys the current solution and then repairs this latter. If the resulting solution s' meets the acceptance criterion (later discussed), s' replaces the current solution s for the next iteration. If appropriate, the algorithm updates the best solution s^* found so far. Then, the search moves to the next iteration. The whole procedure is repeated until one of the stopping criteria is met. The optimization returns solution s^*.

The acceptance of the solutions built throughout the algorithm is an important component of LNS. The original version of LNS proposed by Shaw (1998) uses an elitist strategy to accept solutions (i.e. it accepts only improving solutions). On the other hand, the LNS by Pisinger and Ropke (2007) uses the Metropolis criterion to accept solutions. According to this criterion, solutions are accepted with a given probability. If the newly found solution s' improves the current solution s the probability equals to one. Otherwise, if f refers to the objective value function of the problem, the probability is computed using the Boltzmann expression: $e^{-(f(s)-f(s'))/T}$ (for a maximization problem). Parameter T is commonly known as the *temperature*. It is updated after each iteration using what is commonly known as the geometric cooling schedule: $T = T \times \sigma$, where $\sigma \in [0, 1[$. Then, the probability of accepting non-improving solutions decreases over the iterations. As the same solution can be the starting point of several iterations, randomization of operators usually proves to be necessary.

Ropke and Pisinger (2006) extended the LNS framework into what is known as *adaptive large neighborhood search* (ALNS). The fundamental difference comes from the implementation of several destroy and repair operators. The operators are selected with an adaptive layer according to their performance in the previous iterations. Performance does not necessarily mean improvement as operators leading to diversification are also interesting to avoid being trapped in local optima. Several strategies have been proposed to handle the selection of the operators. Traditionally a weight is assigned to each destroy and repair operators independently, but the score can be directly assigned to each destroy-repair operator pair. The operators are then selected according to these values. The weights are updated at each iteration or after a certain number of iterations (usually referred to to as a segment). A decay or reaction factor controls the degree of response to the changes in the effectiveness of the heuristics. The intuition behind this enhancement is to let the algorithm adapt to the current state of the search by choosing the best suitable operators. Indeed, some operators may prove to be efficient to significantly improve the solution during the first iterations, but they might stuck the algorithm in local optima later on. We therefore expect other operators to take over. Algorithm 2 presents the general structure of the method. It only contains few changes from the LNS structure. Henceforth, at the beginning of each iteration, it now randomly selects (according to some weights) a destroy operator o_1 and a repair operator o_2. It also records their efficiency and updates their weights at the end of each iteration.

It is noteworthy that ALNS is controlled by many parameters mainly coming from the adaptive layer. However, this can be problematic, since finding the right combination of values requires a lot of tuning experiments. This may also question the efficiency of the method as well as its applicability to instances that were not originally considered in the calibration process. As suggested by Pisinger and Ropke (2007), the operators can simply be selected with equal probability.

Algorithm 1: Script of the LNS algorithm

$s \leftarrow$ initial solution
$s^* \leftarrow s$
repeat
  $s' \leftarrow$ **repair** (**destroy** $(s), s)$
  if s' is accepted **then**
    $s \leftarrow s'$
  end
  if s' is strictly better than s^* **then**
    $s^* \leftarrow s'$
  end
until stopping criteria are not met;
return s^*

Algorithm 2: Script of the ALNS algorithm

$s \leftarrow$ initial solution
$s^* \leftarrow s$
$n \leftarrow 1$ (current number of iterations)
repeat
  $(o_1, o_2) \leftarrow$ **selectOperators**(n) .
  $s' \leftarrow$ **repair** $(o_2,$ **destroy** $(o_1, s), s)$
  if s' is accepted **then**
    $s \leftarrow s'$
  end
  if s' is strictly better than s^* **then**
    $s^* \leftarrow s'$
  end
  Update weights of o_1 and o_2 according to the potential solution improvement and to the value of n
  $n \leftarrow n + 1$
until stopping criteria are not met;
return s^*

Moreover, one can consider repairing a solution using MILP or even CP formulations. Indeed, since the problem is reduced, these latter usually become tractable. One can then expect the exploration of larger neighborhoods and a potential faster convergence to high-quality solutions.

2.3 Benders decomposition

In many cases, tackling the full problem all at once is very complex, especially if one is interested in an exact approach. For example, directly solving MILP formulations on large-scale instances is very likely to be impractical. For this reason, decomposition techniques have been widely studied in mathematical programming and, in particular, in LP. One of the most famous method is the Benders decomposition introduced by Benders (1962). This method decouples a large-scale problem into a master problem and one or several independent small-scale subproblems which are easier to solve.

Let us consider the following general class of MILP problems where $c \in \mathbb{R}^{n_1}$,$f \in \mathbb{R}^{n_2}$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{n_1 \times m}$, $B \in \mathbb{R}^{n_2 \times m}$, $X \subseteq \mathbb{R}^{n_1}_+$ and $Y = \mathbb{R}^{n_2}_+$ (the sets X and Y will be discussed further below).

$$\text{maximize} \quad c^T x + f^T y \tag{2.1}$$
$$\text{s.t.} \quad Ax + By \leq b \tag{2.2}$$
$$x \in X \tag{2.3}$$
$$y \in Y \tag{2.4}$$

The idea underlying Benders decomposition is to project the problem (2.1)-(2.4) onto the space defined by the variables x. To this end, let us defined the sub-problem according to a vector $x \in X$.

$$\begin{aligned} \text{maximize} \quad & f^T y \\ \text{s.t.} \quad & By \leq b - Ax \\ & y \in Y \end{aligned}$$

The dual of this sub-problem reads:

$$\begin{aligned} \text{minimize} \quad & (b - Ax)^T \lambda \\ \text{s.t.} \quad & B^T \lambda \geq f \\ & \lambda \in \mathbb{R}^m_+ \end{aligned}$$

Let us introduce Π as the set of *extreme rays* of the cone $\{\pi \in \mathbb{R}^m \mid B^T\pi \geq 0\}$ and Λ as the set of *extreme points* associated with the polytope $\{\lambda \in \mathbb{R}^m_+ \mid B^T\lambda \geq f\}$. Using Farkas' Lemma and duality theory, we then obtain the following *master problem* equivalent to the original problem:

$$\text{maximize} \quad c^T x + z \tag{2.5}$$

$$\text{s.t.} \quad (b - Ax)^T \pi \geq 0 \quad \forall \pi \in \Pi \tag{2.6}$$

$$(b - Ax)^T \lambda \geq z \quad \forall \lambda \in \Lambda \tag{2.7}$$

$$x \in X \tag{2.8}$$

The constraints (2.6) and (2.7) are called *Benders optimality cuts* and *Benders feasibility cuts*. The number of these constraints is finite but may be very large. However, they usually do not need to be exhaustively generated, since only a subset of them will be active in an optimal solution. An iterative cutting plane algorithm can thus be used to generate only the subset of cuts that will yield an optimal solution.

In the following, the term *restricted master problem* refers to a problem (2.5)-(2.8) that contains none (or only a small subset) of constraints (2.6) and (2.7).

The original implementation of Benders decomposition relies on *Kelley's cutting plane algorithm* (Kelley, 1960), which was originally proposed to solve convex non-differentiable problem and is based on the idea that each convex function can be approximated by piecewise linear functions. The algorithm successively solves a restricted master problem and the sub-problem. If the sub-problem is infeasible, this means that the partial solution provided by the current restricted master problem does not yield a feasible solution to the original problem. A violated Benders feasibility cut (2.6) is then identified and introduced in the restricted master problem to discard this partial solution. If the sub-problem is not a constraint satisfaction problem (i.e., $f \neq 0$), a violated Benders optimality cut (2.7) is identified when the sub-problem is feasible and introduced in the restricted master problem. The method iterates between the restricted master problem and the sub-problem until it converges or concludes that there is no solution. Figure 2.1 presents the scheme of the original implementation of Benders decomposition. Notice that the primal or the dual of the sub-problem can be solved equivalently, the choice depending usually on efficiency consideration. This method has been successfully applied to (the list is not exhaustive): power systems (Shahidehopour and Fu, 2005), hub network design (Cordeau et al., 2006), cargo shipping (Agarwal and Ergun, 2008).

One noticeable disadvantage of the original implementation of Benders decomposition is that the restricted master problem is repeatedly solved to optimality. Moreover, the time needed to solve it tends to increase with each iteration since new constraints are introduced. To overcome this drawback, as an alternative implementation, Benders cuts can be generated dynamically in the branch-and-bound tree used to solve an initial restricted master problem. The sub-problem is then solved during the search for a solution to the master problem. More specifically, at each integer node of the branch-and-bound tree, the corresponding solution is sent to the sub-problem in order to generate the Benders cuts. This works since the Benders cuts are valid inequalities for the problem. Naoum-Sawaya and Elhedhli (2010) gave a proof of this latter statement. Figure 2.2 presents the scheme of this alternative implementation of Benders decomposition. This approach is referred as a *Benders-based branch-and-cut* algorithm in (Naoum-Sawaya and Elhedhli, 2010) or as a *branch-and-Benders-cut method* in (Gendron et al., 2014). It has been used to solve several types of problems: hub location (De Camargo et al., 2011), production routing under demand uncertainty (Adulyasak et al., 2015), location-design (Gendron et al., 2014), facility location and network design (Naoum-Sawaya and Elhedhli, 2010), and hop-constrainted survivable network (Botton et al., 2013). Botton et al. (2013) reported a significant improvement using this method compared to the classical Benders decomposition, while Gendron et al. (2014) outlined the benefits of the Benders-based branch-and-cut in terms of solution quality, scalability, and robustness.

The difference between the two implementations previously described is very similar to the difference between two methods primarily used for LP and CP hybridization: the *Logic-based Benders* *decomposition* and the B&C framework (the two methods are described in Section 2.1). As with the Benders-based branch-and-cut, the B&C framework method involves a single resolution of the restricted master problem. Thorsteinsson (2001) applied it to a planning and scheduling problem, while Sadykov (2008) used it for a complex scheduling problem on a single machine.

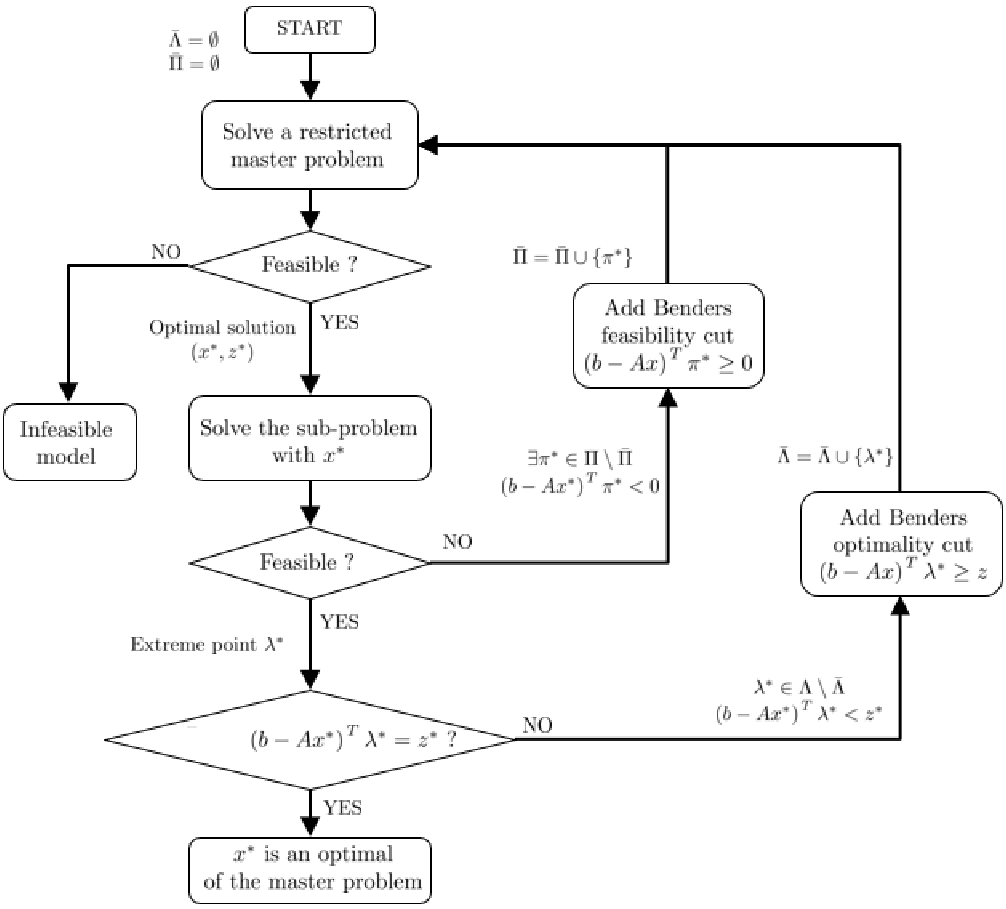


Figure 2.1 – Scheme of the original implementation of Benders decomposition

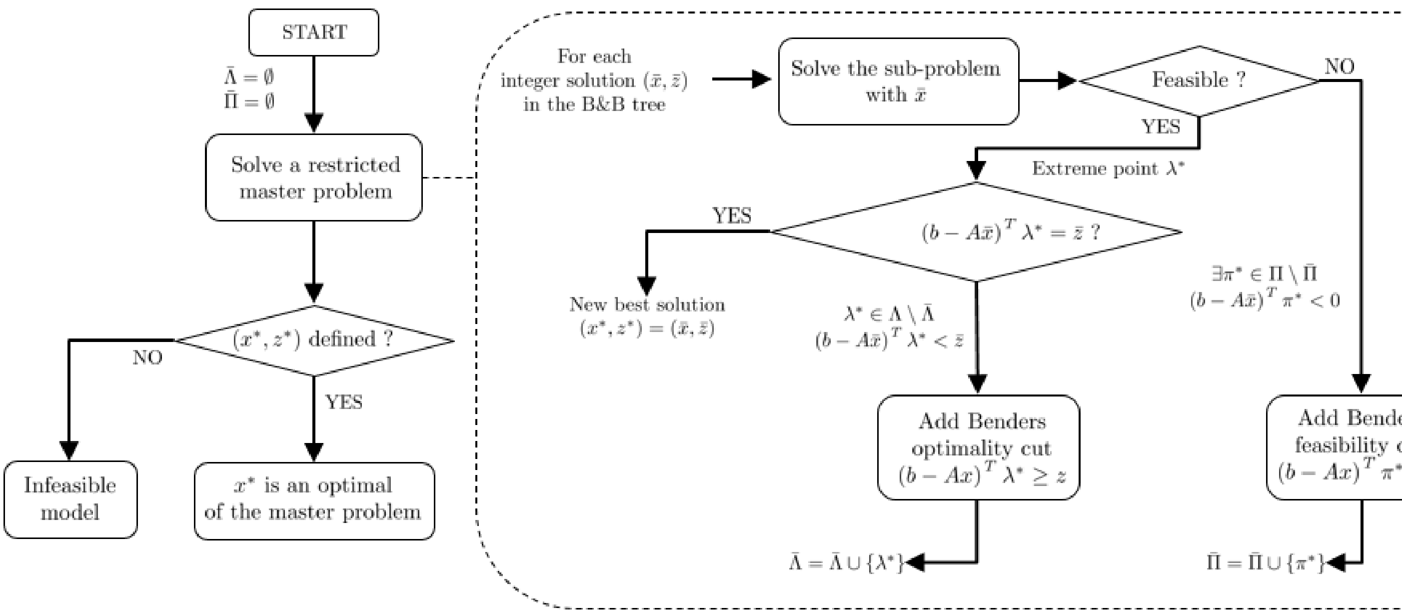


Figure 2.2 – Scheme of the alternative implementation of Benders decomposition

Researchers have proposed a number of improvements to the general Benders decomposition framework. These improvements usually focus on speeding up the convergence of the method (which is typically an issue).

Observing that the sub-problem has usually more than one optimal solutions, the quality of the optimality Benders cuts may vary substantially from one chosen extreme point to another. Magnanti and Wong (1981) thus proposed a technique to generate the optimality cut that dominates all the other optimality cuts. This specific cut is said *Pareto-optimal*. Generating the strongest possible cut is naturally meant to speed-up the convergence. Nonetheless, the method requires a Benders master problem core point and an auxiliary problem needs to be solved along with the sub-problem. As a practical enhancement, Papadakos (2008) suggested to use an approximation to the core point as a convex linear combination of the optimal solutions provided when solving the restricted master problem in the previous iterations. Moreover, the author defined a new auxiliary problem, independent from the solution to the sub-problem, as an alternative to the one introduced by Magnanti and Wong (1981) for which the two previous elements are interdependent.

McDaniel and Devine (1977) proposed an efficient strategy to generate an initial set of valid cuts. The idea is to first solve the linear relaxation of the problem using Benders decomposition. While dealing with a much easier problem, this may identify potential useful Benders cuts.

In decompositions where it is hard to obtain a feasible sub-problem, a large number of Benders feasiblity cuts are generated but no information is provided to the restricted master problem about the sub-problem's objective (since no optimality cuts are generated). When the sub-problem is infeasible, Saharidis and Ierapetritou (2010) generated a cut having the same form as a Benders optimality cut. This cut is based on solving the maximum feasible subsystem of the sub-problem.

In the discussions above, we have considered a mixed-integer master problem ($X \subseteq \mathbb{R}^{n_1}_+$) and a continuous sub-problem ($Y = \mathbb{R}^{n_2}_+$). However, having a mixed-integer or a pure integer sub-problem is common in practice. If the sub-problem does possess the integrality property (i.e., the constraint matrix of the sub-problem is totally unimodular), solving its linear relaxation leads to an optimal solution y^* that belongs to Y. Otherwise, the classical Benders decomposition framework is not well-adapted to solve the problem. However, with mild changes, it can still apply in the case where the x variables are all restricted to be binary. When the sub-problem is infeasible, one generates additional cuts (called integer Benders cuts) to invalidate the current solution to the restricted master problem. Codato and Fischetti (2006) introduced the name *combinatorial Benders cuts* to refer to these cuts. Let C be the set of indices of the variables x restricted to be binary and x^* the current solution to the restricted master problem. Denoting $S = \{i \in C | x^*_i = 1\}$, a combinatorial Benders cut can be defined as follows:

$$\sum_{i \in S} (1 - x_i) + \sum_{i \in C \setminus S} x_i \geq 1 \tag{2.9}$$

Clearly, this cut states that, in the next iteration, at least one of the variables of the master problem must change its value with respect to the current solution x^*. This cuts is similar to the cuts defined in the Logic-based Benders decomposition by Hooker and Ottosson (2003). This cut is also known as a *no-good* cut. Although not required and not self-sufficient, the classical Benders cuts can still be generated for efficiency purposes since they potentially invalidate more than just the current solution to the problem.

When the sub-problem is feasible (and we are not dealing with a constraint satisfaction problem), one has to generate integer optimality cuts as those introduced by Laporte and Louveaux (1993) for the integer L-shaped method. If we assume that the master problem is bounded by UB and we denote h^* the optimal value of the sub-problem, the cut to add to the restricted master problem is defined as follows:

$$(h^* - UB) \left(\sum_{i \in C \setminus S} x_i - \sum_{i \in S} x_i \right) - (h^* - UB)(|S| - 1) + UB \geq z \tag{2.10}$$

If some of the x variables are restricted to be integer (not binary), these variables must be expressed using additional binary variables. The previous cuts (2.9) and (2.10) can then be expressed with these new variables. Notice that those cuts are very weak as they just discard the current solution.

All the cuts mentioned in this part are generic. As a general principle, the efficiency of Benders decomposition is highly related to the strength of the cuts introduced in the restricted master problem. Generating problem-specific cuts can therefore plays a key role. These cuts can replace or be used along with the generic Benders cuts.

2.4 Dantzig-Wolfe decomposition

As Benders decomposition, Dantzig-Wolfe decomposition is an exact approach to solve complex LP formulations.

Let us consider the following general class of LP problems where $c \in \mathbb{R}^n$, $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$, $A_1 \in \mathbb{R}^{n \times m_1}$, $A_2 \in \mathbb{R}^{n \times m_2}$.

$$
\begin{aligned}
[P] \quad \text{maximize} \quad & c^T x && (2.11)\\
\text{s.t.} \quad & A_1 x \leq b_1 && (2.12)\\
& A_2 x \leq b_2 && (2.13)\\
& x \in \mathbb{R}^n_+ && (2.14)
\end{aligned}
$$

The x variables can be written as follows:

$$x = \sum_{q \in Q} \lambda^q x^q + \sum_{r \in R} \eta^r x^r$$

where $\{x^q\}_{q \in Q}$ are the extreme points and $\{x^r\}_{r \in R}$ are the extreme rays of the polyhedron $\{x \in \mathbb{R}^n | A_2 x \leq b_2\}$. The problem (2.11)-(2.14) can be rewritten:

$$
\begin{aligned}
[PM] \quad \text{maximize} \quad & \sum_{q \in Q} \left(c^T x^q\right) \lambda^q + \sum_{r \in R} \left(c^T x^r\right) \eta^r && (2.15)\\
\text{s.t.} \quad & \sum_{q \in Q} \left(A_1 x^q\right) \lambda^q + \sum_{r \in R} \left(A_1 x^r\right) \eta^r \leq b_1 && (2.16)\\
& \sum_{q \in Q} \lambda^q = 1 && (2.17)\\
& \lambda \in \mathbb{R}^n_+ \quad \forall q \in Q && (2.18)\\
& \eta \in \mathbb{R}^n_+ \quad \forall r \in R && (2.19)
\end{aligned}
$$

$[PM]$ is the Dantzig-Wolfe reformulation of $[P]$. Notice that this new formulation $[PM]$ has m_2 less constraints than this original formulation $[P]$, but the number of variables is drastically larger here. Indeed, although the sets Q and R are finite, they may contain an exponential number of points.

Let us just focus here on the case where the previous polyhedron is bounded ($R = \emptyset$). Let us assume that we only know a subset $Q^* \subseteq Q$ of these extreme points. The problem $[PM]$ restricted to this set Q^* reads:

$$
\begin{aligned}
[PMR(Q^*)] \quad \text{maximize} \quad & \sum_{q \in Q^*} \left(c^T x^q\right) \lambda^q && (2.20)\\
\text{s.t.} \quad & \sum_{q \in Q^*} \left(A_1 x^q\right) \lambda^q \leq b_1 && (2.21)\\
& \sum_{q \in Q^*} \lambda^q = 1 && (2.22)\\
& \lambda \in \mathbb{R}^n_+ \quad \forall q \in Q^* && (2.23)
\end{aligned}
$$

Let us now associate the dual vector π and the dual variable μ to the constraints (2.20) and (2.23) of $[PMR(Q^*)]$. The reduced cost of a variable λ^q is thus equal to $\left(c^T - \pi^T A_1\right) x^q - \mu$. If the reduced cost of all points in Q is less or equal to 0, then the solution to problem (2.20)-(2.23) is the optimal solution to problem (2.15)-(2.19). Otherwise, there exists an extreme point $\hat{x}^q$ in $Q \setminus Q^*$ such that $\left(c^T - \pi^T A_1\right) \hat{x}^q - \mu > 0$. One way to check the existence of such a point is to solve the problem $[PP(\pi, \mu)]$ for given dual values π and μ.

$$
\begin{aligned}
[PP(\pi,\mu)] \quad \text{maximize} \quad & \left(c^T - \pi^T A_1\right) x - \mu && (2.24)\\
\text{s.t.} \quad & A_2 x \leq b_2 && (2.25)\\
& x \in \mathbb{R}^n_+ && (2.26)
\end{aligned}
$$

The problem $[PP(\pi, \mu)]$ is called *pricing problem*. If the optimum value of this problem is strictly greater than 0, then the optimal solution x^* of $[P]$ is such that $x^* = x^{q*}$ with $q^* \in Q \setminus Q^*$. Therefore, this extreme point needs to be added to the restricted problem $[PMR(Q^*)]$ (i.e. $Q^* = Q^* \cup \{q^*\}$) as variable λ^{q^*} should enter the basis. Otherwise (i.e., the optimum value is less or equal to 0), an optimal solution to problem $[P]$ has been reached.

The previous remark suggests an iterative algorithm to solve problems rewritten using Dantzig-Wolfe decomposition. Such an algorithm is known as the *column generation* method. It generally starts by initializing the set Q^* using one or more heuristic solutions. Then, the problem $[PMR(Q^*)]$ is solved, yielding a solution x^* and dual values π and μ. Solving the pricing problem $[PP(\pi, \mu)]$ allows to check the optimality of the solution x^*. The method iterates between the restricted problem and the pricing problem as long as there exists a column (i.e., a variable λ^q) with a strictly positive reduced cost. The motivation of this method comes from the observation that a lot of variables λ^q are equal to 0 at the optimum, and the exponential size of Q makes inefficient handling all the columns at once. Moreover, computing Q is usually computationally prohibitive.

The Dantzig-Wolfe decomposition can also apply to MILP and ILP problems. If we assume that the problem $[P^{IP}]$ is bounded, this latter can be rewritten as problem $[PM^{IP}]$.

$$
\begin{aligned}
[P^{IP}] \quad \text{maximize} \quad & c^T x && (2.27)\\
\text{s.t.} \quad & A_1 x \leq b_1 && (2.28)\\
& A_2 x \leq b_2 && (2.29)\\
& x \in \mathbb{N}^n_+ && (2.30)
\end{aligned}
$$

$$
\begin{aligned}
[PM^{IP}] \quad \text{maximize} \quad & \sum_{q \in Q} \left(c^T x^q\right) \lambda^q && (2.31)\\
\text{s.t.} \quad & \sum_{q \in Q} \left(A_1 x^q\right) \lambda^q \leq b_1 && (2.32)\\
& \sum_{q \in Q} \lambda^q = 1 && (2.33)\\
& \lambda^q \in \{0,1\}^n \quad \forall q \in Q && (2.34)
\end{aligned}
$$

Solving the ILP problem involves a combination of branch-and-bound and column generation. This method is known as *branch-and-price*. Column generation is used to solve the linear relaxation of the problem restricted to a pool of columns. Branching generally occurs when two conditions are satisfied: no columns can enter the basis and the LP solution does not satisfy the integrality conditions. The efficiency of branch-and-price primarily relies on fast algorithms to solve the pricing problem (note that it has become a ILP formulation) and on effective branching strategies. For example, branching on the variables λ is usually not the right choice as their number is very large. On the opposite, branching on the variable x of the original problem seems preferable as it reduces the size of the search tree and can generally be handled easily in the pricing problem. For more details on column generation and branch-and-price, we refer to (Barnhart et al., 1998; Desaulniers et al., 2006).

Lastly, it is noteworthy that applying Benders decomposition to a LP problem is equivalent to applying Dantzig Wolfe decomposition to the dual of this problem.

2.5 Robust optimization

It is very common that optimization problems involve uncertain data. One can work with nominal or expected values for these uncertain parameters, but this, when implemented in practice, can lead to infeasible or poor-quality solutions.

To specifically deal with uncertainty, one can distinguish two main methodologies: stochastic programming and robust optimization. While stochastic programming has been widely applied in the OR field, robust optimization has only recently received due attention (despite being introduced back the early 70's by Soyster (1973)). Choosing which approach is the most suitable in a given context depends on the problem to solve and particularly on the available data. On the one hand, stochastic programming relies massively on the availability of historical data to derive probability distributions or generate scenarios (as in the sample average approximation method). Moreover, this approach may suffer from the difficulty of fitting probability distributions to the uncertain data. It can also be complicated to identify or generate scenarios and their probability. Indeed, one should determine the number of scenarios that leads to a good-quality estimation of the uncertainty (which is usually not easy). In addition, the number of scenarios is required to be small for computational efficiency, but this might limit the range of future states under which the decisions are evaluated. In some cases, robust optimization appears then to be a valuable alternative to stochastic programming. Indeed, robust optimization requires a very limited information about the uncertain parameters, since it only needs a discrete or convex description of the uncertainty. From a practical point of view, robust optimization is often easier to implement and usually very understandable by decision-makers. Whereas stochastic programming usually increases the complexity of the problem to solve, the robust counterpart of uncertain problems is computationally tractable if the uncertainty sets defined in robust optimization satisfies mild convexity and computability assumptions (e.g., explicit systems of equations) (Ben-Tal et al., 2009). However, while robust optimization models are usually designed for risk-averse decision-making (immunization against uncertainty), stochastic programming is more flexible, as one can search to compute or estimate the expectation function or to control the risk (e.g., considering the average value-at-risk measure). Deciding the best approach to a problem is not always clear, but, as pointed out by Roy (2010), one can always state that every model is an approximation to a real-world decision problem.

In the following, we focus on robust optimization. Originally, robust optimization was developed to compute a solution that is feasible for any possible realization of the uncertain parameters and/or to guarantee a certain solution value. Robust optimization is therefore often known as worst-case optimization.

Let us consider the following general class of MILP problems where $c = (c_j) \in \mathbb{R}^n$, $b = (b_i) \in \mathbb{R}^m$, $A = (a_{ij}) \in \mathbb{R}^{n \times m}$, and $X \subseteq \mathbb{R}^m_+$.

$$\text{maximize} \quad \sum_{j=1}^{m} c_j x_j \tag{2.35}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} a_{ij} x_j \leq b_i \quad \forall i = 1...n \tag{2.36}$$

$$x \in X \tag{2.37}$$

Conventionally, one assumes that each uncertain coefficient (parameter) a_{ij} belongs to an interval $[\bar{a}_{ij} - \hat{a}_{ij}; \bar{a}_{ij} + \hat{a}_{ij}]$ where $\bar{a}_{ij}$ and $\hat{a}_{ij} \geq 0$ are given in advance. The parameter $\bar{a}_{ij}$ is often referred to as the nominal value of coefficient a_{ij}. For the sake of simplicity, we only consider here symmetric intervals. The concept below apply also for non symmetric intervals.

The most common robust approach considers row-wise uncertainty, meaning that the constraints are independently protected against the uncertainty. For the $i - th$ row, let us introduce U_i as the set of all possible values for vector $(a_{\cdot j})$. We refer to this set as the *uncertainty set* of the parameters. The robust counterpart of the problem (2.35)-(2.37) – hereafter also referred to as the robust problem –

consists in the following problem:

$$\text{maximize} \quad \sum_{j=1}^{m} c_j x_j \tag{2.38}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} a_{ij} x_j \leq b_i \quad \forall i = 1...n, \forall (a_{i\cdot}) \in U_i \tag{2.39}$$

$$x \in X \tag{2.40}$$

Different uncertainty sets have been introduced in the literature. For the sake of comparison, we define the set U_i to be dependent on another set V_i (defined further below) as follows:

$$U_i = \{(a_{i\cdot}) \ |\exists (\xi_{i\cdot}) \in V_i, \ a_{ij} = \bar{a}_{ij} + \hat{a}_{ij}\xi_{ij} \ \ j = 1...m\}$$

The robust problem (2.38)-(2.40) is then equivalent to the problem:

$$\text{maximize} \quad \sum_{j=1}^{m} c_j x_j \tag{2.41}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} \bar{a}_{ij} x_j + \max_{(\xi_{i\cdot}) \in V_i} \sum_{j=1}^{m} \hat{a}_{ij}\xi_{ij} x_j \leq b_i \quad \forall i = 1...n \tag{2.42}$$

$$x \in X \tag{2.43}$$

We now present some of the most commonly used uncertainty sets.

— Soyster (1973) considered the full uncertainty set as V_i is defined as follows:

$$V_i = \{(\xi_{i\cdot}) \mid \|\xi_{i\cdot}\|_\infty \leq 1\}$$

The robust counterpart (2.41)-(2.43) is then equivalent to the problem where each uncertain coefficient takes its extreme value:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{m} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{m} (\bar{a}_{ij} + \hat{a}_{ij}) x_j \leq b_i \quad \forall i = 1...n \\ & x \in X \end{aligned}$$

The over-conservatism of robust optimization is then a common criticism, as one easily observe that the previous approach can result in very poor quality solutions because it is optimal for extreme values of the parameters but usually not for nominal values. To address this issue, one can think to reject extreme realizations from the previous uncertainty set, since it is very unlikely that all the uncertain parameters take extreme values. This strategy motivates the definition of the following uncertainty sets:

— Bertsimas and Sim (2003) proposed to consider a polyhedral uncertainty set – intersection of the L^1-norm and the L^∞-norm – where for each row i the deviations from the nominal values are bounded by a value $\Gamma_i \geq 0$ called the *uncertainty budget*. The uncertainty budget allows to control the robustness of the solution. The value of Γ_i is fixed by decision-makers as a trade off between robustness and performance. Setting Γ_i to 0 means considering the nominal values for all the coefficients whereas setting Γ_i to m is equivalent to considering the approach of Soyster (1973). Bertsimas and Sim (2004) suggested to fix the uncertainty budget in the order of the square of the total number of uncertain parameters.

$$V_i = \{(\xi_{i\cdot}^-, \xi_{i\cdot}^+) \mid \xi_{ij}^- \geq 0,\ \xi_{ij}^+ \geq 0,\ \|\xi_{i\cdot}^- + \xi_{i\cdot}^+\|_1 \leq \Gamma_i,\ \|\xi_{i\cdot}^- + \xi_{i\cdot}^+\|_\infty \leq 1\}$$

Using duality theory, the robust problem (2.41)-(2.43) can be reformulated as:

$$\text{maximize} \quad \sum_{j=1}^{m} c_j x_j \tag{2.44}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} \bar{a}_{ij} x_j + \Gamma_i \nu + \sum_{j=1}^{m} \left(\gamma_{ij}^- + \gamma_{ij}^+\right) \leq b_i \quad \forall i = 1...n \tag{2.45}$$

$$\nu + \gamma_{ij}^- \geq -\hat{a}_{ij} x_j \quad \forall i = 1...n, \forall j = 1...m \tag{2.46}$$

$$\nu + \gamma_{ij}^+ \geq \hat{a}_{ij} x_j \quad \forall i = 1...n, \forall j = 1...m \tag{2.47}$$

$$\nu \geq 0 \tag{2.48}$$

$$\gamma_{ij}^-, \gamma_{ij}^+ \geq 0 \quad \forall i = 1...n, \forall j = 1...m \tag{2.49}$$

$$x \in X \tag{2.50}$$

This approach has the particularity to maintain the linearity of the problem, which makes the robust counterpart as tractable as the nominal problem (i.e., the problem where each uncertain parameter takes its nominal value). One can directly solve the reformulation of the problem as proposed in (Bertsimas and Thiele, 2006; Fischetti and Monaci, 2012; Babonneau et al., 2013; Bertsimas et al., 2016). Alternatively, one can use a cutting plane algorithm as proposed by Fischetti and Monaci (2012) and Bertsimas et al. (2016). In this approach, we separate the constraints (2.39). The computational experiments conducted in the two latter studies suggest that a cutting plane approach usually outperforms the resolution of the linear reformulation of the robust problem.

— Ben-Tal and Nemirovski (2002) proposed an ellipsoidal uncertainty set where the L^2-norm of the deviations is bounded for each row i by a predefined parameter $\Omega_i \geq 0$. As for the previous approach, this parameter controls the degree of overconservatism of the solution.

$$V_i = \{(\xi_{i\cdot}^-, \xi_{i\cdot}^+) \mid \xi_{ij}^- \geq 0,\ \xi_{ij}^+ \geq 0,\ \|\xi_{i\cdot}^- + \xi_{i\cdot}^+\|_2 \leq \Omega_i,\ \|\xi_{i\cdot}^- + \xi_{i\cdot}^+\|_\infty \leq 1\}$$

The robust counterpart (2.41)-(2.43) is then equivalent to the problem (see for example (Babonneau et al., 2009) for a proof):

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{m} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{m} \bar{a}_{ij} x_j + \sum_{j=1}^{m} \hat{a}_{ij} y_j + \Omega_i \sqrt{\sum_{j=1}^{m} (\hat{a}_{ij} z_j)^2} \leq b_i \quad \forall i = 1...n \\
& x_j \leq z_j + y_j \quad \forall j = 1...m \\
& y_j \geq 0 \quad \forall j = 1...m \\
& z_j \gtreqless 0 \quad \forall j = 1...m \\
& x \in X
\end{aligned}$$

This approach has the drawback of changing a linear problem into a quadratic problem (thus non-linear). This makes solving the robust counterpart of the problem more complex than solving the nominal problem.

This short (and not exhaustive) overview shows that tractability and conservatism considerations are very important when modeling the uncertainty. Indeed, the definition of the uncertainty set largely impacts the expression and the complexity of the robust counterpart of a problem. Other norms have been studied in (Bertsimas et al., 2004), but, to our knowledge, they have not been applied to practical problems. For interested readers, Ben-Tal et al. (2015) proposed, in turn, a list of robust counterparts for different types of linear and non-linear constraints.

The uncertainty set can also be a finite set as proposed by Kouvelis and Yu (1997). Let S_i be the set of scenarios for row i and a_{ij}^s the value of coefficient a_{ij} in scenario $s \in S_i$.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{m} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{m} a_{ij}^s x_j \leq b_i \quad \forall i = 1...n, \forall s \in S_i \\
& x \in X
\end{aligned}
$$

The coefficient c of the objective function can also be uncertain. Although different single robustness measures can be considered (e.g., minimization of the absolute or relative regret in the worst case scenario), the most commonly used is the maximization of the solution value in the worst-case. In other words, when the uncertainty is in the objective function, robust optimization seeks to produce a solution that performs the best in the worst-case. All the previous approaches apply since the objective can always be considered as a constraint by simply adding an auxiliary variable. However, this classical approach suffers from the fact that it only considers a single realization of the parameters and does not depend on the other realizations that are part of the uncertainty set. It may be interesting to have a solution that performs decently in the worst-case but very good on the majority of scenarios. This is however not guaranteed when using the previous approaches.

To accomplish the aforementioned goals, Roy (2010) proposed new measures of robustness based on the definition of two values $B, W \in \mathbb{R}$. According to the direction of the optimization, the parameter W corresponds to a value that needs to be satisfied for all the realizations of the uncertainty, whereas B is a value that we want to exceed or not in the largest number of scenarios. Let S be the discrete set of scenarios that contains all the realizations of the uncertain cost vector c and c^s the vector associated with scenario $s \in S$.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{s=1}^{q} y_s \\
\text{s.t.} \quad & \sum_{j=1}^{m} c_j^s x_j \geq W(1 - y_s) + B y_s \quad \forall s \in S \\
& \sum_{j=1}^{m} a_{ij} x_j \leq b_i \quad \forall i = 1...n \\
& y_s \in \{0, 1\} \quad \forall s \in S \\
& x \in X
\end{aligned}
$$

Originally proposed when considering finite sets of scenarios, Gabrel et al. (2013) extended this approach to the case where each parameter takes its value within an interval. However, this comes with an increased complexity of the robust counterpart of the problem.

An alternative approach is to optimize the problem considering the nominal values of the uncertain parameters in the objective function, while ensuring that in the worst-case the value of the solution is greater than W. In that case, assuming that each coefficient c_j belongs to the interval $[\bar{c}_j - \hat{c}_j; \bar{c}_j + \hat{c}_j]$ and denoting U the uncertainty set associated with the cost vector c, the robust counterpart of the problem is defined as follows:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{m} \bar{c}_j x_j \\
\text{s.t.} \quad & \min_{c \in U} \sum_{j=1}^{m} c_j x_j \geq W \\
& \sum_{j=1}^{m} a_{ij} x_j \leq b_i \quad \forall i = 1 ... n \\
& x \in X
\end{aligned}
$$

So far, we have considered row-wise uncertainty with the uncertainty being located in the objective or in the constraint matrix. Notice that the uncertainty can occur on the right-hand side of the constraints (vector b). The applicability of the previous approaches is not challenged as one can consider fixed variables associated with each constraint. The relevance is however questioned as only one coefficient is uncertain per constraint. It should be better to consider a single uncertainty set for the vector b. This approach is known as *column-wise uncertainty*. Soyster (1973) was the first one to investigate it by simply replacing each parameter b_i by its minimal value in the uncertainty set. Aside from this first approach, column-wise uncertainty is essentially investigated by introducing the possibility to take recourse actions once the uncertainty has been revealed. This approach is often known as *adaptive robust optimization* (on the opposite, the term *static robust optimization* can be associated with row-wise uncertainty) and yields two-stage problems. The objective is still to protect against the worst-case scenario, but recourse actions are defined – as in multi-stage stochastic programming – after the uncertainty has been revealed. In the two-stage case, the recourse problem is a non-convex max-min (or min-max) problem that can be solved by cutting plane techniques or by reformulating it as a MILP problem (using duality theory and linearization techniques). The full multi-stage (traditionally two-stage) robust problem can be solved by a cutting plane method (Minoux, 2011; Souyris et al., 2012; Jiang et al., 2012; Zeng and Zhao, 2013; Billionnet et al., 2014; Gabrel et al., 2014a; Lorca and Sun, 2014). A full recourse property is often assumed but is not required. Jiang et al. (2012) derived feasibility cuts inside a cutting plane approach in a case in which the full-recourse property does not hold. Since two-stage robust optimization problems can be intractable when there is large flexibility regarding the recourse decisions, Ben-Tal et al. (2004); Ardestani-Jaafari and Delage (2016) restricted the recourse decisions to be affine functions to the uncertain parameters which yields in approximate policies. For more details, we refer to (Roy, 2010; Bertsimas et al., 2011; Gabrel et al., 2014b) for extensive surveys and to (Ben-Tal et al., 2009) for an exhaustive book on robust optimization.

In this chapter, we have presented two different programming paradigms, namely LP and CP, and pointed out their differences. We have introduced LNS as a framework to heuristically solve optimization problems. We have also described two decomposition methods (Benders and Dantzig-Wolfe decompositions) to exactly solve large-scale problems. Finally, we have introduced robust optimization as a methodology to tackle problems containing uncertain parameters. This technical background intends to help the reader understand the solution methods put in practice or discussed in Part II.

II

A wind turbine maintenance scheduling problem in the onshore wind industry

In this part, we introduce a challenging maintenance scheduling problem with resource management rising in the onshore wind power industry. This problem is deeply inspired by our collaboration with WPred, a Montreal-based company specialized in the supply of weather forecasts and, more specifically, on power production forecasts for wind and solar energy. Chapter 3 defines our research problem. The objective is to schedule the maintenance operations in order to maximize the revenue generated by the total power production of the wind turbines while taking into account wind predictions, multiple task execution modes, task assignment constraints to a multi-skilled workforce, and daily restrictions on the routes of the technicians. Since wind predictions can only be reliably established few days in advance, the work thus targets a short-term horizon that contains multiple days. We point out that reading Chapter 3 is an absolute prerequisite before reading Chapters 4 to 8. At a first step, we tackle the deterministic problem assuming perfect knowledge of the wind speed during the whole time horizon. Chapter 4 proposes different integer linear programming (ILP) formulations of our problem. We then explore two different approaches to tackle it. Chapter 5 provides a heuristic solution method, namely a constraint programming-based large neighborhood search (CPLNS) approach. Chapter 6 investigates the possibility of designing an efficient exact approach to tackle the problem. Based on a decomposition of the problem, we present a branch-and-check (B&C) approach. When the length of the time horizon increase or when good quality forecasts are unavailable, Chapter 7 studies the problem the objective of maximizing the revenue is substituted by the objective of maximizing the availability of the turbines. Finally, Chapter 8 aims to take into account the uncertain nature of the wind speed in the decision making process. We introduce a robust approach taking into consideration the potential spatial and time-wise correlation of the wind speed.

Notice that all the main notations used throughout this part are summarized in Appendix A.

Chapter 3

Problem description

3.1 Context

As the energy sector is facing major challenges to produce low-carbon power or carbon-free electricity, the share of renewable energies has significantly increased in recent years. Boosted by climate change mitigation and adaptation efforts (e.g., tax incentives and Paris climate change agreement) and the constantly-decreasing cost of turbines, wind energy is currently the world's fastest-growing source of electricity (63 GW of new wind power capacity in 2015), accounting nowadays for around 3.3% of the world electricity production (The Global Wind Energy Council, 2016). Although wind turbines AF tops 95%, their CF is usually around 25-30% as a result of the intermittency of the wind and of design decisions (for a fixed wind speed, the larger are the blades the more electricity the turbine can produce). The impact of operational decisions on this value is also non negligible. As the wind industry is steadily growing, reliability and profitability of wind farms naturally becomes one of the priorities of the sector. In this context, developing optimization techniques to efficiently schedule wind turbine maintenance operations is essential to prevent unnecessary downtimes and excessive operational costs.

Maintenance planning and scheduling has been widely studied in different industrial contexts (see for example Budai et al. (2008) for a survey). In general, however, solutions remain sector-specific. In the particular case of the maintenance of the generating units, Chapter 1 shows that the problem is concerned with the definition of time intervals for preventive maintenance of generating units under financial (cost minimization, profits maximization) and/or reliability (leveling, maximization of the net reserves) considerations. Unfortunately, the studies focus essentially on the traditional electricity generating technologies (nuclear, fossil fuels). The approaches are thus inapplicable to the wind power industry. One of the main reasons is that wind farms are usually owned by investment funds, and the operation and the maintenance of the turbines are often outsourced to a third party. As it stands, the stakeholders and the contractors face potentially conflicting objectives: maximize electricity production vs. minimize maintenance costs. Therefore, service contracts are set between these two entities. These contracts essentially fix a target value for the annual availability of the wind farms to produce electricity and fix the timely preventive maintenance operations (e.g., preventive maintenance on a wind farm should be performed every 6 months during the first two years and then every year). This choice is partially motivated by the fact that this value is relatively easy to compute. However, since maximizing AF is not equivalent to maximizing CF, the stakeholders retain the right to question the maintenance plan if it leads to large losses of production. To reduce such interference, service contracts include financial incentives. Indeed, maintenance companies can sometimes expect bonuses if the availability of the wind farm is larger than a certain percentage (e.g., 97%). Moreover, they may obtain a share of the revenue on every percentage of availability larger than a specified value. Some contracts also explicitly mention production targets, but the values are very easily reachable as wind faces volatility (there are more windy years than others). Penalties also exist if the goals related to the AF and/or CF are not met. From a field investigation and from discussions with WPred, it results that maintenance companies are not always taking into account production concerns. Usually, it is essentially done in order to limit the interference with the wind farms owners and/or to avoid potential penalties if they are below the targets or if they feel that some bonuses are reachable. Another specificity of the wind power industry is that maintenance decisions are not correlated with the electricity demand since producers are mostly not required to satisfy production goals fixed in advance. The objective then tends to be the maximization of the efficiency of the wind turbines. Last but not least, the wind power production is inherently volatile, and the meteorological conditions have a great impact on the maintenance plan and can induce last-minute adjustments. In summary, the aim of maintenance companies is to schedule the preventive maintenance in order to meet their contract commitments. Although it is not their top priority, producing maintenance plans for which the production of the turbines is maximized, while taking into consideration their internal constraints, is a meaningful strategy to avoid interference with the stakeholders and to potentially increase their revenue. If the maintenance is not outsourced, this objective is all the more relevant.

Maintenance optimization for wind turbines has only recently started to receive attention in the engineering and reliability literature (once gain, we refer the reader to Ding et al. (2013) for a survey). This stream of research primarily focuses on the definition of maintenance policies according to failure models or/and condition monitoring. The results of the optimization process is a maintenance time interval for each turbine. Although existing studies precisely define time intervals during which the maintenance has to be performed in order to reduce the loss of electricity production, they usually focus on a single wind farm or wind turbine and do not consider a fine-grained resource management. Therefore, the obtained results are used more as guidelines than as an actual maintenance schedule. In this regard, they can be used to set the service contracts.

Fine-grained resource management implies, among others, considering a multi-skilled workforce, coping with individual or global resource unavailability time periods (e.g., vacations), and taking into account resource location-based constraints. Dealing with these issues requires considering a short-term planning horizon. In this context, existing studies do not deliver a solution to the problem but do provide valuable input. Indeed, they allow planners to define the tasks to be performed during the planning horizon and to set the maintenance time window constraints. Considering fine-grained resource management then results in detailed maintenance plans that can be used on a daily or weekly basis. These latter provide more accurate estimates of turbine downtimes and loss of production; two metrics that can otherwise be underestimated, which may lead to significant prediction errors. Indeed, producing a maintenance plan in which no operations generate a loss of production (e.g., is scheduled during time periods where the wind speed is below 3.5 $m.s^{-1}$, which is too low to produce electricity) can almost never be achieved in practice, since human resources are a major bottleneck.

Only few studies included a fine-grained resource management when scheduling the maintenance of wind turbines. For the particular case of offshore wind farms, Irawan et al. (2016) optimized a maintenance routing and scheduling problem minimizing labor, travel, and penalty costs. They proposed a solution method based on Dantzig-Wolfe decomposition in which all the feasible routes for each vessel are generated a priori. If one considers the onshore industry, the problem to solve is very different. Kovács et al. (2011) considered the scheduling of the maintenance operations on a one-day time horizon. These authors aimed to minimize lost production due to maintenance and failures. They introduced incompatibilities between pairs of tasks and managed the assignment of teams of multi-skilled workers to tasks. They modeled the problem as an integer linear program and solved it with a commercial solver. They performed experiments on instances with up to 50 tasks.

We formally introduce our research problem in the next section. Notice beforehand that it differs from that introduced by Kovács et al. (2011) in several ways. First, we consider an individual management of the technicians through a space-time tracking taking into consideration availability calendars. Second, we consider multiple task execution modes that impact the duration of tasks as well as the resource requirements (De Reyck et al., 1998). Third, we present an alternative way to consider the travel times of the technicians by imposing restrictions on their routes.

3.2 Problem statement

The aim of the problem we consider is to schedule a set $\mathcal{I}$ of maintenance tasks during a discrete and finite planning horizon $\mathcal{T}$ while maximizing the revenue generated by the wind electricity production of a set $\mathcal{W}$ of wind turbines. The wind turbines are geographically spread across a set of locations $\mathcal{L}$. We denote $l_w \in \mathcal{L}$ the location of wind turbine $w \in \mathcal{W}$ and l_i the location where task $i \in \mathcal{I}$ has to be performed.

The time horizon is a totally ordered set partitioned into $|\mathcal{T}|$ time periods of identical length. $\mathcal{T}$ spans over several days from a set $\mathcal{D}$. We denote $\mathcal{T}_d$ the time periods that belongs to day $d \in \mathcal{D}$. Moreover, since the execution of a task can impact the production during non-working hours, we introduce a special time period (hereafter referred to as a *rest time period*) between two consecutive days to represent, for example, a night or a weekend. Maintenance tasks are non-preemptive (in the sense that a task cannot be interrupted by another task and resumed after the completion of the latter), but, obviously, they are interrupted during rest time periods if they overlap different consecutive days (e.g., a technician can start a task at the end of one day and complete it at the beginning of the next day).

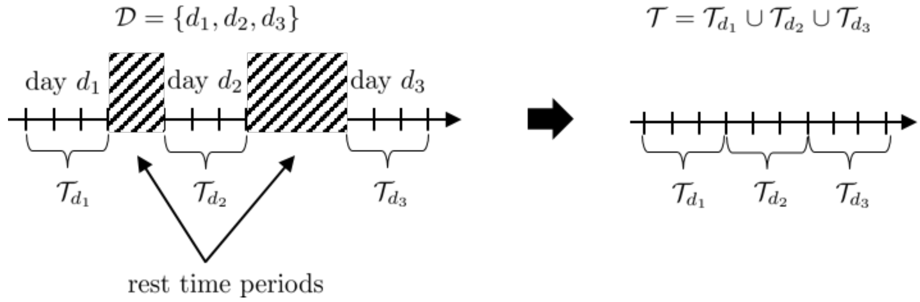


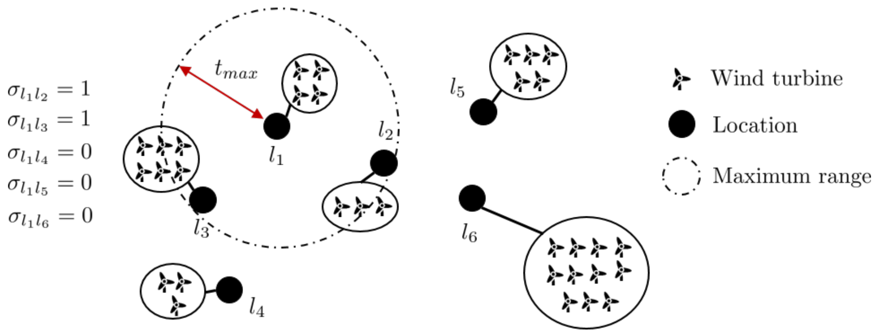
Figure 3.1 – Illusration of the construction of the time horizon $\mathcal{T}$ and the set of days $\mathcal{D}$

Although we do not include rest time periods in $\mathcal{T}$ (see Figure 3.1), we count in the objective function the loss of revenue generated by tasks overlapping these specific time periods. Tasks may have different impacts on the availability of the turbines. Some tasks shut down one (or more) turbine(s) since the task starts until the task ends. For instance, during the maintenance of a wind farm's substation[1] no turbines in the farm can produce electricity. It should be noted, however, that tasks that shut down more than one turbine are very rare in practice. Some tasks shut down the turbines when the technicians are effectively working on the task but not necessarily during the rest time periods they overlap. This is the case for the majority of the preventive maintenance operations, as well as for wind turbines retrofit. Other tasks do not have any impact on the electricity production (e.g., usually wind farm inspections). We model the impact of the tasks on electricity production using two parameters. First, binary parameter b_{wi} takes the value 1 if and only if task $i \in \mathcal{I}$ shuts down turbine $w \in \mathcal{W}$ when technicians are effectively working on the task. Second, binary parameter $\widetilde{b}_{wi}$ takes the value 1 if and only if task i additionally shuts down turbine w during the rest time periods it overlaps. Notice that parameters b_{wi} and $\widetilde{b}_{wi}$ are equal to 0 if turbine w is not located at the location where the task i has to be performed (i.e., if $l_i \neq l_w$).

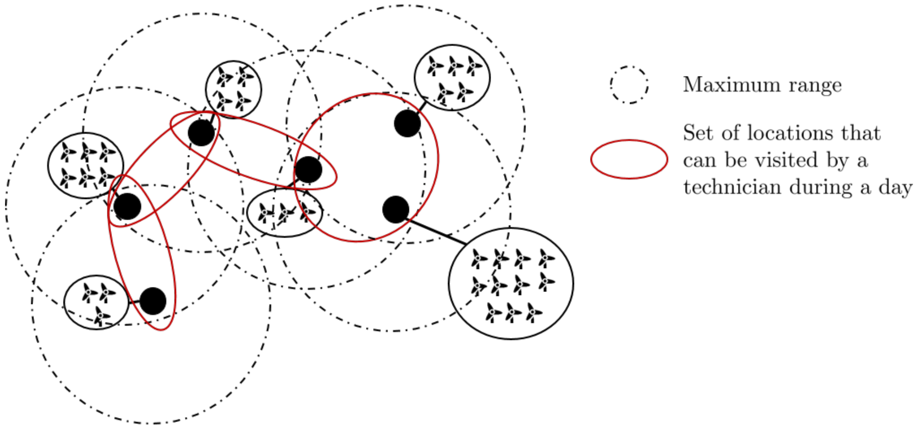
To execute the maintenance tasks, we have a finite set $\mathcal{R}$ of technicians. Each technician masters one or multiple skills from a set $\mathcal{S}$. We express technician skills by a binary vector ζ_r over $\mathcal{S}$ such that $\zeta_{rs} = 1$ if and only if technician r masters skill $s \in \mathcal{S}$. In our problem, each task $i \in \mathcal{I}$ requires a specific skill $s_i \in \mathcal{S}$. Our work on the field revealed that this is a reasonable assumption in our context. Indeed, some companies prefer to assign, say, two interchangeable technicians to the same maintenance operations because they can both take care of any part of the task. For convenience, we define as $\mathcal{R}_i$ the set of technicians that can perform task i (i.e., $\mathcal{R}_i = \{r \in \mathcal{R} | \zeta_{rs_i} = 1\}$).

1. A wind farm substation collects the electricity produced by all the turbines of the farm and distributes it through the grid

To avoid expensive travel times and save valuable time, we constraint technicians to work during a single day on tasks located within reduced geographic regions. To this effect, we define a binary parameter $\sigma_{ll'}$ taking value of 1 if and only if a technician is allowed to work at both locations $l, l' \in \mathcal{L}$ during the same day (naturally $\sigma_{ll'} = \sigma_{l'l}$). Let us assume that t_{max} is the maximum travel time between two locations that we can consider "negligible" with respect to the duration of a time period. The value of parameter $\sigma_{ll'}$ is then equal to 1 if and only if the travel time between l and l' is less than or equal to t_{max}. Figure 3.2a illustrates how these parameters are set. During a single day, a technician is allowed to work at a set of locations if and only if for every couple (l, l') of locations in this set we have $\sigma_{ll'} = 1$ (see Figure 3.2b). For instance, one should observe that, during a day, a technician can only execute tasks at l_1 and l_2 or l_3 but not both. We refer to these restrictions as *daily location-based incompatibilities*. It is worth mentioning that wind turbine maintenance tasks usually span along hours (if not days), and therefore technicians tend to travel between very few locations during a single working day.



(a) Setting of the parameters $\{\sigma_{l_1l'}\}_{l' \in \mathcal{L}}$



(b) Definition of the maximum sets of locations on which a technician is allowed to work during a day

Figure 3.2 – Illustration of the daily location-based incompatibilities

Each technician $r \in \mathcal{R}$ has also an individual availability schedule expressed by a binary vector ρ_r, with $\rho_r^t = 1$ if and only if r is available during time period $t \in \mathcal{T}$. The availability schedule of every technician is related to training times, personal holiday times, and assignments to tasks (not part of the optimization) that have been already started or that are performed along with external companies. When a technician r is not available during a time period t, his or her location is fixed to $l_r^t \in \mathcal{L}$. Notice that for technician personal holidays and training sessions, this parameter is set to a dummy location l^* such that $\forall l \in \mathcal{L}, \sigma_{l^*l} = 1$. We assume that all the technicians work the same shift, which is a common practice according to our industrial partners.

Multiple execution modes are available for each task. For each task $i \in \mathcal{I}$, we denote as $\mathcal{M}_i$ its set of execution modes. For each mode $m \in \mathcal{M}_i$, there are an associated task duration d_{im} and a number q_{im} of required technicians. It is noteworthy that switching modes after starting the execution of a task is forbidden. We also consider that a technician cannot perform more than one task during a given time period. Moreover, an important feature of the problem is that a technician assigned to a task has to work on it from the beginning to the end, even if the task overlaps one or multiple rest time periods.

Tasks can only be executed during some specific time periods. These take into account spare parts availability and external restrictions imposed by the operator and/or the stakeholders. These restrictions are included in the definition of parameter ϑ_i^t which is set equal to 1 if and only if task $i \in \mathcal{I}$ can be executed during time period $t \in \mathcal{T}$. Safety work conditions (e.g., a technician cannot perform certain tasks on a turbine when the wind is too strong) need to be included as well. To this end, we introduce parameter $\ddot{\vartheta}_i^t$ which is set equal to 1 if and only if task i can be executed during time period t according to the wind speed. Additionally, some subsets of tasks cannot overlap due, for instance, to the use of disjunctive resources, an interference (e.g., two tasks cannot be executed on the same turbine at the same time), or managerial preferences. We define $ov(\mathcal{I})$ the set containing all subsets of tasks that should not overlap.

The objective of the problem is to determine a schedule that maximizes the revenue generated by the electricity production of the wind farms while meeting the constraints previously described. The electricity production depends on the wind speed. It also relies on the selling price of wind electricity, the nominal power and the capacity factor curve (see Figure 3.3) associated with the wind turbines. We denote as g_w^t the value of the revenue generated by wind turbine $w \in \mathcal{W}$ if it can produce electricity during time period $t \in \mathcal{T}$. Similarly, we denote as $\tilde{g}_w^d$ the revenue generated by wind turbine w if it can produce electricity during the rest time period following day $d \in \mathcal{D}$. The revenue is estimated according to the forecasted wind speed. These values are used in Chapters 4, 5, and 6, while the uncertainty on the wind speed predictions is tackled in Chapter 8.

In this study, we do not consider maintenance costs: we assume that, as it is common in practice, technicians earn a fixed salary, and we disregard travel costs as they are irrelevant to the decision-making process.

One particularity of the problem is the possibility to postpone the scheduling of some tasks until the next planning horizon. To model the postponement of task $i \in \mathcal{I}$, we create an additional execution mode m_i^0 and we add it to $\mathcal{M}_i$ (we have $q_{im_i^0} = 0$ and $d_{im_i^0} = 0$). When task i is postponed, we apply to the objective a penalty of $o_i \geq 0$. The value of this penalty is fixed according to the relative degree of priority of the task. This priority depends on reliability consideration (the more a maintenance operation is delayed, the higher is the probability of failure) and contract commitments. If a task is postponed, it obviously does not impact the production of any wind turbines, and thus the value of the revenue. Therefore, if a task needs to be scheduled during the time horizon, this penalty has to be fixed in connection to the revenue in order to ensure that the postponement of this task is non-profitable. This penalty includes an estimation of the loss of revenue induced by the schedule of the corresponding task, to which may be added outsourcing costs (the decision maker then being responsible for the choice of outsourcing of task rather than postponing it). Since the time horizon contains a finite number of time periods and the number of technicians is limited, we cannot ensure scheduling all the maintenance tasks. So, if the penalties are high enough, postponing a task is just triggered to overcome a possible lack of technicians. In this case, the problem is a lexicographic optimization problem where maximizing the number of tasks is the primary objective and maximizing the revenue generated by the electricity production of the turbines is the secondary objective. Combining these two objectives into one single-objective scalar function (often known as *scalarization technique* and in that case *weighted-sum technique*) enables to solve this multiobjective problem. In summary, regardless of the intended purpose, the objective function to be maximized in the problem always corresponds to the difference between the revenue and the postponing penalties.

Finally, we use the following terminology to refer to some critical wind speed. The *cut-in* speed value [WCI] corresponds to the wind speed – around 3.5 $m.s^{-1}$ – below which the capacity factor is equal to 0 (i.e., the wind turbine cannot produce electricity). On the opposite side, at *cut-out* speed [WCO], the turbine is forced to stop because there is a risk of damage to the rotor. We also introduce the *rated output* speed [WR] – around 14 $m.s^{-1}$ – from which the capacity factor is equal to 1 (i.e, the turbine generates at full capacity until the wind reaches the cut-out speed). The CF function is a con-

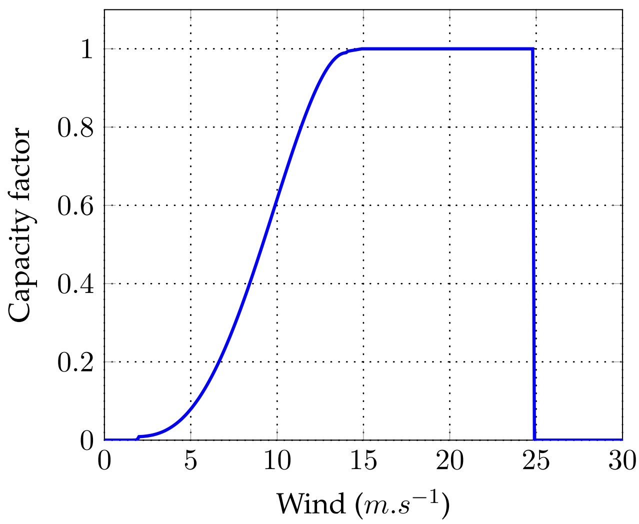


Figure 3.3 – Capacity factor of a wind turbine according to the wind speed

tinuous non-decreasing function on the interval $[0, WCO[$. The point WCO is a jump discontinuity of the function where the value of this latter drops from 1 to 0.

3.3 Complexity

It is rather direct to note that the wind turbine maintenance scheduling problem (WTMSP) described in the previous section includes various central scheduling problems as particular cases. Before analyzing more formally its complexity, we define the decision problem WTMSP^{dec} associated with WTMSP. In this problem, a parameter $G \in \mathbb{R}$ is given as a lower bound on the value of the objective function (computed as the difference between the revenue and the penalties incurred due to postponed tasks). The problem WTMSP^{dec} is to decide whether there exists a schedule of the tasks such that the objective value is greater than or equal to G. We propose in this section a polynomial reduction of the *cumulative scheduling problem* (CuSP) – known to be NP-complete in the strong sense (Baptiste et al., 1999) – to WTMSP^{dec} (see Proposition 3.3.1).

Let us recall that a CuSP instance consists of a single resource with a given capacity C and a set $\mathcal{J}$ of n jobs where each job $j \in \mathcal{J}$ has a release date r_j, a deadline d_j, a processing time p_j and a capacity resource requirement a_j. The problem is to decide whether there exists a schedule of all the jobs satisfying the timing and the resource capacity constraints.

Proposition 3.3.1. *WTMSP^{dec} is NP-complete in the strong sense.*

Proof. First, let us prove that CuSP polynomially reduces to WTMSP^{dec}. From an instance of CuSP, we build an instance of WTMSP^{dec} by setting $\mathcal{T} = \{\min_{j\in\mathcal{J}} r_j, ..., \max_{j\in\mathcal{J}} d_j\}$[2], $|\mathcal{D}| = 1$, $\mathcal{L} = \{l\}$, $\mathcal{S} = \{s\}$, $|\mathcal{R}| = C$, $|\mathcal{W}| = 1$, $|\mathcal{I}| = n$. For each job $j \in \mathcal{J}$, we create a task i such that $l_i = l$, $s_i = s$, $\mathcal{M}_i = 1$, $d_{i1} = p_j$ and $q_{i1} = a_j$. We set $\ddot{\vartheta}_i^t = 1$ for every time period $t \in \mathcal{T}$, and set $\vartheta_i^t = 1$ if $r_j \leq t < d_j$ and $\vartheta_i^t = 0$ otherwise. For every technician $r \in \mathcal{R}$, we set $\rho_r^t = 1$ for every time period t and $\zeta_{rs} = 1$. For turbine $w \in \mathcal{W}$, we fix $b_{wi} = 0$ and $\widetilde{b}_{wi} = 0$ for each task $i \in \mathcal{I}$, $g_w^t = 0$ for each time period t, and $g_w^d = 0$ for each day $d \in \mathcal{D}$. The penalty incurred when a task i is postponed is fixed to $c_i = 1$. We finally set $G = 0$. For the resulting instance, if o_i takes value 1 when task i is postponed, the objective to maximize is $-\sum_{i\in\mathcal{I}} c_i o_i$. In summary, we reduce CuSP to a slightly relaxed version of WTMSP^{dec} where: i) the tasks have no impact on the availability of the turbines, ii) there exists only one execution mode for each task, iii) there exists only 1 skill, iv) the technicians do not have any availability schedule and can be assigned to any task, v) there are no daily location-based incompatibilities, and vi) the aim of the problem is to maximize the number of tasks scheduled during the planning horizon.

We now prove that there exists a solution to CuSP if and only if WTMSP has a solution with an objective value greater than 0.

2. For two integers a_1 and a_2, symbol $\{a_1, ..., a_2\}$ refers to $[a_1; a_2] \cap \mathbb{Z}$ where $\mathbb{Z}$ is the set of integers.

Conversely, let us assume that there exists a solution to CuSP. Each job $j \in \mathcal{J}$ has then been assigned a starting time S_j such that the timing and the resource capacity constraints are satisfied. For each task i we set its starting time to the starting time S_j of its associated job j. Since $r_j \leq S_j$ and $r_j + p_j \leq d_j$, the task i is scheduled during d_{i1} consecutive time periods t such that $\vartheta_i^t = 1$ and $\ddot{\vartheta}_i^t = 1$. Then, we iterate over the time horizon while keeping a pool $\hat{\mathcal{R}}$ of available technicians (initialized to $\mathcal{R}$). More specifically, at each time period t, we operate as follows: (1) for each task i starting at the beginning of time period t we remove q_{i1} technicians from the set $\hat{\mathcal{R}}$ and we assign them to i (2) for each task i ending at the end of time period t, we add the technicians previously assigned to i to the set $\hat{\mathcal{R}}$. Since during each time period the jobs require less than or exactly C resources, we are ensured that there are always enough technicians to be assigned to the tasks. Lastly, we end up this construction scheme with no tasks set to be postponed. The objective value of the resulting solution is therefore equal to 0.

Let us assume that there exists a solution to WTMSP with $G = 0$. To each job $j \in \mathcal{J}$, we assign the same starting time as the associated task in WTMSP. From the definition of the problem WTMSP, we are ensured that during each time period the operations require less than or exactly C resources and that the timing constraints are satisfied. The resulting schedule is therefore a solution to CuSP.

We can now state that CuSP polynomially reduces to WTMSP^{dec}. Moreover, WTMSP^{dec} is obviously in NP, as it is easy to see that a solution can be checked in polynomial time. Since CuSP is NP-complete in the strong sense, the same holds for WTMSP^{dec}. □

Since the decision problem WTMSP^{dec} is NP-complete in the strong sense, WTMSP is therefore strongly NP-hard. Notice that in some particular cases solving WTMSP becomes trivial. Indeed, if the postponement penalties are all equal to 0, the solution consisting in delaying all the tasks is optimal. A rather similar observation is that each task having a postponement penalty equal to 0 can be set to be delayed, without affecting the value of the optimal solution, prior to the optimization.

3.4 Instance generation

In order to assess the quality of our models and optimization algorithms for the problem, we needed data. Unfortunately, despite our close collaboration with WPred and their best efforts to help us accessing data we were unable to obtain reliable real datasets. One of the main obstacles we faced was their customers' (i.e., the maintenance companies) lack of proper information systems. For instance, when we could get access to the location of the maintenance task, we could not trace back their duration. To overcome these difficulties, we put together the insight on wind prediction and maintenance operations that we obtained from WPred and their customers to build an instance generator. Although devising a perfect instance generator is nothing but impossible, we believe our instances represent reality with a good degree of accuracy. For a thorough discussion on the instance generation process, the reader is referred to Appendix B.

We created one testbed, denoted G1, to test the efficiency of our different solutions methods to solve the problem defined in this chapter. We considered time horizons of different lengths (5 days and 10 days with 2 or 4 time periods per day), different number of tasks (20, 40, 80), and different number of skills (1 or 3). Each task can be executed in several modes (1 to 3). For each combination of parameters, we generated two categories of instances: 5 instances with a tight technicians-to-work ratio (i.e., technicians can perform the majority of the tasks, but they are not guaranteed to be enough to perform all the tasks), and 5 instances with a regular technicians-to-work ratio (i.e., we are sure that all the tasks can be executed by the technicians). We refer to the former as type A and to the latter as type B. We obtained 160 instances divided into 32 families of instances (see Table 3.1) as representative as possible of the wide range of situations which may occur.

Table 3.1 – Families of instances in testbed G1

Family	$\|\mathcal{T}\|$	$\|\mathcal{D}\|$	$\|\mathcal{S}\|$	$\|\mathcal{I}\|$	Type	# instances
10_2_1_20_A	10	2	1	20	A	5
10_2_1_20_B	10	2	1	20	B	5
10_2_3_20_A	10	2	3	20	A	5
10_2_3_20_B	10	2	3	20	B	5
10_2_1_40_A	10	2	1	40	A	5
10_2_1_40_B	10	2	1	40	B	5
10_2_3_40_A	10	2	3	40	A	5
10_2_3_40_B	10	2	3	40	B	5
20_4_1_20_A	20	4	1	20	A	5
20_4_1_20_B	20	4	1	20	B	5
20_4_3_20_A	20	4	3	20	A	5
20_4_3_20_B	20	4	3	20	B	5
20_4_1_40_A	20	4	1	40	A	5
20_4_1_40_B	20	4	1	40	B	5
20_4_3_40_A	20	4	3	40	A	5
20_4_3_40_B	20	4	3	40	B	5
20_2_1_40_A	20	2	1	40	A	5
20_2_1_40_B	20	2	1	40	B	5
20_2_3_40_A	20	2	3	40	A	5
20_2_3_40_B	20	2	3	40	B	5
20_2_1_80_A	20	2	1	80	A	5
20_2_1_80_B	20	2	1	80	B	5
20_2_3_80_A	20	2	3	80	A	5
20_2_3_80_B	20	2	3	80	B	5
40_4_1_40_A	40	4	1	40	A	5
40_4_1_40_B	40	4	1	40	B	5
40_4_3_40_A	40	4	3	40	A	5
40_4_3_40_B	40	4	3	40	B	5
40_4_1_80_A	40	4	1	80	A	5
40_4_1_80_B	40	4	1	80	B	5
40_4_3_80_A	40	4	3	80	A	5
40_4_3_80_B	40	4	3	80	B	5

Chapter 4

Formalization

This chapter proposes different integer linear programming (ILP) formulations of the problem defined in Chapter 3. These formulations differ by the definition of the variables and the modeling of the different constraints. They also provide an alternative way (using mathematical notations) to understand the problem. Section 4.1 presents two natural formulations for the problem. Section 4.2 then introduces two compact formulations (the term "compact" is coming from the fact that they have a reduced number of constraints) better suited to run on commercial solvers. Finally, Section 4.3 reports computational experiments carried on the four proposed formulations.

Remark: To avoid introducing a large amount of notation, we use the same letter to represent decision variables that are part of different formulations (in that case, we aim to use the same letter for variables sharing the closest definition).

4.1 Natural ILP formulations

Let us introduce the following decision variables:

$$x_{im} = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ is executed in mode } m \in \mathcal{M}_i, \\ 0 & \text{otherwise.} \end{cases}$$

$$s_i^t = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ starts at the beginning of time period } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ri} = \begin{cases} 1 & \text{if technician } r \in \mathcal{R} \text{ is assigned to task } i \in \mathcal{I}, \\ 0 & \text{otherwise.} \end{cases}$$

$$c_i^t = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ ends at the end of time period } t-1 \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

$$e_i^t = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ is executed during time period } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

$$u_i^d = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ is executed during day } d \in \mathcal{D}, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_w^t = \begin{cases} 1 & \text{if turbine } w \in \mathcal{W} \text{ can produce electricity during time period } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\widetilde{f}_w^d = \begin{cases} 1 & \text{if turbine } w \in \mathcal{W} \text{ can produce electricity during the rest time period following day } d \in \mathcal{D}, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ri}^t = \begin{cases} 1 & \text{if technician } r \in \mathcal{R} \text{ is assigned to task } i \in \mathcal{I} \text{ during time period } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

$$v_{rl}^t = \begin{cases} 1 & \text{if technician } r \in \mathcal{R} \text{ is at location } l \in \mathcal{L} \text{ during time period } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

A first intuitive formulation is defined as the following integer linear program $[P1]$:

$$[P1] \;\; \max \sum_{w\in\mathcal{W}} \left(\sum_{t\in\mathcal{T}} g_w^t f_w^t + \sum_{d\in\mathcal{D}} \widetilde{g}_w^d \widetilde{f}_w^d \right) - \sum_{i\in\mathcal{I}} o_i x_{im_i^0} \tag{4.1}$$

subject to:

$$\sum_{m\in\mathcal{M}_i} x_{im} = 1 \;\; \forall i \in \mathcal{I}, \tag{4.2}$$

$$e_i^0 = 0 \;\; \forall i \in \mathcal{I}, \tag{4.3}$$

$$e_i^t = e_i^{t-1} + s_i^t - c_i^t \;\; \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.4}$$

$$\sum_{t\in\mathcal{T}} s_i^t = 1 \;\; \forall i \in \mathcal{I}, \tag{4.5}$$

$$\sum_{t\in\mathcal{T}} c_i^t = 1 \;\; \forall i \in \mathcal{I}, \tag{4.6}$$

$$e_i^t \le \vartheta_i^t \ddot{\vartheta}_i^t \;\; \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.7}$$

$$\sum_{i\in\mathcal{B}} e_i^t \le 1 \;\; \forall \mathcal{B} \in ov(\mathcal{I}), \forall t \in \mathcal{T}, \tag{4.8}$$

$$\sum_{t\in\mathcal{T}_d} e_i^t \le |\mathcal{T}_d| u_i^d \;\; \forall i \in \mathcal{I}, \forall d \in \mathcal{D}, \tag{4.9}$$

$$f_w^t + b_{wi} e_i^t \le 1 \;\; \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.10}$$

$$\widetilde{f}_w^d + \widetilde{b}_{wi} \left(u_i^d + u_i^{d+1} \right) \le 2 \;\; \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall d \in \mathcal{D}, \tag{4.11}$$

$$\sum_{t\in\mathcal{T}} e_i^t = \sum_{m\in\mathcal{M}_i} d_{im} x_{im} \;\; \forall i \in \mathcal{I}, \tag{4.12}$$

$$\sum_{r\in\mathcal{R}_i} y_{ri} = \sum_{m\in\mathcal{M}_i} q_{im} x_{im} \;\; \forall i \in \mathcal{I}, \tag{4.13}$$

$$e_i^t + y_{ri} - z_{ri}^t \le 1 \;\; \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i, \forall t \in \mathcal{T}, \tag{4.14}$$

$$z_{ri}^t \le y_{ri} \;\; \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i, \forall t \in \mathcal{T}, \tag{4.15}$$

$$z_{ri}^t \le e_i^t \;\; \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i, \forall t \in \mathcal{T}, \tag{4.16}$$

$$\sum_{i\in\mathcal{I}_l\cap\mathcal{R}_i} z_{ri}^t \le \rho_r^t v_{rl}^t \;\; \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}, \tag{4.17}$$

$$\sum_{l\in\mathcal{L}} v_{rl}^t = 1 \;\; \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \tag{4.18}$$

$$v_{rl_r^t}^t = 1 \;\; \forall r \in \mathcal{R}, \forall t \in \mathcal{T} \; s.t. \; \rho_r^t = 0, \tag{4.19}$$

$$v_{rl}^t + \sum_{l'\in\mathcal{L}|\sigma_{ll'}=0} v_{rl'}^{t'} \le 1$$

$$\forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \forall (t,t') \in \mathcal{T}_d \times \mathcal{T}_d, t \ne t', \forall l \in \mathcal{L}, \tag{4.20}$$

$$e_i^t \in \{0,1\} \;\; \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \cup \{0\}, \tag{4.21}$$

$$s_i^t, c_i^t \in \{0,1\} \;\; \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.22}$$

$$u_i^d \in \{0,1\} \;\; \forall i \in \mathcal{I}, \forall d \in \mathcal{D}, \tag{4.23}$$

$$f_w^t \in \{0,1\} \;\; \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \tag{4.24}$$

$$\widetilde{f}_w^d \in \{0,1\} \;\; \forall w \in \mathcal{W}, \forall d \in \mathcal{D}, \tag{4.25}$$

$$y_{ri} \in \{0,1\} \;\; \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i, \tag{4.26}$$

$$z_{ri}^t \in \{0,1\} \;\; \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i, \forall t \in \mathcal{T}, \tag{4.27}$$

$$v_{rl}^t \in \{0,1\} \;\; \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}. \tag{4.28}$$

The objective in (4.1) is defined as the difference between the revenue generated by the electricity production from the wind turbines and the penalties induced by the postponement of tasks. Constraints (4.2) guarantee that exactly one execution mode is selected for each task. For each task, constraints (4.3)-(4.6) ensure consistency between the starting time, ending time, and execution time period variables. Constraints (4.7) prevent a task to be scheduled during forbidden time periods. Constraints (4.8) are the non-overlapping constraints. Constraints (4.9) couple the time periods during which each task is performed to its execution days. Constraints (4.10) and (4.11) compute the impact of the tasks on the availability of the turbines to produce electricity. Constraints (4.12) connect the duration of each task to its selected execution mode. Constraints (4.13) ensure that the technician requirements are fulfilled. Constraints (4.14) force technicians to be assigned to a task from its beginning to its end. For each technician, constraints (4.15)-(4.16) ensure consistency between the global assignment and the time-indexed assignment variables. Constraints (4.17) couple the location of the technicians to the tasks they perform. Constraints (4.18) prevent technicians to perform multiple tasks during the same time period. When technicians are not available, constraints (4.19) ensure compliance with their known locations. Constraints (4.20) define the daily location-based incompatibilities for each technician. Finally, (4.21)-(4.28) state the binary nature of the decision variables.

As an alternative to avoid the use of the variables c_i^t and the constraints (4.3)-(4.6), one can redefine the variables s_i^t as follows:

$$s_{im}^t = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ starts at the beginning of time period } t \in \mathcal{T} \text{ and is executed in mode } m \in \mathcal{M}_i \setminus \{m_i^0\}, \\ 0 & \text{otherwise.} \end{cases}$$

This alternative formulation – denoted [P1bis] – reads:

$$[P1bis] \quad \max \sum_{w \in \mathcal{W}} \left(\sum_{t \in \mathcal{T}} g_w^t f_w^t + \sum_{d \in \mathcal{D}} \tilde{g}_w^d \tilde{f}_w^d \right) - \sum_{i \in \mathcal{I}} o_i x_{im_i^0}$$

subject to:

(4.2), (4.7) – (4.20)

$$x_{im} = \sum_{t \in \mathcal{T}} s_{im}^t \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}_i \setminus \{m_i^0\}, \tag{4.29}$$

$$e_i^t = \sum_{m \in \mathcal{M}_i \setminus \{m_i^0\}} \sum_{t'=t-d_{im}+1}^{t'=t} s_{im}^t \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.30}$$

$$s_{im}^t \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \forall m \in \mathcal{M}_i \setminus \{m_i^0\} \tag{4.31}$$

(4.21), (4.23) – (4.28)

Constraints (4.29) and (4.30) couple the selected executing mode and the execution time period variables to the newly created variables. Note that the constraints (4.12) are not useful anymore, but it may improve the efficiency of an ILP solver in case of a direct resolution of $[P1bis]$.

4.2 Compact ILP formulations

In order to restrict the number of constraints involved in the formulations $[P1]$ and $[P1bis]$, we built two more compact formulations. The models are based on the concept of *plans*. A plan associated with task $i \in \mathcal{I}$ defines a feasible schedule for i by setting an execution mode, a consistent starting time period, and, by induction, a duration and a resource requirement. For example, consider a task i with two execution modes m_1 and m_2. Let d_{m_1} and d_{m_2} denote the corresponding duration and q_{m_1} and q_{m_2} the corresponding number of required technicians. Assume that task i can be executed during the whole time horizon. For each time period $t \in \mathcal{T}$ such that $t \leq |T| - d_{m_1}$, we create a feasible plan to represent the planning of task i within mode m_1 from period t to period $t + d_{m_1}$ with a requirement of q_{m_1} technicians. The same procedure is applied for mode m_2. Obviously, we take into consideration the impossibility of preempting tasks when building plans.

All the plans are generated a priori. Since the number of time periods is assumed to be short and there are only a few execution modes, the total number of plans is not so large. We denote by $\mathcal{P}$ the set of plans, i_p the task involved in plan $p \in \mathcal{P}$, and $\mathcal{P}_i$ the set of all plans involving task i (i.e.,

$\mathcal{P}_i = \{p \in \mathcal{P} | i_p = i\}$). For each task i, we also create a plan $p_i^0 \in \mathcal{P}_i$ representing the postponement of the task. For a plan p, we express the execution time periods of i_p by a boolean vector a_p where $a_p^t = 1$ if and only if i_p is executed during time period $t \in \mathcal{T}$. We also denote S_p and C_p the starting and completion time periods of plan p (i.e., $S_p = \min_{t\in\mathcal{T}} a_p^t t$ and $C_p = \max_{t\in\mathcal{T}} a_p^t t$). Similarly, we introduce another binary vector $\widetilde{a}_p$ over $\mathcal{D}$ such that $\widetilde{a}_p^d = 1$ if and only if i_p overlaps the rest time period following day $d \in \mathcal{D}$. We also denote $\mathcal{D}_p$ as the set of days overlapped by the plan p. For convenience and with a slight abuse of notation, we introduce parameters s_p, l_p, b_{wp}, and $\widetilde{b}_{wp}$ equal to s_{i_p}, l_{i_p}, b_{wi_p}, and $\widetilde{b}_{wi_p}$ respectively. Moreover, we denote $\mathcal{R}_p$ the set of technicians that can be assigned to plan p. More specifically, $\mathcal{R}_p$ contains all technicians $r \in \mathcal{R}_{i_p}$ such that for every time period t we have $\rho_r^t \geq a_p^t$ and for every day $d \in \mathcal{D}_p$ and every time period $t \in \mathcal{T}_d$, we have $\pi_r^t = 1$ or both $\rho_r^t = 0$ and $\sigma_{l_p l_r^t} = 1$. We also define q_p as the number of technicians required to execute plan p. Finally, parameter o_p is the penalty incurred if plan p is selected (note that $\forall i \in \mathcal{I}, \forall p \in \mathcal{P}_i \setminus \{p_i^0\},\ o_p = 0$ and $o_{p_i^0} = o_i$).

Scheduling the tasks becomes rather implicit as it simply means to select a plan for each task. Nevertheless, we still need to manage the technician-to-task assignments without violating for each technician the daily location-based incompatibilities. Clearly, technicians can be indifferently assigned to plans sharing the same starting and completion time periods, the same location and the same skill. These previous four parameters define what we call a *pattern*. Directly assigning technicians to patterns results in a smallest number of assignment variables. We denote $\mathcal{H}$ and h_p the set of all patterns and the pattern linked to plan $p \in \mathcal{P}$. We also define as $\mathcal{P}_h$ the set of plans sharing the same parameters than pattern $h \in \mathcal{H}$. For convenience, for a pattern h, we introduce parameters s_h, l_h and $\mathcal{R}_h$ that respectively define: its required skill, its location, and the set of technicians that can be assigned to it. Conversely, we define $\mathcal{H}_l = \{h \in \mathcal{H} \mid l_h = l\}$ the set of patterns associated with location $l \in \mathcal{L}$. For a pattern h, we express its active time periods by a binary vector a_h over $\mathcal{T}$ such that $a_h^t = 1$ if and only if h is active during time period $t \in \mathcal{T}$ (S_h and C_h are used to represent the starting and completion time periods of the pattern).

From these previous notions (plan and pattern), we create two new ILP formulations that only differ on the space-time tracking of the technicians. We still use some decision variables defined in Section 4.1, but we also introduce the following decision variables:

$$x_p = \begin{cases} 1 & \text{if plan } p \in \mathcal{P} \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{rh} = \begin{cases} 1 & \text{if technician } r \in \mathcal{R}_p \text{ is assigned to pattern } h \in \mathcal{H}, \\ 0 & \text{otherwise.} \end{cases}$$

4.2.1 Baseline formulation

The first *compact* formulation of the problem is defined as the following integer linear program denoted as $[P2]$.

$$[P2] \quad \max \sum_{w\in\mathcal{W}} \left(\sum_{t\in\mathcal{T}} g_w^t f_w^t + \sum_{d\in\mathcal{D}} \widetilde{g}_w^d \widetilde{f}_w^d \right) - \sum_{p\in\mathcal{P}} o_p x_p \tag{4.32}$$

subject to:

$$\sum_{p\in\mathcal{P}_i} x_p = 1 \ \ \forall i \in \mathcal{I}, \tag{4.33}$$

$$\sum_{p\in\mathcal{P}_i} a_p^t x_p \leq \vartheta_{i_p}^t \ddot{\vartheta}_{i_p}^t \ \ \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.34}$$

$$\sum_{i\in\mathcal{B}} \sum_{p\in\mathcal{P}_i} a_p^t x_p \leq 1 \ \ \forall \mathcal{B} \in ov(\mathcal{I}), \forall t \in \mathcal{T}, \tag{4.35}$$

$$f_w^t + \sum_{p\in\mathcal{P}_i} b_{wp} a_p^t x_p \leq 1 \ \ \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{4.36}$$

$$\widetilde{f}_w^d + \sum_{p \in \mathcal{P}_i} \widetilde{b}_{wp} \widetilde{a}_p^d x_p \leq 1 \;\; \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall d \in \mathcal{D}, \tag{4.37}$$

$$\sum_{i \in \mathcal{I} | s_i \in \bar{\mathcal{S}}} \sum_{p \in \mathcal{P}_i} a_p^t q_p x_p \leq |R_{\bar{\mathcal{S}}}^t| \;\; \forall t \in \mathcal{T}, \forall \bar{\mathcal{S}} \subseteq \mathcal{S}, \tag{4.38}$$

$$\sum_{r \in \mathcal{R}_h} y_{rh} = \sum_{p \in P_h} q_p x_p \;\; \forall h \in \mathcal{H}, \tag{4.39}$$

$$\sum_{h \in \mathcal{H}_l} a_h^t y_{rh} \leq \rho_r^t v_{rl}^t \;\; \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}, \tag{4.40}$$

(4.18), (4.19), (4.20), (4.28)

$$x_p \in \{0,1\} \;\; \forall p \in \mathcal{P}, \tag{4.41}$$

$$f_w^t \in \{0,1\} \;\; \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \tag{4.42}$$

$$\widetilde{f}_w^d \in \{0,1\} \;\; \forall w \in \mathcal{W}, \forall d \in \mathcal{D}, \tag{4.43}$$

$$y_{rh} \in \{0,1\} \; \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h \tag{4.44}$$

The objective in (4.32) is defined as the difference between the revenue generated by the wind turbines and the penalties related to the postponement of some tasks. Constraints (4.33) ensure that at least one plan involving each task is selected (i.e., each task is either executed or postponed). Constraints (4.34) are related to the restrictions on the time periods during which each task can be effectively executed. Notice that these constraints are not needed if we build the set of plans such that for every plan $p \in \mathcal{P}$ we have $a_p^t \leq \vartheta_{i_p}^t \ddot{\vartheta}_{i_p}^t$. In our experiments, we build the set $\mathcal{P}$ to avoid including these constraints. Constraints (4.35) are the non-overlapping constraints. Constraints (4.36) and (4.37) couple turbine availability variables to plan selection variables. Constraints (4.39) ensure that the technician requirements are fulfilled. Constraints (4.40) couple the locations of the technicians to the tasks they perform. Constraints (4.18), (4.19), and (4.20) are used to handle the daily location-based incompatibilities and the availability calendars of the technicians. Finally, constraints (4.28),(4.41)-(4.44) state the binary nature of the decision variables.

To strengthen the formulation $[P2]$, we add the cumulative scheduling constraints (4.38), even if they are redundant since they can be deduced from constraints (4.39) and (4.40). To build these constraints, we introduce for each time period $t \in \mathcal{T}$ the bipartite graph $\mathcal{G}^t = \left(\left(\mathcal{S}, \mathcal{R}^t\right), \mathcal{U}^t\right)$ in which, with a slight abuse of notation, vertices from $\mathcal{S}$ represent skills, vertices from $\mathcal{R}^t$ indicate technicians available during time period t (i.e., $\mathcal{R}^t = \{r \in \mathcal{R} | \rho_r^t = 1\}$), and edges from $\mathcal{U}^t$ are defined as follows: $\forall s \in \mathcal{S}, \forall r \in \mathcal{R}^t \;\; (s,r) \in \mathcal{U}^t$ if and only if $\zeta_{rs} = 1$. Applying a generalization of König-Hall theorem (see Lemma 4.2.1), the constraints (6.7) thus correspond to necessary and sufficient condition of the existence of a maximum cardinality b-matching from $\mathcal{S}$ to $\mathcal{R}^t$ where function b is defined for every vertex s that belongs to $\mathcal{S}$ by $b(s) = \sum_{i \in \mathcal{I} | s_i = s} \sum_{p \in \mathcal{P}_i} a_p^t q_p x_p$, and by $b(r) = 1$ for every vertex r that belongs to $\mathcal{R}^t$. To express these constraints, we denote $\mathcal{R}_{\bar{\mathcal{S}}}^t$ the set of technicians available during time period t and mastering at least one skill in subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$ (i.e., $\mathcal{R}_{\bar{\mathcal{S}}}^t = \{r \in \mathcal{R} | \exists s \in \bar{\mathcal{S}}, \zeta_{rs} = 1 \wedge \rho_r^t = 1)\}$).

Theorem 4.2.1. ***(König-Hall theorem):***
Let $\mathcal{G} = [\mathcal{X} \cup \mathcal{Y}, \mathcal{U}]$ be a bipartite graph, a matching of $\mathcal{G}$ saturating $\mathcal{X}$ exists if and only if $\forall \hat{\mathcal{X}} \subseteq X, |N(\hat{\mathcal{X}})| \geq |\hat{\mathcal{X}}|$ where $\mathcal{N}(\hat{\mathcal{X}})$ denotes the set of vertices of $\mathcal{Y}$ adjacent to the vertices of $\hat{\mathcal{X}}$.

Lemma 4.2.1. ***(Generalization of König-Hall theorem):***
Let $\mathcal{G} = [\mathcal{X} \cup \mathcal{Y}, \mathcal{U}]$ be a bipartite graph and $b : V \to \mathbb{N}$ a function. A b-matching of $\mathcal{G}$ saturating $\mathcal{X}$ exists if and only if $\forall \hat{\mathcal{X}} \subseteq X, \sum_{v \in N(\hat{\mathcal{X}})} b(v) \geq \sum_{v \in \hat{\mathcal{X}}} b(v)$ where $\mathcal{N}(\hat{\mathcal{X}})$ denotes the set of vertices of $\mathcal{Y}$ adjacent to the vertices of $\hat{\mathcal{X}}$.

Last but not least, the number of these constraints are exponential $\left(2^{|\mathcal{S}|} - 1\right) \times |\mathcal{T}|$). In our experiments, however, the number of these constraints tends to be small; we therefore add all these constraints to our model.

4.2.2 Alternative formulation

A potential improvement of the previous model concerns the space-time tracking of the technicians. Observing that the number of constraints (4.20) is usually very large, we attempt to develop an alternative technician management strategy. This new strategy is based on finding all the maximal cliques (cliques that cannot be enlarged) in a graph where each vertex represents a location, and there exists an edge between two vertices if the underlying locations l and l' can be visited during the same day by the same technician (i.e., we have $\sigma_{ll'} = 1$). Figure 4.1 illustrates the computation of these maximal cliques. For the purpose of finding all the maximal cliques, we use the algorithm introduced by Bron and Kerbosch (1973).

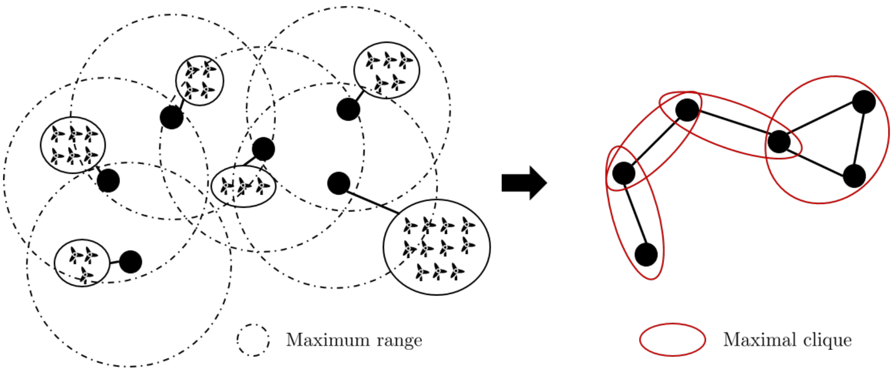


Figure 4.1 – Example of how maximal cliques are computed

Denoting $\mathcal{K}$ the set of all maximal cliques in the previous graph, we define $\mathcal{K}_r^d$ as the set of cliques to which technician r can be assigned during day d. More specifically, $\mathcal{K}_r^d$ is equal to $\{k \in \mathcal{K} \mid \forall t \in \mathcal{T}_d, \rho_r^t = 1 \vee \left(\rho_r^t = 0 \wedge l_r^t \in k\right)\}$. We then introduce the following binary variables:

$$u_{rk}^d = \begin{cases} 1 & \text{if during day } d \in \mathcal{D} \text{ technician } r \in \mathcal{R} \text{ can only performed tasks at locations included in clique } k \in \mathcal{K}_r^d \\ 0 & \text{otherwise.} \end{cases}$$

Denoting d_t as the day to which time period t belongs, we can track the location of the technicians with the following constraints:

$$\sum_{k \in \mathcal{K}_r^d} u_{rk}^d = 1 \quad \forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \tag{4.45}$$

$$\sum_{h \in \mathcal{H} \mid r \in \mathcal{R}_h} a_h^t y_{rh} \leq \rho_r^t \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \tag{4.46}$$

$$\sum_{h \in \mathcal{H}_l} a_h^t y_{rh} \leq \sum_{k \in \mathcal{K}_r^{d_t} \mid l \in k} u_{rk}^{d_t} \quad \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T} \tag{4.47}$$

$$u_{rk}^d \in \{0,1\} \quad \forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \forall k \in k \in \mathcal{K}_r^d \tag{4.48}$$

Constraints (4.45) state that a technician is assigned to only one clique on each day. This ensures the compliance with the location-based incompatibility constraints. Constraints (4.46) prevent a technician to be assigned to multiple tasks during a given time period and constraints (4.47) couple assignment and space-time tracking variables.

Finally, we define as $[P3]$ the model resulting from replacing constraints (4.18)-(4.20),(4.28),(4.40) by constraints (4.45)-(4.48) in the formulation $[P2]$.

4.2.3 Breaking symmetries

From a feasible solution to the problem, we observe that we can build alternative solutions by just switching the tasks assigned to two technicians who master the same skills (as long as the new assignment is compatible with both techicians' unavailability time periods and the daily-location based incompatibilities). The technician-to-task assignments therefore faces symmetries which usually affect the efficiency of a direct resolution using an ILP solver. One way to counter that is to add symmetry breaking constraints. First, we decompose $\mathcal{R}$ into n disjoint subsets $\{\widetilde{\mathcal{R}}_b\}_{b\in\{1,..,n\}}$ of technicians such that each subset $\widetilde{\mathcal{R}}_b$ contains all technicians mastering the same skills. Clearly, we have $\mathcal{R} = \bigcup_{b\in\{1,..,n\}} \widetilde{\mathcal{R}}_b$ and $\widetilde{\mathcal{R}}_b \cap \widetilde{\mathcal{R}}_{b'} = \emptyset$ when $b \neq b'$. We also introduce two additional parameters. Binary parameter $\varsigma_{hh'}$ is equal to 1 if and only if a technician can be assigned to plans h and $h^{'}$ considering the daily location-based incompatibilities. Binary parameter τ_{rh} takes value 1 if and only if technician $r \in \mathcal{R}$ can be assigned to pattern $h \in \mathcal{H}$ regarding its own personal availability schedule. We also consider here the set $\mathcal{H}$ as a totally ordered set.

We introduce the following symmetry breaking constraints:

$$y_{rh} \leq y_{(r-1)h} + \left(\sum_{h'\in\mathcal{H}|\varsigma_{hh'}=0\wedge h'<h} y_{(r-1)h'} \right) + \left(1 - \tau_{(r-1)h}\right) \quad \forall b \in \{1,..,n\}, \forall r \in \widetilde{\mathcal{R}}_b \setminus \{1\}, \forall h \in \mathcal{H} \; s.t. \; r \in \mathcal{R}_h \tag{4.49}$$

These constraints rank technicians having the same skills according to a lexicographic order, and ensure that the technician r can be assigned to a pattern h only if one of the three following conditions hold:

— The technician $r-1$ is assigned to a pattern h

— The technician $r-1$ is assigned to a pattern h' with an index less than h (we assume that the patterns are ranked) that prevents him to work on pattern h.

— The technician $r-1$ is unavailable during a time period that prevents him or her to be assigned to pattern h

One should however observe that the number of these constraints (4.49) is very large which may prevent them to be efficient in practice.

4.3 Computational experiments

We ran our experiments on a Linux 64 bit-machine, with an Intel(R) Xeon(R) X5675 (3.07Ghz) and 12GB of RAM. We rely on *Gurobi 6.5.1* for solving the ILP formulations of the problem. We set a 3-hour time limit to solve the different formulations (notice that all CPU times are reported in seconds and rounded to the closest integer). In order to assess the quality of our results, we compute the gap with respect to the optimal solution when it is known, or to the best upper bound we found among all the tests reported in this thesis manuscript. More precisely, all gaps reported in the manuscript are computed as: $gap = (z^{UB} - z)/|z|$, where z is the objective function of the computed solution and z^{UB} is the objective function of the optimal solution or the best (minimal) upper bound.

We do not include the symmetry breaking constraints (4.49) in formulations $[P2]$ and $[P3]$ as preliminary tests show that they do not help speeding up the resolution of the models.

In Table 4.1, we report the average number of variables (#Vars) and constraints (#Cstrs) in each ILP formulation. We also show in Table 4.2 the average number of plans (#Plans), patterns (#Patterns), and location-based cliques used in formulation $[P4]$ (#Cliques). These figures allow the reader to assess the size of the different formulations. We observe that the two compact formulations [P2] and [P3] have around seven and twelve times less constraints than the natural formulations [P1] and [P1bis] while having a small number of additional variables. Notice also that the number of plans is low-enough to lead to models that can be handled by ILP solvers.

In Table 4.3, we report the average, over all the instances belonging to the same family or sharing a common characteristic, of: the gap (Gap), the solution time (Time), and the percentage of tasks scheduled (i.e., not-posponed) in the best solution (%S). We also report the number of optimal solutions found within the time limit (#Opt).

Table 4.1 – Average number of variables and constraints in each ILP formulation

Testbed	[P1]		[P1bis]		[P2]		[P3]	
	#Vars	#Cstrs	#Vars	#Cstrs	#Vars	#Cstrs	#Vars	#Cstrs
G1	50	129	50	126	59	24	51	13

NB: numbers are in thousands in the table are rounded to the nearest thousand

Table 4.2 – Average number of plans, patterns and location-based cliques

Testbed	#Plans	#Patterns	#Cliques
G1	2,394	1,218	8

NB: numbers are rounded to the nearest integer

Remark: In order to have a meaningful comparison, the average solution times only takes into account those instances for which an optimal solution has been found within the time limit. Similarly, the average gap and percentage of scheduled tasks takes into account only the instances which are not optimally solved. This allows a better understanding of the results. Indeed, since in our instances postponing a task is non-profitable and heavily penalized, a large gap is often related to a low percentage of tasks scheduled during the time horizon. Notice that on average 99% of the tasks are scheduled in the optimal or best-known solutions for testbed G1.

First, we observe that the compact formulations $[P2]$ and $[P3]$ outperform by far the natural formulations $[P1]$ and $[P1bis]$ ($[P1bis]$ seems slightly better than $[P1]$) for small and medium-sized instances. For the large-sized instances, all the formulations struggle reaching optimal solutions, but compact formulations perform worst than natural formulations (as they fail more often to schedule a large proportion of the tasks). We believe these results can be explained as follows. The compact formulations contains far less constraints than the natural formulations and the value of the LP relaxation is around 2% smaller on average, which leads to tighter upper bounds computed by the ILP solver.

Formulation $[P3]$ also seems to slightly outperform formulation $[P2]$. However, the comparison between these two formulations is difficult as the best model regarding the gap and solution time if it is not optimally solved can vary within a family from one instance to another one. This may highlight the erratic behavior of the solver for the medium and large-sized instances when one does not tailor the search and leaves the solver to make branching decisions and to build heuristic solutions. It is then hard to derive definitive conclusions on the relative efficiency of formulations $[P2]$ and $[P3]$. We also observe that, at least in our 3-hour time limit, optimality is only reached for small-sized instances and that whenever optimality is reached the CPU time is rather long (around 30 minutes on average).

In summary, the average gap when directly solving the ILP formulations is considerable for the majority of the families of instances (the solver fails to schedule a large proportion of the tasks), and when optimality is reached it is on average after a considerable solution time. It is not very surprising since the formulations only involve binary variables and their size is quite large. We therefore reach the following conclusion: directly solving the ILP formulations using a commercial solver does not yield suitable exact approaches for the problem.

Table 4.3 – Detailed computational results on testbed G1 when solving the ILP formulations.

Family	[P1]				[P1bis]				[P2]				[P3]			
	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time
10_2_1_20_A	1.4%	97%	0/5	-	1.3%	98%	1/5	2,368	-	-	5/5	598	-	-	5/5	**152**
10_2_1_20_B	0.01%	100%	4/5	744	0.01%	100%	4/5	1,242	-	-	5/5	**12**	-	-	5/5	37
10_2_1_40_A	7.1%	95%	0/5	-	5.9%	96%	0/5	-	0.01%	100%	**2/5**	1,813	0.01%	100%	1/5	7
10_2_1_40_B	0.00%	100%	3/5	6,015	0.00%	100%	3/5	4,421	-	-	5/5	205	-	-	5/5	**121**
10_2_3_20_A	2.0%	96%	1/5	326	1.4%	96%	2/5	891	-	-	5/5	2,635	-	-	5/5	**1,574**
10_2_3_20_B	0.02%	100%	3/5	305	0.0%	100%	3/5	180	-	-	5/5	31	-	-	5/5	**18**
10_2_3_40_A	9.6%	96%	0/5	-	8.3%	96%	0/5	-	-	-	5/5	**3,220**	-	-	5/5	3,996
10_2_3_40_B	0.00%	100%	4/5	2,857	0.03%	100%	4/5	1,707	-	-	5/5	**180**	-	-	5/5	295
20_2_1_40_A	42%	76%	0/5	-	7.7%	93%	0/5	-	2.4%	98%	2/5	8,074	**1.4%**	98%	2/5	2,858
20_2_1_40_B	1.6%	99%	0/5	-	2.0%	99%	0/5	-	0.01%	100%	4/5	232	-	-	**5/5**	2,078
20_2_1_80_A	28%	87%	0/5	-	**19%**	90%	0/5	-	436%	0%	0/5	-	334%	20%	0/5	-
20_2_1_80_B	6.2%	96%	0/5	-	3.0%	98%	0/5	-	318%	50%	3/5	**1,485**	229%	49%	3/5	3,823
20_2_3_40_A	6.2%	93%	0/5	-	4.2%	96%	0/5	-	1.25%	99%	1/5	322	1.2%	99%	**3/5**	5,534
20_2_3_40_B	0.02%	100%	2/5	2,561	0.05%	100%	4/5	3,322	-	-	5/5	376	-	-	5/5	**155**
20_2_3_80_A	23%	89%	0/5	-	**17%**	91%	0/5	-	257%	20%	0/5	-	156%	39%	0/5	-
20_2_3_80_B	4.3%	97%	0/5	-	1.3%	99%	0/5	-	-	-	**5/5**	3,415	196%	50%	3/5	2,456
20_4_1_20_A	5.3%	93%	0/5	-	6.3%	91%	0/5	-	1.8%	96%	0/5	-	**1.3%**	97%	0/5	-
20_4_1_20_B	0.25%	100%	3/5	6,264	1.3%	98%	2/5	1,726			5/5	**204**	-	-	5/5	405
20_4_1_40_A	161%	46%	0/5	-	**43%**	81%	0/5	-	264%	0%	0/5	-	61%	75%	0/5	-
20_4_1_40_B	12.8%	91%	0/5	-	8.6%	94%	0/5	-	174%	49%	1/5	5,208	107%	74%	1/5	**1,968**
20_4_3_20_A	7.9%	92%	0/5	-	6.9%	93%	0/5	-	1.2%	98%	0/5	-	2.4%	95%	**4/5**	5,005
20_4_3_20_B	1.2%	98%	2/5	2,152	0.55%	100%	2/5	3,251	0.01%	100%	4/5	52	-	-	**5/5**	113
20_4_3_40_A	416%	33%	0/5	-	25%	84%	0/5	-	373%	20%	0/5	-	**5.3%**	95%	0/5	-
20_4_3_40_B	204%	56%	0/5	-	14%	92%	0/5	-	1.5%	99%	2/5	6,112	0.03%	100%	**3/5**	2,108
40_4_1_40_A	147%	54%	0/5	-	**44%**	77%	0/5	-	352%	0%	0/5	-	106%	76%	0/5	-
40_4_1_40_B	157%	73%	0/5	-	8.3%	94%	0/5	-	1,594%	40%	0/5	-	**3.0%**	98%	0/5	-
40_4_1_80_A	**49%**	80%	0/5	-	50%	77%	0/5	-	4,948%	0%	0/5	-	4,948%	0%	0/5	-
40_4_1_80_B	**39%**	83%	0/5	-	42%	83%	0/5	-	331%	0%	0/5	-	331%	0%	0/5	0-
40_4_3_40_A	170%	43%	0/5	-	36%	78%	0/5	-	1,087%	20%	0/5	-	**4.6%**	96%	0/5	-
40_4_3_40_B	13%	88%	0/5	-	13%	90%	0/5	-	477%	20%	0/5	-	0.8%	99%	**2/5**	2,118
40_4_3_80_A	**48%**	77%	0/5	-	55%	77%	0/5	-	2,813%	0%	0/5	-	2,727%	18%	0/5	-
40_4_3_80_B	24%	84%	0/5	-	**15%**	89%	0/5	-	3,899%	0%	0/5	-	3,899%	0%	0/5	-

Characteristic		[P1]				[P1bis]				[P2]				[P3]			
		Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time
$\lvert\mathcal{S}\rvert =$	1	47%	83%	10/80	3,981	17%	91%	10/80	2,405	854%	35%	32/80	1,108	621%	61%	**34/80**	1,035
	3	68%	81%	12/80	1,841	15%	91%	14/80	2,004	1,036%	39%	37/80	1,677	982%	61%	**43/80**	1,896
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	10%	94%	17/80	2,283	5.9%	96%	20/80	2,107	179%	56%	57/80	**1,383**	144%	60%	57/80	1,461
	4	97%	73%	5/80	4,619	24%	87%	4/80	2,488	1,197%	30%	12/80	1,555	1,014%	61%	**20/80**	1,757
Type =	A	71%	78%	1/80	326	21%	88%	2/80	1,629	878%	38%	20/80	2,618	759%	65%	**25/80**	2,838
	B	39%	88%	21/80	2,932	9%	94%	22/80	2,220	1059%	35%	49/80	921	802%	53%	**52/80**	913
All		57%	82%	22/160	2,814	16%	91%	24/160	2,171	940%	37%	69/160	1,413	773%	61%	**77/160**	1,538

For testbed G1, our results suggests that the number of skills does not have a strong impact on the difficulty of the instances (although we observe that instances with 3 skills appear to be easier to solve). This may be a result of a smaller number of symmetries among the technicians and of feasible configurations to schedule the tasks. On the other hand, the number of tasks seems to have an impact on the difficulty of the type A instances. This can be explained by the higher difficulty of finding a maintenance plan when considering more tasks. Moreover, the ILP formulations perform better on instances with 2 time periods per day; the solution time is shorter and the number of optimal solutions is larger than in those with 4 time periods per day. A plausible explanation is that the daily location-based incompatibilities are more binding in the latter case. Indeed, a larger number of periods provides a wider choice of task starting times and therefore more opportunities to move technicians between locations during a single day. Instances with 4 time periods per day also have a larger number of plans and patterns; this may also explain their higher difficulty. In conclusion, according to our experiments on testbed G1, the difficulty of an instance increases with the number of time periods per day and the tightness of the technicians-to-work ratio.

Chapter 5

A constraint programming-based large neighborhood search to solve the deterministic problem

The research reported in this chapter (except Section 5.5) has been wrapped up on a article submitted for possible publication in Journal of Scheduling and currently under second round of review.

Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016). Solving a wind turbine maintenance scheduling problem. Submitted to *Journal of Scheduling*. Under review.

This chapter presents a heuristic solution method for the problem defined in Chapter 3. Section 5.1 formally defines the problem using an alternative programming paradigm to ILP, namely, constraint programming (CP). Section 5.2 introduces a constraint programming-based large neighborhood search (CPLNS) to tackle the problem. The proposed approach uses destruction operators either specifically conceived for the problem or adapted from the literature. The repair operator consists in solving a CP model with some fixed variables using branching strategies specially tailored for the problem. We also present a new acceptance criterion based on elitism and Metropolis. Section 5.3 reports computational experiments on testbed G1. Section 5.4 concludes the research presented in this chapter up to this point. At the end of the chapter, Section 5.5 outlines our specific work with WPred as the CPLNS is the solution method implemented for the industrial prototype we provide them. We point out specifities of the problem being solved and briefly discuss the inclusion of the created optimization tool in their existing product.

5.1 Constraint programming formulation

Chapter 4 presents several ILP formulations of the problem. Motivated by the successful implementation of CP models for solving other hard and, to some extend, related optimization problems (Baptiste et al., 2001; Rodriguez, 2007; Malapert et al., 2012), we decided to approach our problem using CP.

First of all, note that defining for each task: i) an execution mode, ii) a starting time, and iii) the technicians assigned to it, is enough to obtain a solution to our problem. Therefore, for each task $i \in \mathcal{I}$, we introduce the variables $M_i \in \mathcal{M}_i$ and $S_i \in \mathcal{T}$ to represent its execution mode and starting time period, and we use binary variables $(y_{ri})_{r \in \mathcal{R}_i}$ to decide if technician r performs or not task i. More specifically, y_{ri} is equal to 1 if and only if the technician r is assigned to task i. To make some constraints easier to model, we introduce integer variables $C_i \in \mathcal{T}$, $D_i \in \{d_{im}\}_{m \in \mathcal{M}_i}$, $Q_i \in \{q_{im}\}_{m \in \mathcal{M}_i}$ and set variables $E_i \subseteq \{t \in \mathcal{T} \mid \vartheta_i^t \ddot{\vartheta}_i^t = 1\}$ defining for task i its completion time period, its duration, its number of assigned technicians, and its set of execution time periods, respectively.

Execution time periods of each task are coupled to their starting and ending time periods with constraints (5.1) and (5.2).

$$S_i + D_i - 1 = C_i \quad \forall i \in \mathcal{I}, \tag{5.1}$$
$$t \in E_i \Leftrightarrow t \in \{S_i, ..., C_i - 1\} \quad \forall i \in \mathcal{I} \tag{5.2}$$

For each task, its duration (5.3) as well as the number of assigned technicians (5.4) are coupled with the selected execution mode.

$$D_i = d_{iM_i} \quad \forall i \in \mathcal{I}, \tag{5.3}$$
$$Q_i = q_{iM_i} \quad \forall i \in \mathcal{I} \tag{5.4}$$

Constraints (5.5) are the non-overlapping constraints.

$$\bigcap_{i \in \mathcal{B}} E_i = \emptyset \quad \forall \mathcal{B} \in ov(\mathcal{I}) \tag{5.5}$$

Constraints (5.6) ensure that the technician requirements are fulfilled for each task.

$$\sum_{r \in \mathcal{R}_i} y_{ri} = Q_i \quad \forall i \in \mathcal{I} \tag{5.6}$$

To forbid a technician $r \in \mathcal{R}$ to be assigned to multiple tasks during a time period, we introduce set variables $Y_r^t \subseteq \mathcal{I} \cup \{i^0\}$ defining the set of tasks that the technician can potentially perform during time period $t \in \mathcal{T}$. Index i^0 represents a dummy task, created in order to prevent a technician to work during his or her unavailability time periods. Constraints (5.7) couple these variables to the global assignment variables $(y_{ri})_{i \in \mathcal{I} | r \in \mathcal{R}_i}$. Restrictions imposed on the locations visited by a technician within each day lead to the introduction of the set variables $V_r^t \subseteq \mathcal{L}$ defining the set of potential locations for technician r during time period t. Constraints (5.8) and (5.9) restrict the set of tasks that a technician can possibly execute according to his or her potential locations. Set $\mathcal{L}(\hat{\mathcal{I}})$ defines the set of locations of the tasks in set $\hat{\mathcal{I}}$. Note that $\mathcal{L}(\hat{\mathcal{I}}) = \{l \in \mathcal{L} \mid \exists i \in \hat{\mathcal{I}} \ s.t. \ l_i = l\}$.

$$y_{ri} = 1 \Rightarrow \left(Y_r^t = \{i\} \quad \forall t \in E_i\right) \quad \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i, \tag{5.7}$$
$$V_r^t = \mathcal{L}(Y_r^t) \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \ s.t. \ \rho_r^t = 1, \tag{5.8}$$
$$V_r^t = \{l_r^t\} \wedge Y_r^t = \{i^0\} \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \ s.t. \ \rho_r^t = 0 \tag{5.9}$$

Constraints (5.10) ensure that the daily location-based incompatibilites are not violated for each technician.

$$V_r^t = \{l\} \Rightarrow \left(l' \notin V_r^{t'} \ \ \forall l' \in \mathcal{L} \ s.t. \ \sigma_{ll'} = 0, \forall t' \in \mathcal{T}_{d_t} \ s.t. \ t' \neq t\right) \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \tag{5.10}$$

In order to define the objective function of our problem, we introduce two set variables. Variables $F_w^{day} \subseteq \{1, \ldots, |\mathcal{T}|\}$ define the set of all time periods during which turbine $w \in \mathcal{W}$ can produce electricity. Variables $F_w^{rest} \subseteq \{1, \ldots, |\mathcal{D}|\}$ define the set of days for which turbine w can produce electricity during the corresponding rest time periods. More specifically, a day d belongs to this set if turbine w can produce electricity during the rest time period that occurs between d and $d+1$. Additionally, we denote by t_d^{rest} the last period $t \in \mathcal{T}$ before the rest time period following day $d \in \mathcal{D}$. We introduce constraints (5.11), (5.12), (5.13), and (5.14) which state that a turbine is available to produce electricity during a time period if and only if no tasks requiring its shutdown are scheduled at the same time.

$$t \in E_i \Rightarrow t \notin F_w^{day} \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I} \ s.t. \ b_{wi} = 1, \forall t \in \mathcal{T}, \tag{5.11}$$

$$t \notin \bigcup_{i \in \mathcal{I} | b_{wi}=1} E_i \Rightarrow t \in F_w^{day} \quad \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \tag{5.12}$$

$$t_d^{rest} \in E_i \wedge (t_d^{rest} + 1) \in E_i \Rightarrow d \notin F_w^{rest} \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \ s.t. \ \widetilde{b}_{wi} = 1, \forall d \in \mathcal{D}, \tag{5.13}$$

$$\bigwedge_{i \in \mathcal{I} | \widetilde{b}_{wi}=1} (\{t_d^{rest}, t_d^{rest} + 1\} \not\subseteq E_i) \Rightarrow d \in F_w^{rest} \quad \forall w \in \mathcal{W}, \forall d \in \mathcal{D} \tag{5.14}$$

Constraint (5.15) defines the objective function variable $obj \in \mathbb{R}$ of our problem.

$$obj = \sum_{w \in \mathcal{W}} \left(\sum_{t \in F_w^{day}} g_w^t + \sum_{d \in F_w^{rest}} \widetilde{g}_w^d \right) - \sum_{i \in \mathcal{I} | M_i = m_i^0} o_i \tag{5.15}$$

To remove some symmetries, we add constraints (5.16) to impose the starting time period of a postponed task to be equal to 0.

$$\mathcal{M}_i = \mathcal{M}_i^0 \Leftrightarrow S_i = 0 \quad \forall i \in \mathcal{I} \tag{5.16}$$

5.2 A constraint programming-based large neighborhood approach

We use the CP model introduced in Section 5.1 as the main building block of a constraint programming-based large neighborhood (CPLNS) approach. This method is based on the LNS metaheuristic presented in Section 2.2. In the CPLNS, we use the CP model to repair the solutions that have been destroyed. We tested the use of the ALNS framework rather than the LNS framework. After some preliminary experimentation we decided to drop the adaptive layer because its contribution to the accuracy of the method did not payoff the loss of simplicity and the effort needed to fine tune the additional parameters. We then randomly select operators with equal probability. The general structure of the method is outlined in Algorithm 2 (in Section 2.2). To compute the initial solution, we use the CP model and we stop the execution as soon as we find the first solution.

5.2.1 Destroy operators

At each iteration, the algorithm selects Γ tasks to remove from the current solution. The value of Γ is randomly fixed in the interval $[\max(n^-, n \times p^-); \min(n^+, n \times p^+)]$, where n^- and n^+ denote the minimal and maximal number of tasks that are allowed to be removed during an iteration; similarly, p^- and p^+ denote the minimal and maximal proportion of tasks that could be removed. The parameters p^- and p^+ allow the algorithm to adapt to all instances independently of their size. We use the following settings: $(n^-, n^+, p^-, p^+) = (5, 20, 0.1, 0.4)$. We also always consider postponed tasks in the current solution as tasks to be removed. However, we do not count them among the Γ tasks to remove.

After setting Γ, the algorithm selects the tasks using one of the following six removal operators:

— Operator A: random removal
This operator randomly removes Γ tasks from the current solution. The intention behind this operator is to diversify the search.

— Operator B: worst removal
This operator removes the tasks which penalize the most the objective function of the current

solution. Let f be the current value of the objective function, f_{-i} its value if task i is removed, and $\Delta f(i) = f - f_{-i}$. The Γ tasks with the greatest values of $\Delta f(i)$ are removed from the current solution in order to insert them at better positions.

— Operator C: technicians duties removal
This operator is based on the following procedure. First, it randomly selects a skill s^*. Second, as long as the number of removed tasks is lower than Γ, it randomly selects a technician mastering s^* and remove from the current solution those tasks in which the selected technician uses skill s^*. The operator then switches to another skill if it has not removed Γ tasks yet. Freeing up a pool of technicians along the whole time horizon may allow the reinsertion of possibly misplaced tasks at more convenient time periods (i.e, periods where they penalize less the revenue).

— Operator D: similar tasks removal
This operator removes *similar tasks*. More specifically, the operator aims to remove non-overlapping tasks (or tasks that overlap as little as possible) having similar duration and skill requirements. The similarity between two tasks $i, j \in \mathcal{I}$ in a solution sol is formally defined as: $\phi(i, j, sol) = \alpha_1 \times |\bar{d}_i - \bar{d}_j| + \alpha_2 \times \mathbb{1}_{(s_i \neq s_j)} + \alpha_3 \times ov(i, j, sol)$, where $\bar{d}_i$ is the average duration of task i (i.e., $\bar{d}_i = \dfrac{1}{|\mathcal{M}_i \setminus \{m_i^0\}|} \sum_{m \in \mathcal{M}_i \setminus \{m_i^0\}} d_{im}$). Function $ov(i, j, sol)$ computes the number of overlapping time periods between i and j in the current solution sol. Coefficients α_1, α_2, and α_3 weight the three components of the similarity function, namely, task duration, skill requirements, and task overlapping. In our experiments, $(\alpha_1, \alpha_2, \alpha_3) = (1, 3, 5)$. To select the tasks to remove, the operator first initializes a set $\widetilde{\mathcal{I}}$ with a random task. While $|\widetilde{\mathcal{I}}| \leq \Gamma$, the procedure randomly selects a task i^* from $\widetilde{\mathcal{I}}$, and it adds to $\widetilde{\mathcal{I}}$ the task $j \in \mathcal{I} \setminus \widetilde{\mathcal{I}}$ with the minimal value of $\phi(i^*, j, sol)$. The intuition behind this operator is that removing and re-inserting similar tasks that are scheduled in non-overlapping time periods increases the likelihood of a solution improvement.

— Operator E: task maximal regret
This operator removes the tasks having the largest difference between the loss of revenue they currently generate and the minimal loss of revenue they can induce (we called this difference *regret*). Let us denote $\mathcal{W}_i$ the set of turbines shut down by the execution of a task i (clearly, $\mathcal{W}_i = \{w \in \mathcal{W} | b_{wi} = 1 \vee \widetilde{b}_{wi} = 1\}$). The loss induced by task i is equal to the sum over all the turbines in $\mathcal{W}_i$ of the revenue lost due to its scheduling. Notice that if multiple tasks impact a turbine during a specific time period, the loss is set proportionally to the number of these tasks. Prior to the optimization, the operator computes for each task i a metric called $loss_i^{best}$ equal to the smallest loss of revenue that can be achieved when one only considers the scheduling of this task. Then, during the optimization, the operator first computes the lost revenue $loss_i^{sol}$ generated by task i in the current solution sol. Afterwards, the operator computes the regret $\Delta loss(i) = loss_i^{sol} - loss_i^{best}$ for each scheduled task i. The operator then removes from the current solution sol the Γ tasks associated with the largest value of $\Delta loss(i)$. Removing tasks that currently generate considerably more loss of revenue than they could may allow the algorithm to schedule those tasks in better positions in the next iterations. It is then plausible to assume that this operator increases the probability of finding better-quality solutions.

— Operator F: turbine maximal regret
This operator works almost in the same way as operator E. Instead of reasoning by task, we focus on each turbine. Prior to the optimization, the procedure computes for each turbine $w \in \mathcal{W}$ a metric called $loss_w^{best}$, estimating the smallest loss of revenue that can be achieved when one only considers the set $\mathcal{I}_w$ of tasks that prevent turbine w to produce electricity when scheduled (i.e., $\mathcal{I}_w = \{i \in \mathcal{I} | b_{wi} = 1 \vee \widetilde{b}_{wi} = 1\}$). The value of $loss_w^{best}$ is computed by running the CP

formulation presented in Section 5.1 on the instance containing only the tasks belonging to $\mathcal{I}_w$. The solution time is most of the time insignificant, but nevertheless we impose a time limit of 1 second. It is noteworthy that, if we find a smaller loss of revenue during the execution of the CPLNS, we update the value of $loss_w^{best}$. Our tests, however, suggest that this is a very rare event. During the optimization, the procedure starts by computing the lost revenue $loss_w^{sol}$ generated by the tasks in $\mathcal{I}_w$ if they are executed as scheduled in the current solution. Notice that the penalties related to postponed tasks are included in the computation of $loss_w^{best}$ and $loss_w^{sol}$. Afterwards, the operator initializes a set $\widetilde{\mathcal{W}}$ with all the turbines of $\mathcal{W}$ and compute the regret $\Delta loss(w) = loss_w^{sol} - loss_w^{best}$ associated with each turbine $w \in \widetilde{\mathcal{W}}$. As long as $\widetilde{\mathcal{W}}$ is not empty and Γ tasks are not removed, the operator removes from $\widetilde{\mathcal{W}}$ the turbine w^* associated with the largest value of $\Delta loss(w)$ and removes from the current solution sol all the scheduled tasks belonging to $\mathcal{I}_{w^*}$.

We work with randomized versions of operators B, D, E, and F to explore the search space more broadly. Indeed, an operator can destroy different parts of the same solution each time it is applied to it. This can then lead to building different solutions and to avoiding being trapped in local optima. Although the randomization strategy we use is relatively simple, we explain it here for the sake of completeness. The strategy is based on the one proposed by Cordeau et al. (2010). Let ϱ_o denote the *randomization factor* of operator o. When selecting tasks for removal, the operator first sorts a list L containing all the tasks using its selection criterion (i.e., largest penalization for operator B, largest similarity with a specified task for operator D, largest regret for operators E and F). The first positions of L contains the tasks that the destroy operator has to target first according to its criterion. Then the operator draws a random number $y \in [0, 1[$ and it selects for removal task i in position $\lfloor y^{\varrho_o} \times |L| \rfloor$ in L (positions in L are indexed from 0). A randomization factor $\varrho_o = 1$ makes the operator completely random, while higher values of ϱ_o make the operators more deterministic. In our experiments we set $\varrho_B = \varrho_D = \varrho_E = \varrho_F = 3$ and we use only the randomized versions of these four operators.

Although it is very simple, Algorithm 3 presents the general structure of a destroy operator used as a subroutine in Algorithm 2.

Algorithm 3: Destroy(o,sol)

Data: a solution sol
a destroy operator o
Result: a set of tasks to remove from sol

$\mathcal{F} \leftarrow \emptyset$
$\mathcal{F} \leftarrow$Apply destroy operator o to sol
return $\mathcal{F}$

5.2.2 Repair operators

We use the CP formulation introduced in Section 5.1 to repair partially destroyed solutions. More specifically, if $\mathcal{F}$ denotes the set of tasks that have been removed, we fix for each task $i \in \mathcal{I} \setminus \mathcal{F}$ the value of the variables $\mathcal{M}_i$, S_i, and $(y_{ri})_{r \in \mathcal{R}_i}$ to their value in the current solution, and we solve the resulting model.

A solution to the CP model is found as soon as the decision variables M_i, S_i, and $(y_{ri})_{r \in \mathcal{R}_i}$ are instantiated for every task $i \in \mathcal{I}$. Therefore, the branching strategy should focus only on these variables. It is worth noting that a CP solver can make meaningful deductions for a task when the domain of the variable related to its executing mode not longer contains the postponement mode. Moreover, fixing the starting time period of a task before knowing its execution mode leads to a weak propagation on the bound of the revenue variable and on the possible starting time periods and execution modes of other tasks. Furthermore, since variables y_{ri} have an impact only on the feasibility of a solution but not on its quality, fixing last these variables (i.e., after having fixed the variables M_i and S_i for each task $i \in \mathcal{I}$) implies that the solver has to explore a large sub-tree before reconsidering a bad decision. Based on these observations we adopt a task-by-task scheduling strategy in which the technicians assignment is made after having chosen an execution mode and a starting time period for the current task.

It is well-known that quickly reaching a good-quality solution increases the efficiency of the search. It is, however, not clear if fixing the execution mode of a task $i \in \mathcal{I}$ (i.e., $\mathcal{M}_i$) to a specific execution mode and then exploring all its potential starting time periods before setting $\mathcal{M}_i$ to another value is the best searching strategy. This observation suggests that simultaneously setting variables M_i and S_i may lead to achieve a greater flexibility during the search. To implement this mechanism, we therefore choose to reuse the concept of plans introduced for the ILP formulations of the problem in Section 4.2. For each task $i \in \mathcal{I}$, we introduce variable $X_i \in \mathcal{P}_i$ that defines the plan selected for task i. We add the constraints (5.17)-(5.18) to couple these variables to the variables $\mathcal{M}_i$ and S_i.

$$\mathcal{M}_i = mode_{X_i} \quad \forall i \in \mathcal{I}, \tag{5.17}$$
$$S_i = start_{X_i} \quad \forall i \in \mathcal{I} \tag{5.18}$$

For a plan $p \in \mathcal{P}$, $mode_p$ is the selected execution mode for the task i_p and $start_p$ represents the starting time period of i_p. In summary, task by task, we first define its execution mode along with its starting time by fixing variable X_i, and we finally assign the required technicians by fixing the variables $(y_{ri})_{r \in \mathcal{R}_i}$.

To reach feasible solutions faster, we maintain arc consistency on constraints (5.17) and (5.18). We also designed customized propagators to try to keep, during the search, the domain of X_i consistent with the availability of the technicians. More specifically, these propagators rely on a comparison between the task requirements and the number of technicians available at each time period of the planning horizon considering the required skills and the daily location-based incompatibilities. They also take into account that technicians have to work on a task from its beginning to its end. For instance, if during a time period t^* no more than 2 technicians mastering a specific skill s^* are available, then for each task i such that $s_i = s^*$ we can remove from the domain of X_i all the plans overlapping t^* and requiring more than 2 technicians.

The most critical part of the procedure is the selection of the next task to be considered by the branching strategy. We select the next task to schedule using a look-ahead regret heuristic that operates as follows. Let $\mathcal{I}^0$ denote the set of tasks which have not yet been processed at the current node of the search. Let also Δf_i^k be the k-th smallest value of the revenue loss that task i can generate when scheduled using one of its possible plans. Our heuristic, *regret-q*, chooses the task to be scheduled as task $i^* = \arg\max_{i \in \mathcal{I}^0} \sum_{k=2}^{k=q} \left(\Delta f_i^k - \Delta f_i^1\right)$. The algorithm computes Δf_i^k according to the values of $\Psi(i,p)$, a function representing the revenue loss if task i uses plan $p \in \mathcal{P}_i$ (i.e., the task is performed in mode $mode_p$ and starts at the beginning of time period $start_p$). Function $\Psi(i,p)$ is computed using functions $\Psi^{day}(i,p)$ and $\Psi^{rest}(i,p)$ which represent, if task i uses plan $p \in \mathcal{P}_i$, the lost revenue during the time periods from $\mathcal{T}$ and during the rest time periods. These functions are defined as follows:

$$\Psi(i,p) = \Psi^{day}(i,p) + \Psi^{rest}(i,p),$$

$$\Psi^{day}(i,p) = \begin{cases} o_p & \text{if } p = p_i^0, \\ \sum_{w \in \mathcal{W} | b_{wi}=1} \sum_{t=start_p}^{t<start_p+d_{imode_p}} g(w,t) & \text{otherwise.} \end{cases},$$

$$\Psi^{rest}(i,p) = \begin{cases} 0 & \text{if } p = p_i^0, \\ \sum\limits_{w \in \mathcal{W} | \widetilde{b}_{wi}=1} \sum\limits_{d \in \mathcal{D}_p} \widetilde{g}(w,d) & \text{otherwise.} \end{cases},$$

Functions $g(w,t)$ and $\widetilde{g}(w,d)$ are defined as:

$$\forall w \in \mathcal{W}, \forall t \in \mathcal{T},\; g(w,t) = \begin{cases} g_w^t & \text{if } t \in Env(F_w^{day}), \\ 0 & \text{otherwise.} \end{cases},$$

$$\forall w \in \mathcal{W}, \forall d \in \mathcal{D},\; \widetilde{g}(w,d) = \begin{cases} \widetilde{g}_w^d & \text{if } d \in Env(F_w^{rest}), \\ 0 & \text{otherwise.} \end{cases},$$

where $Env(Z)$ denotes the set of elements that may belong to the set variable Z in a solution at the current node of the search tree.

Let $Dom(z)$ denote the domain of variable z (i.e., all the possible values that z can take). We have $\Delta f_i^1 = \min\limits_{p \in Dom(X_i)} \Psi(i,p)$. More generally, Δf_i^k is the k-th smallest value of $\Psi(i,p)$. Once task i^* has been selected, it is scheduled using plan $p^* = \arg\min\limits_{p \in Dom(X_{i^*})} \Psi(i^*,p)$.

During our preliminary experiments we observed that sometimes our regret-q heuristic is unable to lead the search to good solutions. It is indeed possible that a task with a small regret at a given point of the search is not chosen to be scheduled, but that this decision leads to a large revenue loss later when exploring the associated subtree. To overcome this potential issue, we designed another branching strategy that selects the task $i^* = \arg\max\limits_{i \in \mathcal{I}_0} \left(\min\limits_{p \in \mathcal{P}_i} \Psi(i,p) \right)$ for which the minimal revenue loss is maximal. Again, once task i^* has been selected, it is scheduled using plan $p^* = \arg\min\limits_{p \in Dom(X_{i^*})} \Psi(i^*,p)$. We refer to this branching strategy as $MaxMinLoss$.

The resources assignment is then done technician by technician as long as the request is not fulfilled. We choose as a priority the compatible technician which is already working during the days that belong to $\mathcal{D}_{p^*}$. Since constraints (5.10) related to the daily location-based incompatibilities are very restrictive, it should be preferable to use technicians that are already working at the same location or at compatible locations. Otherwise, the number of technicians that will be available for other tasks, especially those at incompatible locations, may be drastically restricted. If during the days $d \in \mathcal{D}_{p^*}$ multiple technicians work the same number of time periods, we choose first the technician that could perform the least number of tasks among those remaining. If several technicians can still be selected, we select one randomly.

Exploring the whole neighborhood of a solution is time-consuming; therefore we only allow a certain number ϖ_{max} of backtracks (we set $\varpi_{max} = 200$ in our experiments). Thus, different solutions can be obtained using different branching strategies. Different repair operators are therefore defined using different branching strategies. In our experiments, we use regret-2 and regret-3 branching strategies, as well as a randomized version of $MaxMinLoss$, where the probability of selecting a task is inversely proportional to the minimal revenue loss it generates at this point of the search.

Algorithm 4 presents the general structure of a repair operator used as a subroutine in Algorithm 2.

5.2.3 Acceptance criteria

In our experiments, we tested the elitist strategy and the Metropolis criterion to accept new solutions. We also tested a mix of them: we apply an elitist strategy during the first k iterations, and then we activate the Metropolis criterion. We based our choice in two observations. First, using the elitist strategy, the search is often trapped in local optima after a certain amount of iterations, and then it struggles to improve the solution. Second, as we do not ensure that our algorithm starts from a good-quality solution, reaching a good solution can be time-consuming.

Algorithm 4: Repair(o,$\mathcal{F}$,s)

Data: a solution sol
a set $\mathcal{F}$ of tasks
a repair operator o (branching strategy)
Result: a new solution sol'
foreach $i \in \mathcal{I} \setminus \mathcal{F}$ **do**
  Fix the values of $\mathcal{M}_i$, S_i, $(y_{ri})_{r \in \mathcal{R}_i}$ as in solution sol in the CP model
end
Solve the CP model applying repair operator o, yielding sol'
return sol'

In our experiments, k is set to 250 and T to 0.9975. The initial temperature is fixed to $-\frac{0.25}{ln 0.5} f(sol_0)$ where $f(sol_0)$ is the value of the objective function of the initial solution sol_0. Therefore, in the first iteration our approach accepts solutions that are 2.5% worse than the current solution with a probability of 0.5.

5.3 Computational experiments

We implemented our algorithms using *Java 8 (JVM 1.8.0.25)*. We rely on *Choco 3.3.1* for solving the CP formulation (see Prud'homme et al. (2014)). We ran our experiments on a Linux 64 bit-machine, with an Intel(R) Xeon(R) X5675 (3.07Ghz) and 10GB of RAM.

In order to assess the quality of our results, we compute the gap with respect to the optimal solution when it is known, or to the best upper bound we found among all the tests reported in this thesis manuscript.

5.3.1 CP formulation

Figure 5.1 summarizes the global results found solving the CP model with our regret-2 branching strategy (BS) and with a randomized version of the latter coupled to a geometrical restart policy (we restart the search from the root node) based on the number of backtracks (BS+restart). Notice that the default branching strategy of the solver is most of the time unable to provide a feasible solution to the problem after several minutes. This is somehow expected since we work with many different kinds of variables. On the opposite, a solution is always quickly found using our regret branching strategy. Guaranteeing feasibility is, however, relatively easy since tasks can be postponed. Coupling our branching strategy with a restart policy give the best results as the average gap is improved approximately by 6% on testbed G1. Jointly using a randomized branching strategy with a restart policy allows us to explore different parts of the search tree which increases the likelihood of finding better solution. However, we observe that the initial solutions are little improved during the search whatever solution strategy we use. It seems that the CP model is facing some symmetry issues, especially on the technicians assignment. This drawback is not overcome with our restart policy.

Table 5.1 reports additional results when solving the CP model with the BS+restart configuration. It shows in the different columns the relative average mean gap [1] (Gap) and the mean percentage of tasks scheduled (i.e., not-posponed) in the solution [2] ($\%S$) with 15, 30, 60, 180, and 300 seconds of CPU time limit. Appendix C.1 presents the detailed results when solving the CP model with the first solution strategy (BS). Solving the CP model provides near optimal solutions for instances with 2 time periods per day and for type B instances. Moreover, we can notice that the CP formulation gives better overall results on the large-sized instances than the ILP formulations presented in Chapter 4.

1. average of the mean gap found for each instance over 3 runs
2. average of the mean percentage of tasks scheduled in the solution found for each instance over 3 runs

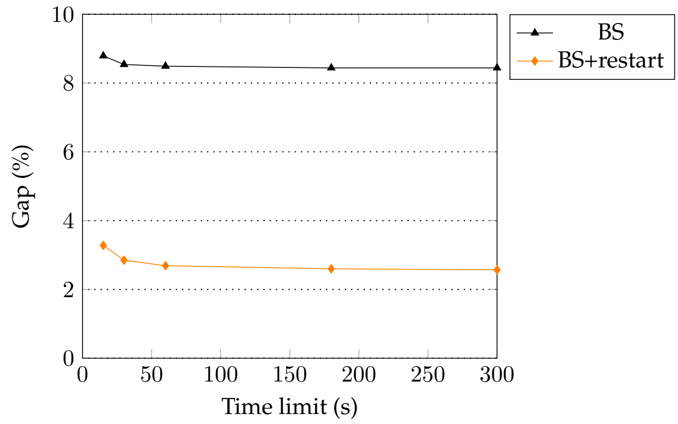


Figure 5.1 – Average computational results on testbed G1 when solving the CP formulation according to the solution strategy (average over 3 runs)

Table 5.1 – Computational results on testbed G1 when solving of the CP formulation (BS+restart - average over 3 runs)

Family	15s		30s		60s		180s		300s	
	Gap	**%S**	**Gap**	**%S**	**Gap**	**%S**	**Gap**	**%S**	**Gap**	**%S**
10_2_1_20_A	1.6%	98%	1.3%	98%	1.3%	98%	1.3%	98%	1.3%	98%
10_2_1_20_B	0.5%	100%	0.5%	100%	0.4%	100%	0.4%	100%	0.4%	100%
10_2_1_40_A	2.9%	99%	2.5%	99%	2.4%	99%	2.4%	99%	2.4%	99%
10_2_1_40_B	0.5%	100%	0.4%	100%	0.4%	100%	0.4%	100%	0.4%	100%
10_2_3_20_A	2.4%	96%	2.0%	97%	2.0%	97%	2.0%	97%	2.0%	97%
10_2_3_20_B	0.3%	100%	0.3%	100%	0.3%	100%	0.3%	100%	0.3%	100%
10_2_3_40_A	3.5%	98%	2.4%	99%	2.4%	99%	2.3%	99%	1.9%	99%
10_2_3_40_B	1.0%	100%	0.8%	100%	0.8%	100%	0.8%	100%	0.8%	100%
20_2_1_40_A	2.8%	98%	2.7%	98%	2.5%	99%	2.5%	99%	2.5%	99%
20_2_1_40_B	0.4%	100%	0.3%	100%	0.3%	100%	0.3%	100%	0.3%	100%
20_2_1_80_A	4.6%	98%	3.8%	98%	3.2%	99%	3.2%	99%	3.2%	99%
20_2_1_80_B	0.3%	100%	0.2%	100%	0.2%	100%	0.2%	100%	0.2%	100%
20_2_3_40_A	1.7%	99%	1.6%	99%	1.6%	99%	1.6%	99%	1.6%	99%
20_2_3_40_B	0.2%	100%	0.2%	100%	0.2%	100%	0.2%	100%	0.2%	100%
20_2_3_80_A	2.1%	99%	1.7%	100%	1.2%	100%	1.2%	100%	1.1%	100%
20_2_3_80_B	0.3%	100%	0.3%	100%	0.3%	100%	0.2%	100%	0.2%	100%
20_4_1_20_A	2.6%	95%	2.5%	95%	2.4%	95%	2.2%	95%	2.1%	95%
20_4_1_20_B	1.7%	99%	1.2%	100%	1.0%	100%	0.9%	100%	0.9%	100%
20_4_1_40_A	10.9%	93%	10.1%	93%	9.9%	93%	9.6%	93%	9.5%	93%
20_4_1_40_B	4.3%	98%	3.4%	99%	2.9%	99%	2.9%	99%	2.9%	99%
20_4_3_20_A	6.8%	94%	6.4%	94%	6.3%	94%	5.7%	94%	5.7%	94%
20_4_3_20_B	1.7%	99%	1.7%	99%	1.7%	99%	1.7%	99%	1.7%	99%
20_4_3_40_A	8.5%	93%	7.0%	94%	6.9%	94%	6.9%	94%	6.9%	94%
20_4_3_40_B	3.2%	98%	2.2%	99%	1.9%	99%	1.8%	99%	1.8%	99%
40_4_1_40_A	8.1%	94%	7.8%	94%	7.8%	94%	7.8%	94%	7.8%	94%
40_4_1_40_B	1.3%	100%	1.2%	100%	0.8%	100%	0.7%	100%	0.5%	100%
40_4_1_80_A	11.5%	93%	10.6%	94%	9.7%	94%	8.8%	95%	8.8%	95%
40_4_1_80_B	1.5%	100%	0.6%	100%	0.5%	100%	0.5%	100%	0.5%	100%
40_4_3_40_A	6.2%	96%	5.7%	96%	5.5%	96%	5.5%	96%	5.4%	96%
40_4_3_40_B	1.3%	100%	1.0%	100%	0.6%	100%	0.6%	100%	0.6%	100%
40_4_3_80_A	9.3%	94%	8.1%	95%	8.0%	95%	7.9%	95%	7.9%	95%
40_4_3_80_B	1.1%	100%	0.5%	100%	0.5%	100%	0.4%	100%	0.4%	100%

Characteristic		15s		30s		60s		180s		300s	
		Gap	**%S**	**Gap**	**%S**	**Gap**	**%S**	**Gap**	**%S**	**Gap**	**%S**
$\lvert\mathcal{S}\rvert =$	1	3.5%	98%	3.1%	98%	2.9%	98%	2.7%	98%	2.7%	98%
	3	3.1%	98%	2.6%	98%	2.5%	98%	2.4%	98%	2.4%	98%
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	1.6%	99%	1.3%	99%	1.2%	99%	1.2%	99%	1.2%	99%
	4	5.0%	96%	4.4%	97%	4.2%	97%	4.0%	97%	4.0%	97%
Type =	*A*	5.3%	96%	4.8%	96%	4.6%	96%	4.4%	96%	4.4%	96%
	B	1.2%	100%	0.9%	100%	0.8%	100%	0.8%	100%	0.8%	100%
All		3.3%	98%	2.9%	98%	2.7%	98%	2.6%	98%	2.6%	98%

5.3.2 CPLNS

Previous results show some limitations of solving the CP model. They demonstrate the relevance of a CPLNS approach that may be more suitable than a restart policy to escape local optima. For this latter method, we also imposed a time limit as the stopping criterion. Since the neighborhoods are partially randomized, we launched the algorithm ten times for each instance.

Our first experiment aimed to select the best acceptance criterion for our CPLNS. To achieve our goal, we ran our algorithm with three different solution acceptance criteria: elitism (El), Metropolis (MT), and both (El+MT) and three different time limits: 60, 180, and 300 seconds. Table 5.2 shows that coupling an elitist strategy with the Metropolis acceptance criterion leads to the smaller average gap independently of the time limit. We therefore use this acceptance criterion in the remainder of our experiments.

Table 5.2 – Average computational results on testbed G1 according to the solution acceptance criterion

Time limit	Average gap		
	60s	**180s**	**300s**
El	1.54%	1.33%	1.26%
MT	1.58%	1.34%	1.25%
El + MT	1.54%	1.30%	1.21%

We now discuss more thoroughly the performance of the CPLNS algorithm. Table 5.3 reports the results delivered by our CPLNS for each family of instances. It shows in the different columns the relative average mean gap [3] (Gap) and the mean percentage of tasks scheduled in the solution [4] ($\%S$) running with a 15, 30, 60, 180, and 300 seconds of CPU time limit. These experiments aim to enable a decision-maker to define a CPU time limit according to the trade-off between solution time and quality of the results he or she is interested in.

Since the gap is computed with respect to upper bounds for some type A instances of testbed G1, assessing the intrinsic quality of the CPLNS using only the gap is sometimes not conclusive enough. However, the overall average gaps of 1.2% after 5 minutes for testbed G1 show the effectiveness of our approach. We may expect to be closer to the optimal solutions for the large-sized instances. For testbed G1, the algorithm provides near optimal solutions for all the type B instances, but the performance is slightly inferior for the type A instances in which the number of time periods per day is equal to 4. This last observation can be explained by the fact that the number of plans and thus the model to be considered by the CP model when repairing the solution is larger. Since we only allow a limited number of backtracks, the quality of the first decisions taken in our branching strategies strongly impacts the capacity of the algorithm to improve the current solutions. The algorithm may then sometimes fail building better solutions with the CP model, although it could be have been possible if it explored the whole search space.

3. average of the mean gap found for each instance over 10 runs
4. average of the mean percentage of tasks scheduled in the solution found for each instance over 10 runs

Table 5.3 – Computational results on testbed G1 for the CPLNS (average over 10 runs)

Family	15s		30s		60s		180s		300s	
	Gap	%S	Gap	%S	Gap	%S	Gap	%S	Gap	%S
10_2_1_20_A	1.0%	98%	0.9%	98%	0.8%	98%	0.8%	98%	0.8%	98%
10_2_1_20_B	0.1%	100%	0.0%	100%	0.0%	100%	0.0%	100%	0.0%	100%
10_2_1_40_A	1.6%	99%	1.3%	99%	1.0%	100%	0.5%	100%	0.4%	100%
10_2_1_40_B	0.1%	100%	0.0%	100%	0.0%	100%	0.0%	100%	0.0%	100%
10_2_3_20_A	1.1%	98%	0.9%	99%	0.8%	99%	0.7%	99%	0.6%	99%
10_2_3_20_B	0.0%	100%	0.0%	100%	0.0%	100%	0.0%	100%	0.0%	100%
10_2_3_40_A	2.0%	99%	1.8%	99%	1.5%	99%	1.1%	99%	1.0%	99%
10_2_3_40_B	0.1%	100%	0.1%	100%	0.1%	100%	0.1%	100%	0.0%	100%
20_2_1_40_A	2.0%	98%	1.8%	99%	1.4%	99%	1.0%	99%	0.9%	99%
20_2_1_40_B	0.1%	100%	0.1%	100%	0.1%	100%	0.0%	100%	0.0%	100%
20_2_1_80_A	3.8%	98%	3.5%	98%	3.1%	98%	2.5%	99%	2.2%	99%
20_2_1_80_B	0.1%	100%	0.1%	100%	0.1%	100%	0.1%	100%	0.1%	100%
20_2_3_40_A	1.2%	99%	1.1%	99%	0.9%	99%	0.6%	100%	0.6%	100%
20_2_3_40_B	0.0%	100%	0.0%	100%	0.0%	100%	0.0%	100%	0.0%	100%
20_2_3_80_A	0.8%	100%	0.6%	100%	0.5%	100%	0.4%	100%	0.3%	100%
20_2_3_80_B	0.1%	100%	0.1%	100%	0.1%	100%	0.1%	100%	0.1%	100%
20_4_1_20_A	1.3%	95%	1.2%	95%	1.1%	95%	1.1%	95%	1.0%	95%
20_4_1_20_B	0.5%	100%	0.2%	100%	0.1%	100%	0.1%	100%	0.1%	100%
20_4_1_40_A	7.2%	93%	6.5%	94%	6.2%	94%	5.4%	94%	5.3%	94%
20_4_1_40_B	1.6%	99%	1.3%	99%	1.2%	99%	0.5%	100%	0.2%	100%
20_4_3_20_A	4.1%	95%	3.5%	95%	3.1%	95%	2.6%	96%	2.0%	96%
20_4_3_20_B	0.6%	100%	0.5%	100%	0.3%	100%	0.2%	100%	0.2%	100%
20_4_3_40_A	5.7%	94%	5.1%	94%	4.7%	95%	4.0%	95%	3.9%	95%
20_4_3_40_B	0.9%	100%	0.8%	100%	0.8%	100%	0.6%	100%	0.5%	100%
40_4_1_40_A	5.8%	95%	5.0%	96%	4.7%	96%	4.5%	96%	4.4%	96%
40_4_1_40_B	0.3%	100%	0.3%	100%	0.2%	100%	0.2%	100%	0.1%	100%
40_4_1_80_A	6.2%	96%	6.0%	96%	5.8%	96%	5.4%	96%	5.4%	96%
40_4_1_80_B	0.4%	100%	0.3%	100%	0.3%	100%	0.2%	100%	0.2%	100%
40_4_3_40_A	4.3%	97%	4.0%	97%	3.8%	97%	3.1%	98%	2.8%	98%
40_4_3_40_B	0.3%	100%	0.3%	100%	0.2%	100%	0.2%	100%	0.1%	100%
40_4_3_80_A	7.0%	95%	6.7%	95%	6.3%	95%	5.9%	95%	5.6%	96%
40_4_3_80_B	0.4%	100%	0.3%	100%	0.2%	100%	0.2%	100%	0.2%	100%

Characteristic		15s		30s		60s		180s		300s	
		Gap	%S	Gap	%S	Gap	%S	Gap	%S	Gap	%S
$\lvert\mathcal{S}\rvert =$	1	2.0%	98%	1.8%	98%	1.6%	98%	1.4%	99%	1.3%	99%
	3	1.8%	98%	1.6%	99%	1.5%	99%	1.2%	99%	1.1%	99%
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	0.9%	99%	0.8%	99%	0.7%	99%	0.5%	100%	0.4%	100%
	4	2.9%	97%	2.6%	97%	2.4%	98%	2.1%	98%	2.0%	98%
Type =	A	3.4%	97%	3.1%	97%	2.9%	97%	2.5%	97%	2.3%	97%
	B	0.4%	100%	0.3%	100%	0.2%	100%	0.2%	100%	0.1%	100%
All		1.9%	98%	1.7%	98%	1.5%	99%	1.3%	99%	1.2%	99%

In Table 5.4, we report in the different columns the relative average mean gap (Mean), the average best gap [5](Best), and the average worst gap [6](Worst) for the CPLNS with 300 seconds of CPU time limit (detailed results are available in Appendix C.2). The CPLNS seems to have a suitable stability for testbed G1 as on average the difference between the best and the worst solution found over the 10 runs is a reduced 0.68%.

Table 5.4 – Aggregated computational results on testbed G1 for the CPLNS with a time limit of 300 seconds (10 runs)

Characteristic		Mean	Best	Worst
$\|\mathcal{S}\| =$	1	1.32%	1.04%	1.65%
	3	1.12%	0.76%	1.50%
$\frac{\|\mathcal{T}\|}{\|\mathcal{D}\|} =$	2	0.45%	0.25%	0.66%
	4	2.00%	1.55%	2.50%
Type $=$	A	2.33%	1.75%	2.93%
	B	0.12%	0.05%	0.22%
All		1.22%	0.90%	1.58%

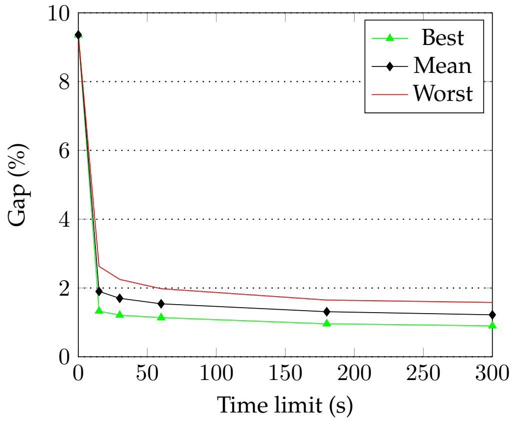


Figure 5.2 – Average computational results on testbed G1 for the CPLNS according to the time limit

5.4 Conclusions

In this chapter, we have proposed a mathematical formulation of the problem based on CP. Computational results indicate that the CP model produces high quality solutions for small-sized instances. However, it does not yield very good results, in general, for realistic instances. Its performance actually seems to be affected by symmetry issues, especially on the technicians assignment. To provide an alternative solution approach, we have developed a CPLNS. We have successfully adapted some destroy operators to this new problem and proposed some new ones. Moreover, we have designed several branching strategies to effectively repair solutions solving a CP model with fixed variables. We have also introduced and demonstrated the relevance of a new acceptance criterion combining elitism and Metropolis. If we fix a reasonable CPU time limit of 5 minutes, the CPLNS shows on testbed G1 an average gap of 1.2% with respect to the optimal solutions if known, or to the best upper bounds otherwise. It provides near optimal solutions when the availability of the technicians is not binding, whereas, when this availability is scarce, the gap increases as the problem size (number of tasks and number of time periods per day) grows. Nonetheless, the computational results demonstrate the efficiency of the proposed method.

5. average of the best (minimal) gap found for each instance over 10 runs
6. average of the worst (maximal) gap found for each instance over 10 runs

5.5 Industrial prototype for WPred

We present here our specific collaboration with the canadian company WPred. This company is specialized in the supply of weather forecasts and, more specifically, on power production forecasts for wind and solar energy. Wind farm operators as well as wind-turbines maintenance companies are among the main customers of WPred. Indeed, this company proposes a calendar application to help its customers to find the best windows of opportunity for their maintenance operations. This application is essentially a communication tool where companies can manually input a maintenance plan and work assignments with the help of a weather and production predictions feed. A study by WPred estimates to 1% the increase of the CF using this online maintenance calendar overlayed with a weather predictions feed rather than using Excel tables and post-it notes as it still sometimes occurs today in this industry. Adding an optimization engine to this online calendar application is aimed to assist maintenance schedulers in taking well-advised decisions. Indeed, when manually scheduling the maintenance it is very difficult for them to take into account all the fined-grained data. Moreover, the combinatorial nature of the problem makes it virtually impossible to build the best maintenance plan according to what they consider relevant. As part of our collaboration with WPred, we developed a prototype for this optimization tool.

Let us define how this optimization tool is going to interact with the online calendar application. First of all, this interaction aims to take advantage of the knowledge of the schedulers about maintenance scheduling. Indeed, they have a strong expertise and know some information that cannot be embedded in the optimizer (e.g., preferences to assign specific work to some technicians, some technicians may work faster on particular tasks, adverse weather conditions may quickly happen in some wind farms, there must be not too many *holes* in the resulting maintenance plan, the capacity of the resulting planning to absorb delays is not adequate). Therefore, WPred and its customers think the maintenance scheduling optimization as an iterative process. This process is summarized in Figure 5.3. First, the schedulers define the time horizon, the set of tasks that need to be scheduled and they also set all the others parameters that are required by the optimization tool. Then, they launch the optimization. When they stop it (possible at any time), one or multiple maintenance plans are proposed to the schedulers according to the selected settings. If one of this plan satisfies the schedulers then they still can slightly modify it before accepting it as the baseline maintenance plan. Otherwise, they can fix the schedule of some tasks according to their own expertise, or they can refine their time windows according to what they consider to be the best time periods to schedule the maintenance. They can also assign technicians to some tasks while leaving them unscheduled. Then they call again the optimization tool as long as they are not satisfied or they distinguish the opportunity to improve the current plan (mainly based on production considerations).

The optimization tool needs to take into consideration this intended use as well as the restriction imposed by WPred to work with free optimization software (WPred cannot afford any licensing fees). This restricts a bit the solution methods we can implement (although they are nowadays more and more free optimization software). We therefore came up with the idea to use the CPLNS. This present multiple advantages. First, we can implement the CPLNS with Choco, a free (and open-source) library that enables working with CP (notice that this library requires the use of the Java programming language). Second, with an eye on the future, the method is flexible, in the sense that new constraints can be easily added. Third, the method is clearly understandable by practitioners with limited OR knowledge.

A short time has also been dedicated to the integration of the optimization tool in WPred existing product. Figure 5.4 actually shows in the solid line boxes the work that we took care of.

As previously mentioned, the problem considered in this work for WPred slightly differs from the one defined in Chapter 3. Let us now present the specificities taken into account in this particular case. We list them below (for the sake of readability, we keep the same notations):

— In this new problem, we do not directly take into account skills for technicians. Actually, each wind farm is assigned to a service point which is in charge of the maintenance operations. A technician usually belongs to one service point (seen then as his or her home depot), but it may

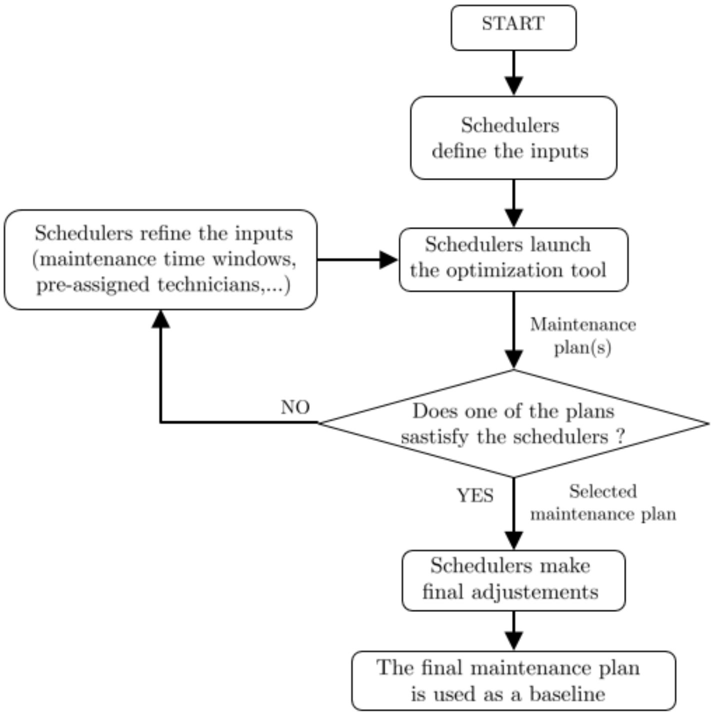


Figure 5.3 – Flow chart of the iterative process intended to schedule the maintenance

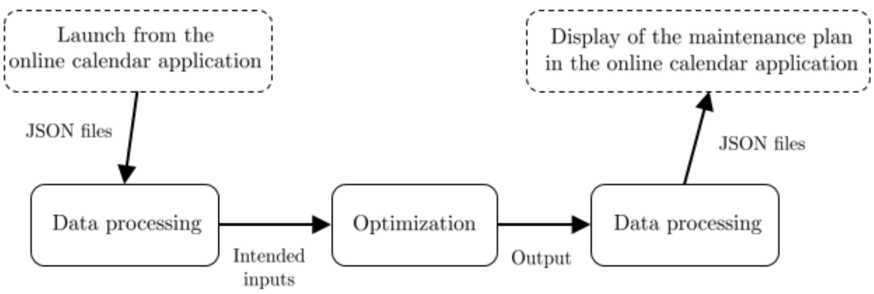


Figure 5.4 – A brief overview of the integration of the optimization tool into the online calendar application fo WPred

happen that some technicians belong to multiple service points. This still underlie the notion of skills (clearly we create as many skill as there are combination of service points considered for the technicians).

— Before the optimization, we allow technician-to-task (pre-)assignments. These assignments may correspond to preferences or some hidden required skill(s). This translates into the fixing of some variables y in the CP model. More specifically, we set $y_{ri} = 1$ if technician r should be assigned to task i (regardless to its schedule). In that case, the task i cannot be postponed anymore and can only be scheduled when the technician r is available.

— Wind farms may belong to different customers. According to the relative importance of these customers, one should prioritize the maintenance operations of some them. This means to avoid postponing these tasks and to prefer scheduling them when the wind is at its lowest. The importance of each customer is summarized in a positive value from a finite predefined set. The easiest way to take this specificity into account is by weighting the revenue and the penalties in the objective function.

— The optimization tool may have to schedule known corrective maintenance operations even if, currently, a pool of technicians is continuously kept to only deal with these unexpected breakdowns. Indeed, if the wind is too low to produce electricity for many consecutive days, there is a flexibility in the scheduling of these tasks. For a task i, we denote $\ddot{b}_{wi} = 1$ if and only if the task is a corrective task to perform on turbine w. We remove constraints (5.12) and (5.14) and we add the following constraints to the CP model:

$$t \leq C_i \Rightarrow t \notin F_w^{day} \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I} \; s.t. \; \ddot{b}_{wi} = 1, \forall t \in \mathcal{T}, \tag{5.19}$$

$$\left(t \notin \bigcup_{i \in \mathcal{I} | b_{wi} = 1} E_i \right) \wedge \left(\bigwedge_{i \in \mathcal{I} | \ddot{b}_{wi} = 1} (t > C_i) \right) \Rightarrow t \in F_w^{day} \quad \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \tag{5.20}$$

$$(t_d^{rest} + 1) \leq C_i \Rightarrow d \notin F_w^{rest} \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \; s.t. \; \ddot{b}_{wi} = 1, \forall d \in \mathcal{D}, \tag{5.21}$$

$$\left(\bigwedge_{i \in \mathcal{I} | \widetilde{b}_{wi} = 1} (\{t_d^{rest}, t_d^{rest} + 1\} \not\subseteq E_i) \right) \wedge \left(\bigwedge_{i \in \mathcal{I} | \ddot{b}_{wi} = 1} (t_d^{rest} + 1 > C_i) \right) \Rightarrow d \in F_w^{rest} \quad \forall w \in \mathcal{W}, \forall d \in \mathcal{D} \tag{5.22}$$

Constraints (5.19) and (5.21) state that a turbine w is unavailable during a time period as soon as a corrective maintenance task to perform on w is still not completed. Constraints (5.20) and (5.22) state that a turbine is available to produce electricity during a time period if and only if no preventive tasks requiring its shutdown are scheduled at the same time and if all the corrective tasks are completed.

For each technician $r \in \mathcal{R}$, the vector ρ_r is set according to: i) training and/or personal holiday times already planned, ii) some work to perform along with external companies and that is already fixed in time, and iii) an assignment to a task started before the beginning of the planning horizon (for the cases ii) and iii) the locations l_r^t for each unavailability time period t are set accordingly). For each task i, the vector $\ddot{\vartheta}_i$ is fixed according to the weather predictions feed of WPred, and the maximal wind speed allowed for maintenance operations is fixed according to the customer owning the wind farm l_i.

To assess the efficiency of the optimization tool, we completed along with WPred a short proof of concept based on data generated by WPred (according to some private customer data). According to these preliminary tests, the addition of the optimization engine to the online calendar is thought to increase the CF by at least 1% with respect to the current situation. More extensive testing by their clients is expected to start soon.

This work (more specifically the resulting product) has been presented at the Canadian Wind Energy Association annual conference (CANWEA2015). This collaboration has met the expectations of WPred.

Chapter 6

A branch-and-check approach to solve the deterministic problem

The research reported in this chapter has been wrapped up on a journal article submitted in Computers & Operations Research and currently under first round of review.

Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2016). A branch-and-check approach to solve a wind turbine maintenance scheduling problem. Under review.

Chapter 5 presents a heuristic method to tackle the wind turbine maintenance scheduling problem introduced in Chapter 3. We now aim to develop an efficient exact approach to that problem. Chapter 4 shows that directly solving integer linear programming (ILP) formulations is usually computationally intractable as these formulations are very large. Another way to address this complex combinatorial problem may come from decomposition techniques that allow to decouple a large scale problem into several problems that are easier to solve. This chapter therefore proposes an exact solution method based on this idea. The problem is decomposed into a task scheduling problem and a technician-to-task assignment sub-problem, and solved using a branch-and-check (B&C) approach. More specifically, while solving the task scheduling problem, we discard, by means of cuts all along the branch-and-bound tree, selections of plans that cannot be performed by the technicians. Section 6.1 describes the decomposition of the problem. In addition to the generic Benders cuts, Section 6.2 introduces problem-specific cuts. Section 6.3 discusses the general scheme of the method. Section 6.4 reports computational results on testbed G1. Finally, Section 6.5 presents our conclusions on the efficiency of the method.

6.1 Problem decomposition

The exact approach presented in this chapter takes advantage of the intrinsic decomposition of the problem into a task scheduling problem and a technician-to-task assignment sub-problem. The task scheduling problem consists in selecting a plan for each task in order to maximize the difference between revenue generated by the wind electricity production and the postponing penalties. In this problem, technician considerations have been partially dropped. If we assume a fixed selection of plans, the technician-to-task assignment sub-problem (hereafter occasionally referred to simply as the sub-problem) checks if the technician requests can be satisfied while respecting the daily location-based incompatibilities and coping with individual resource availability time periods. The aim of our approach is thus to design a coordination procedure between these two problems. Note that an optimal solution to the scheduling problem leading to a feasible technician-to-task assignment sub-problem is optimal for the whole problem. The ILP formulations (presented below) of these two problems are essentially inspired by formulations $[P2]$ and $[P3]$ presented in Chapter 4 (we refer the reader to this chapter for the definition of the variables and the explanation of the constraints).

First, let us introduce the scheduling problem. An initial ILP formulation $[ShP_1]$ of this problem reads:

$$[ShP_1] \quad \max \sum_{w\in\mathcal{W}} \left(\sum_{t\in\mathcal{T}} g_w^t f_w^t + \sum_{d\in\mathcal{D}} \widetilde{g}_w^d \widetilde{f}_w^d \right) - \sum_{p\in\mathcal{P}} o_p x_p \tag{6.1}$$

subject to:

$$\sum_{p\in\mathcal{P}_i} x_p = 1 \quad \forall i \in \mathcal{I}, \tag{6.2}$$

$$\sum_{p\in\mathcal{P}_i} a_p^t x_p \le \vartheta_{i_p}^t \ddot{\vartheta}_{i_p}^t \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{6.3}$$

$$\sum_{i\in\mathcal{B}} \sum_{p\in\mathcal{P}_i} a_p^t x_p \le 1 \quad \forall \mathcal{B} \in ov\left(\mathcal{I}\right), \forall t \in \mathcal{T}, \tag{6.4}$$

$$f_w^t + \sum_{p\in\mathcal{P}_i} b_{wp} a_p^t x_p \le 1 \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{6.5}$$

$$\widetilde{f}_w^d + \sum_{p\in\mathcal{P}_i} \widetilde{b}_{wp} \widetilde{a}_p^d x_p \le 1 \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall d \in \mathcal{D}, \tag{6.6}$$

$$\sum_{i\in\mathcal{I}|s_i\in\bar{\mathcal{S}}} \sum_{p\in\mathcal{P}_i} a_p^t q_p x_p \le |R_{\bar{\mathcal{S}}}^t| \quad \forall t \in \mathcal{T}, \forall \bar{\mathcal{S}} \subseteq \mathcal{S}, \tag{6.7}$$

$$x_p \in \{0,1\} \quad \forall p \in \mathcal{P}, \tag{6.8}$$

$$f_w^t \in \{0,1\} \quad \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \tag{6.9}$$

$$\widetilde{f}_w^d \in \{0,1\} \quad \forall w \in \mathcal{W}, \forall d \in \mathcal{D}, \tag{6.10}$$

Let us now assume a fixed selection $(\bar{x}_p)_{p\in\mathcal{P}}$ of plans (hereafter referred to simply as $\bar{x}$) solution to $[ShP_1]$. An ILP formulation $[SP_2(\bar{x})]$ of the technician-to-task assignment sub-problem reads:

$$[SP_2(\bar{x})] \quad \min \sum_{h\in\mathcal{H}} \theta_h \tag{6.11}$$

subject to:

$$\sum_{r\in\mathcal{R}_h} y_{rh} + \theta_h = \sum_{p\in P_h} q_p \bar{x}_p \quad \forall h \in \mathcal{H}, \tag{6.12}$$

$$\sum_{\substack{h\in H \\ s.t.\ r\in\mathcal{R}_h}} y_{rh} \le 1 \quad \forall r \in \mathcal{R}, \forall H \in \mathcal{C}^{max}\left(G\right), \tag{6.13}$$

$$\theta_h \ge 0 \quad \forall h \in \mathcal{H}, \tag{6.14}$$

$$y_{rh} \in \{0,1\} \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h \tag{6.15}$$

We introduce slack variables $(\theta_h)_{h\in\mathcal{H}}$ for the technician requirements constraints (6.12). The unavailability time periods of each technician are respected by definition of the set $\mathcal{R}_h$. The clique constraints (6.13) ensure that the technician assignments comply with the daily location-based incompatibilities. More specifically, set $\mathcal{C}^{max}\left(G\right)$ contains to all the maximal cliques in a graph G where each vertex represents a pattern in $\mathcal{H}$, and there exists an edge between two vertices if the underlying patterns h and h' cannot be visited by the same technician. More precisely, this edge exists if the following clause holds:

$$(S_{h'} \le C_h \ \wedge \ S_h \le C_{h'}) \ \vee \ \left((\exists d \in \mathcal{D}, \mathcal{T}_d \cap \{S_h, ..., C_h\} \neq \emptyset \wedge \mathcal{T}_d \cap \{S_{h'}, ..., C_{h'}\} \neq \emptyset) \wedge \sigma_{l_h l_{h'}} = 0\right)$$

One can visualize graph G as an extended version of an interval graph. To define the sub-problem for a given solution $\bar{x}$ (and therefore to compute all the maximal cliques), we only need to consider the patterns of set $\mathcal{H}(\bar{x}) = \{h \in \mathcal{H} \mid \sum_{p \in P_h} q_p \bar{x}_p > 0\}$. This allows to significantly reduce the number of variables in $[SP_2(\bar{x})]$ and the number of clique constraints (6.13) as we consider a sub-graph of G.

It is worth noting that the technician-to-task assignment sub-problem is NP-complete. This can be proved by its equivalence to a L-coloring problem. Nonetheless, under certain assumptions, the sub-problem becomes solvable in polynomial time. For the sake of clarity, we do not present any details here. We refer the reader to Section 6.6 at the end of this chapter for a thorough discussion on the complexity of the sub-problem.

Notice also that the cumulative scheduling constraints (6.7) in $[ShP_1]$ help to speed-up the convergence of a coordination procedure between the two problems. Indeed, solving the task scheduling problem without any information regarding the availability of the technicians would result in selection of plans that would unlikely lead to a feasible technician-to-task assignment sub-problem.

It is fairly easy to observe that $[SP_2(\bar{x})]$ always admits a feasible solution thanks to the slack variables $(\theta_h)_{h \in \mathcal{H}}$. However, we can conclude that the technician-to-task assignment sub-problem is feasible only if the value of the optimal solution to $[SP_2(\bar{x})]$ is equal to zero.

Let $[SP_2^{LR}(\bar{x})]$ be the linear relaxation of formulation $[SP_2(\bar{x})]$. More precisely, constraints (6.15) of $[SP_2(\bar{x})]$ are substituted in $[SP_2^{LR}(\bar{x})]$ by the following constraints:

$$y_{rh} \leq 1 \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h, \tag{6.16}$$
$$y_{rh} \geq 0 \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h, \tag{6.17}$$

Because the constraint matrix of $[SP_2^{LR}(\bar{x})]$ is not totally unimodular, integrity constraints (6.15) on variables y_{rh} cannot be relaxed while ensuring they will be satisfied by any optimal solution to problem $[SP_2(\bar{x})]$.

When the cost of a solution to $[SP_2(\bar{x})]$ is strictly positive (i.e., the technician-to-task assignment sub-problem is infeasible), we therefore use a *combinatorial Benders cut* as introduced in Section 2.3. Using the binary variables $(x_p)_{p \in \mathcal{P}}$, we can define such a cut as follows:

$$\sum_{p \in \mathcal{P} | \bar{x}_p = 1} x_p + \sum_{p \in \mathcal{P} | \bar{x}_p = 0} (1 - x_p) \geq 1 \tag{6.18}$$

Clearly, this cut states that at least one of the variables related to the selection of plans in the scheduling problem $[ShP_1]$ must change value with respect to $\bar{x}$. It is also known as a *no-good* cut. Observing that a solution contains always $|\mathcal{I}|$ non-zero variables x_p since exactly one plan has to be selected per task, we can then replace inequality (6.18) by the following cover inequality (denoting $\mathcal{P}(\bar{x}) = \{p \in \mathcal{P} | \bar{x}_p = 1\}$):

$$\sum_{p \in \mathcal{P}(\bar{x})} x_p \leq |\mathcal{I}| - 1, \tag{6.19}$$

Hereafter, we refer to the combinatorial Benders cuts (6.19) as CB cuts.

Since the feasible region of our problem is bounded, the number of integer points satisfying all the constraints of $[ShP_1]$ is finite and, thus, the same holds for the number of CB cuts. Let us denote $\bar{\mathcal{F}}$ the set of all solutions $\bar{x}$ to $[ShP_1]$ that lead to an infeasible technician-to-task assignment sub-problem. The whole maintenance scheduling problem can therefore be reformulated as the following master problem $[P]$:

$$[P] \quad \max \sum_{w \in \mathcal{W}} \left(\sum_{t \in \mathcal{T}} g_w^t f_w^t + \sum_{d \in \mathcal{D}} \widetilde{g}_w^d \widetilde{f}_w^d \right) - \sum_{p \in \mathcal{P}} o_p x_p$$

subject to:

(6.2), (6.3), (6.4), (6.5), (6.6), (6.7), (6.8), (6.9), (6.10)

$$\sum_{p \in \mathcal{P}(\bar{x})} x_p \leq |\mathcal{I}| - 1 \quad \forall \bar{x} \in \bar{\mathcal{F}} \tag{6.19}$$

In the remainder of the document, we denote $[RMP]$ a restricted master problem of problem $[P]$ that contains none or only a small subset of constraints (6.19).

6.2 Cut generation procedure

For every solution $\bar{x}$ to the restricted master problem $[RMP]$, we need to check the feasibility of the technician-to-task assignment sub-problem. We can directly solve $[SP_2(\bar{x})]$ to optimality using a commercial solver. Nevertheless, this approach has two major drawbacks. First, since $[SP_2(\bar{x})]$ is a pure ILP model, solving the model may be too time-consuming. Second, if the cost of the solution is strictly positive, the resulting CB cut (6.19) may be too weak because it does not identify the causes of the infeasibility of the technician-to-task assignment sub-problem. Indeed, the infeasibility is likely to be caused by only a subset of the selected plans. To overcome these drawbacks and to build up stronger cuts, we propose 3 different cut generation strategies based on different approximations to the sub-problem.

6.2.1 Benders feasibility cuts

First, we can generate cuts based on solving the linear relaxation $[SP_2^{LR}(\bar{x})]$ of the formulation $[SP_2(\bar{x})]$. Since a solution $\bar{x}$ to $[RMP]$ is feasible for the whole problem only if the optimum of $[SP_2^{LR}(\bar{x})]$ is zero, $\bar{x}$ is feasible for the whole problem only if the optimum of the dual $[DSP_2^{LR}(\bar{x})]$ of $[SP_2^{LR}(\bar{x})]$ is less or equal to zero (duality theorem). Let us associate the dual variables ι_h, ϱ_r^H, and φ_{rh} to constraints (6.12),(6.13), and (6.16), respectively. The dual $[DSP_2^{LR}(\bar{x})]$ of $[SP_2^{LR}(\bar{x})]$ reads:

$$[DSP_2^{LR}(\bar{x})] \quad \max \quad \Theta_{\bar{x}}(\iota,\varrho,\varphi) = \sum_{h\in\mathcal{H}}\left(\sum_{p\in\mathcal{P}_h} q_p\bar{x}_p\iota_h + \sum_{r\in\mathcal{R}_h}\varphi_{rh}\right) + \sum_{r\in\mathcal{R}} \sum_{\substack{H\in\mathcal{C}(\mathcal{H}) \\ s.t.\ r\in\bigcup_{h\in H}\mathcal{R}_h}} \varrho_r^H \tag{6.20}$$

subject to:

$$\iota_h + \varphi_{rh} + \sum_{\substack{H\in\mathcal{C}(\mathcal{H}) \\ s.t.\ h\in H}} \varrho_r^H \leq 0 \quad \forall h\in\mathcal{H}, \forall r\in\mathcal{R}_h, \tag{6.21}$$

$$\iota_h \leq 1 \quad \forall h\in\mathcal{H}, \tag{6.22}$$

$$\varphi_{rh} \leq 0, \quad \forall h\in\mathcal{H}, \forall r\in\mathcal{R}_h, \tag{6.23}$$

$$\varrho_r^H \leq 0 \quad \forall H\in\mathcal{C}(\mathcal{H}), \forall r\in\mathcal{R}\ s.t.\ r\in\bigcup_{h\in H}\mathcal{R}_h \tag{6.24}$$

Let D be the polyhedron defined by the constraints of the dual problem $[DSP_2^{LR}(\bar{x})]$. Since $[SP_2^{LR}(\bar{x})]$ always admits a feasible solution, the dual problem $[DSP_2^{LR}(\bar{x})]$ is bounded and achieves its optimum on an extreme point of D. Denoting $\eta^1, \eta^2, ..., \eta^n$ (with $\eta^k = (\iota^k, \varrho^k, \varphi^k)$) the finite set of extreme points of D, by weak duality theorem, the following inequalities must hold to ensure the existence of a zero value solution to $[SP_2^{LR}(x)]$:

$$\Theta_x(\iota^k, \varrho^k, \varphi^k) \leq 0 \ \forall k\in\{1,...,n\} \tag{6.25}$$

Constraints (6.25) are the classical Benders feasibility cuts (hereafter referred to as BF cuts). Since a solution to $[SP_2^{LR}(\bar{x})]$ is not guaranteed to be a solution to $[SP_2(\bar{x})]$, a cut generation algorithm responsible for identifying violated constraints (6.25) will therefore not, in general, retrieve a feasible solution to $[P]$. Nevertheless, identifying violated BF cuts may help to generate less CB cuts. The advantages are that: i) BF cuts are faster to compute than CB cuts since we only need to solve a continuous linear model and ii) they may discard more solutions than just the current solution to the restricted master problem $[RMP]$. One persistent drawback of BF cuts is that they are generic, and therefore likely to be weak.

The efficiency of a coordination procedure between the two problems relies primarily on finding reduced subsets of plans causing the infeasibility of the technician-to-task assignment sub-problem. In the following subsections, we then describe two different problem-specific procedures to find these reduced subsets and we show how we build up stronger problem-specific cuts.

First, we introduce the following notation to refer to the constraints of the sub-problem:

[C1] The technician requirements for each task have to be fulfilled by technicians mastering the desired skill.

[C2] A technician cannot perform more than one task during a given time period.

[C3] The technician assignments must not violate the daily location-based incompatibilities.

[C4] Each technician has an availability schedule which must be respected.

[C5] A technician assigned to a task has to work on it from the beginning to the end, even if the task overlaps some rest time periods.

Obviously, the sub-problem is feasible if and only if constraints [C1], [C2], [C3], [C4], and [C5] are all satisfied.

Second, to simplify the discussion, we introduce the concept of *jobs* that, hereafter, simply refer either to patterns or to technician unavailability time periods. We define a job j with the following notation $(l_j, S_j, C_j, s_j, \mathcal{R}_j, q_j)$ where l_j denotes the location where j is executed, S_j its starting time period, C_j its completion time period, $\mathcal{S}_j$ a set of skills such that a technician should master at least one of these skills in order to perform j, $\mathcal{R}_j$ the set of technicians who can perform j, and q_j the number of technicians required for executing job j. For every unavailability time period of a technician $r \in \mathcal{R}$ occurring at a time period t ($t \in \mathcal{T}$ such that $\rho_r^t = 1$), we build an artificial job defined by the vector $\left(l_r^t, t, t, \mathcal{S}_r, \{r\}, 1\right)$ where $\mathcal{S}_r = \{s \in \mathcal{S} | \zeta_{rs} = 1\}$ is the set of skills mastered by technician r. If a technician is unavailable during contiguous time periods and if he or she is assigned each time at the same location (with regard to l_r^t), we associate only one job for his or her unavailability time periods. We denote as $\mathcal{J}^{\mathcal{R}}$ the set of jobs associated with technician unavailability time periods. We also associate with each solution $\bar{x}$ to the restricted master problem $[RMP]$ a set $\mathcal{J}(\bar{x})$ of jobs. We build this set following two steps. First, we add to $\mathcal{J}(\bar{x})$ all the jobs of $\mathcal{J}^{\mathcal{R}}$. Then, for every pattern $h \in \mathcal{H}$ such that $q_h(\bar{x}) > 0$ (with $q_h(\bar{x}) = \sum_{p \in \mathcal{P}_h} q_p \bar{x}_p$), we create a job defined by the vector $(l_h, S_h, C_h, \{s_h\}, \mathcal{R}_h, q_h(\bar{x}))$. Hereafter, we denote as $\mathcal{J}^{\mathcal{H}}(\bar{x})$ the set of jobs associated with patterns. We also define the parameter h_j as the pattern associated with job $j \in \mathcal{J}^{\mathcal{H}}(\bar{x})$.

6.2.2 Maximum cardinality b-matching cuts

In this section, we aim to extend the idea used to generate constraints (6.7) by partially taking into account the constraints [C3] and [C5] while fully taking into account the constraints [C1], [C2], and [C4]. More precisely, when building the potential assignments of the technicians to the tasks, we consider the unavailability time periods of the technicians and the restriction of not switching technicians during the execution of a task. We then show that the sub-problem can be approximated solving a series of maximum cardinality b-matching problems (as many as the length of the time horizon).

First, for a fixed time period $t \in \mathcal{T}$ of the time horizon and for a given solution $\bar{x}$ to the restricted master problem $[RMP]$, we introduce an undirected graph $\check{G}^t(\bar{x})$ composed of:

- a set of vertices $\check{\mathcal{V}}^t$ where $\check{\mathcal{V}}^t = \check{\mathcal{V}}^t_{\mathcal{J}} \cup \check{\mathcal{V}}^t_{\mathcal{R}}$
  - $\check{\mathcal{V}}^t_{\mathcal{J}}$: for each job $j \in \mathcal{J}(\bar{x})$ such that $S_j \le t \le C_j$, we add a vertex in $\check{\mathcal{V}}^t_{\mathcal{J}}$. Parameter j_ν denotes the job associated with a vertex $\nu \in \check{\mathcal{V}}^t_{\mathcal{J}}$. Conversely ν_j denotes the vertex associated with job j.
  - $\check{\mathcal{V}}^t_{\mathcal{R}}$: a vertex of $\check{\mathcal{V}}^t_{\mathcal{R}}$ represents a technician $r \in \mathcal{R}$ during time period t. We denote r_v the technician associated with a vertex $\nu \in \check{\mathcal{V}}^t_{\mathcal{R}}$.
- a set of edges $\check{\mathcal{U}}^t$ defined such that $\forall \nu_1 \in \check{\mathcal{V}}^t_{\mathcal{J}}, \forall \nu_2 \in \check{\mathcal{V}}^t_{\mathcal{R}}$: $(\nu_1, \nu_2) \in \check{\mathcal{U}}^t \Leftrightarrow r_{\nu_2} \in \mathcal{R}_{j_{\nu_1}}$

We now formally describe in Proposition 6.2.1 and Corollary 6.2.1 the link between the technician-to-task assignment sub-problem and a series of maximum cardinality b-matching problems defined in the graphs previously introduced.

Proposition 6.2.1. *Let $\bar{x}$ be a solution to the restricted master problem $[RMP]$ and assume that constraints [C3] and [C5] are relaxed. The technician-to-task assignment sub-problem for $\bar{x}$ is equivalent to $|\mathcal{T}|$ maximum cardinality b-matching problems in graph $\breve{G}^t(\bar{x})$ where for each time period $t \in \mathcal{T}$ function b is defined by $b_\nu = q_{j_\nu}$ for every vertex $\nu \in \breve{\mathcal{V}}^t_{\mathcal{J}}$ and by $b_\nu = 1$ for every vertex $\nu \in \breve{\mathcal{V}}^t_{\mathcal{R}}$.*

Proof. Notice that since constraints [C3] and [C5] are relaxed, the sub-problem can be independently solved for each time period of the planning horizon.

Assuming that during each time period t we can find a maximum cardinality b-matching in each of the graphs $G^t(\bar{x})$ with a cardinality equal to $\sum_{j\in\mathcal{J}(\bar{x})} \mathbb{1}_{\{S_j \le t \le C_j\}} q_j$, we can immediately build a solution to the technician-to-task assignment sub-problem from the selected edges by making the underlying assignments.

Assume now that we know a feasible solution to the technician-to-task assignment sub-problem. For each time period t, we can build a b-matching in $\breve{G}^t(\bar{x})$ from the working schedule of each technician during this specific time period. If a technician $r \in \mathcal{R}$ is assigned to a pattern $h \in \mathcal{H}$ during time period t, we select the edge $(\nu_1, \nu_2) \in \mathcal{U}$ where $j_{\nu_1} = j_h$ (j_h denoting the job associated with the pattern h) and $r_{\nu_2} = r$. This construction ensures the building of a b-matching. Moreover, since all the requirements are fulfilled, this b-matching has the maximum possible cardinality. □

Corollary 6.2.1. *If we assume that constraints [C3] and [C5] are relaxed, the technician-to-task assignment sub-problem is feasible for a solution $\bar{x}$ to the restricted master problem $[RMP]$ if and only if the maximum cardinality b-matching in graph $\breve{G}^t(\bar{x})$ for each time period $t \in \mathcal{T}$ contains $\sum\limits_{j\in\mathcal{J}(\bar{x})} \mathbb{1}_{\{S_j \le t \le C_j\}} q_j$ edges of $\breve{\mathcal{U}}^t$.*

Proof. This is a direct consequence of Proposition 6.2.1.

□

Let us assume a fixed time period t. Solving the maximum cardinality b-matching in $\breve{G}^t(\bar{x})$ from $\breve{\mathcal{V}}^t_{\mathcal{J}}$ to $\breve{\mathcal{V}}^t_{\mathcal{R}}$ is equivalent to solving a maximum flow problem in a slightly modified version of this graph. We use this equivalence to derive new cuts. We denote $\widehat{G}^t(\bar{x})$ this new directed graph and $\widehat{\mathcal{V}}^t$ and $\widehat{\mathcal{U}}^t$ the new sets of vertices and arcs. We build graph $\widehat{G}^t(\bar{x})$ as follows:

1. We define $\widehat{\mathcal{V}}^t$ as $\widehat{\mathcal{V}}^t = \breve{\mathcal{V}}^t \cup \{\varpi^t, \underline{\omega}^t\}$ where the two new vertices ϖ^t and $\underline{\omega}^t$ represent the source and the sink vertices.
2. For every directed arc $(\nu_1, \nu_2) \in \widehat{\mathcal{U}}^t$, we denote as $\gamma^{max}_{\nu_1\nu_2}$ its maximal capacity and as $f_{\nu_1\nu_2}$ the number of units of flow on the arc. We formally define $\widehat{\mathcal{U}}^t$ as follows:
   - $\forall \nu_1 \in \breve{\mathcal{V}}^t_{\mathcal{J}}, \forall \nu_2 \in \breve{\mathcal{V}}^t_{\mathcal{R}}$: $(\nu_1, \nu_2) \in \widehat{\mathcal{U}}^t \Leftrightarrow r_{\nu_2} \in \mathcal{R}_{j_{\nu_1}}$ and $\gamma^{max}_{\nu_1\nu_2} = +\infty$
   - $\forall \nu \in \breve{\mathcal{V}}^t_{\mathcal{J}}$: $(\varpi^t, \nu) \in \widehat{\mathcal{U}}^t$ and $\gamma^{max}_{\varpi^t\nu} = q_{j_\nu}$
   - $\forall \nu \in \breve{\mathcal{V}}^t_{\mathcal{R}}$: $(\nu, \underline{\omega}^t) \in \widehat{\mathcal{U}}^t$ and $\gamma^{max}_{\nu\underline{\omega}^t} = 1$

Let us denote $f^*(t, \bar{x})$ the value of the maximum flow in $\widehat{G}^t(\bar{x})$. If $f^*(t, \bar{x}) < \sum_{\nu\in\breve{\mathcal{V}}^t_{\mathcal{J}}} q_{j_\nu}$ then the jobs of $\mathcal{J}(\bar{x})$ overlapping t cannot be fully scheduled during this time period. We therefore have to to discard the solution $\bar{x}$. We first compute the minimum flow cut in graph $\widehat{G}^t(\bar{x})$ (see Figure 6.1). The minimum flow cut can be described by the sets $\widehat{\mathcal{V}}^t(+)$ and $\widehat{\mathcal{V}}^t(-)$ that are composed as follows: $\widehat{\mathcal{V}}^t(+) = \{\varpi^t\} \cup \breve{\mathcal{V}}^t_{\mathcal{J}}(+) \cup \breve{\mathcal{V}}^t_{\mathcal{R}}(+)$ and $\widehat{\mathcal{V}}^t(-) = \breve{\mathcal{V}}^t_{\mathcal{J}}(-) \cup \breve{\mathcal{V}}^t_{\mathcal{R}}(-) \cup \{\underline{\omega}^t\}$ with $\breve{\mathcal{V}}^t_{\mathcal{J}} = \breve{\mathcal{V}}^t_{\mathcal{J}}(-) \cup \breve{\mathcal{V}}^t_{\mathcal{J}}(+)$ and $\breve{\mathcal{V}}^t_{\mathcal{R}} = \breve{\mathcal{V}}^t_{\mathcal{R}}(-) \cup \breve{\mathcal{V}}^t_{\mathcal{R}}(+)$.

Applying the *max-flow/min-cut* theorem on graph $\widehat{G}^t(\bar{x})$, we can state that:

$$f^*(t, \bar{x}) = \sum_{\nu\in\breve{\mathcal{V}}^t_{\mathcal{J}}(-)} \gamma^{max}_{\varpi^t v} + \sum_{v\in\breve{\mathcal{V}}^t_{\mathcal{R}}(+)} \gamma^{max}_{v\underline{\omega}^t} \tag{6.26}$$

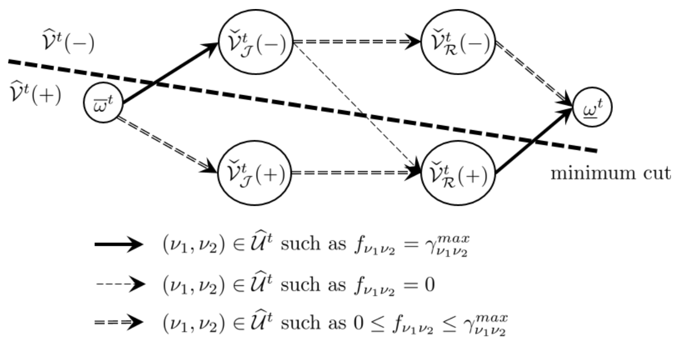


Figure 6.1 – Minimum cut in graph $\widehat{G}^t(\bar{x})$

If we replace the capacity of each arc by its value, we obtain:

$$f^*(t, \bar{x}) = \sum_{\nu \in \check{\mathcal{V}}^t_{\mathcal{J}}(-)} q_{j_\nu} + |\check{\mathcal{V}}^t_{\mathcal{R}}(+)| \tag{6.27}$$

The valid minimum flow cut (that invalidates $\bar{x}$) reads:

$$\sum_{\nu \in \check{\mathcal{V}}^t_{\mathcal{J}}(-)} q_{j_\nu} + |\check{\mathcal{V}}^t_{\mathcal{R}}(+)| \geq \sum_{\nu \in \check{\mathcal{V}}^t_{\mathcal{J}}} q_{j_\nu} \tag{6.28}$$

which we can reformulate as follows:

$$\sum_{\nu \in \check{\mathcal{V}}^t_{\mathcal{J}}(+)} q_{j_\nu} \leq |\check{\mathcal{V}}^t_{\mathcal{R}}(+)| \tag{6.29}$$

Note that inequality (6.29) leads to the following valid constraint (hereafter referred to as a *maximum cardinality b-matching* (MCbM) cut) that eliminates the solution $\bar{x}$ from the feasible region of the restricted master problem $[RMP]$:

$$\sum_{\substack{j \in \mathcal{J}^{\mathcal{H}}(\bar{x}) \\ s.t.\ \nu_j \in \check{\mathcal{V}}^t_{\mathcal{J}}(+)}} \sum_{p \in \mathcal{P}_{h_j}} q_p x_p \leq |\check{\mathcal{V}}^t_{\mathcal{R}}(+)| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ s.t.\ \nu_j \in \check{\mathcal{V}}^t_{\mathcal{J}}(+)}} q_j \tag{6.30}$$

We now demonstrate how in some cases it is possible to tighten the previous MCbM cut by reasoning about its composition. First, note that $|\check{\mathcal{V}}^t_{\mathcal{R}}(+)| \neq |\mathcal{R}|$ because at least one of the cumulative constraints (6.7) would be unsatisfied otherwise. This also implies that $|\check{\mathcal{V}}^t_{\mathcal{R}}(-)| \neq 0$. By definition of the max-flow/min-cut, the technicians associated with the set $\check{\mathcal{V}}^t_{\mathcal{R}}(-)$ are either not connected to any other vertices or they are assigned to the jobs $j \in \mathcal{J}(\bar{x})$ such that $\nu_j \in \check{\mathcal{V}}^t_{\mathcal{J}}(-)$, but they cannot be assigned to any of the jobs j such that $\nu_j \in \check{\mathcal{V}}^t_{\mathcal{J}}(+)$. The latter means that either these technicians do not have the required skills to perform those jobs or they have at least one unavailability time period that prevent them to be assigned to those jobs. We can deduce that inequality (6.29) is also valid for every potential job that overlaps time period t and that cannot be performed by any of the technicians associated with a vertex of set $\check{\mathcal{V}}^t_{\mathcal{R}}(-)$. The MCbM cut (6.30) can be rewritten as:

$$\sum_{h \in \mathcal{H}} \Psi^t_{\check{\mathcal{V}}^t_{\mathcal{R}}(-)}(h) \sum_{p \in \mathcal{P}_h} q_p x_p \leq |\check{\mathcal{V}}^t_{\mathcal{R}}(+)| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ s.t.\ \nu_j \in \check{\mathcal{V}}^t_{\mathcal{J}}(+)}} q_j \tag{6.31}$$

where $\Psi^t_{\check{\mathcal{V}}^t_{\mathcal{R}}(-)}(h)$ is equal to 1 if and only if pattern h overlaps time period t and none of the technicians associated with the set $\mathcal{V}^t_{\mathcal{R}}(-)$ can be assigned to h.

Last but not least, it is noteworthy that the maximum cardinality b-matching problem has only to be solved for every time period t where at least one technician cannot be assigned to a job because of an unavailability time period occurring at a time period other than t. Otherwise, constraints (6.7) are necessary and sufficient condition of the existence of a b-matchings with the desired cardinality.

6.2.3 Maximum-weight clique cuts

Another strategy to check that a given solution $\bar{x}$ to the restricted master problem $[RMP]$ leads to a feasible sub-problem relies on proving that it is impossible to assign the technicians to the tasks without violating the location-based incompatibilities. Since these constraints are defined by day, this search decomposes into $|\mathcal{D}|$ independent searches in which for each day $d \in \mathcal{D}$ we only consider the jobs of $\mathcal{J}(\bar{x})$ that overlap d. Moreover, since the daily location-based incompatibilities are checked individually for each technician, they impact the number of available technicians at each location. During the search, it is therefore necessary to take into account the skills required to perform the different jobs. For a fixed subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$, we only consider the jobs $j \in \mathcal{J}(\bar{x})$ such that $\mathcal{S}_j \cap \bar{\mathcal{S}} \neq \emptyset$ and the technicians mastering at least one skill of this subset. This procedure increases the likelihood of finding violated daily location-based incompatibilities if the current solution to the restricted master problem does not lead to a feasible technician-to-task assignment sub-problem. This is particularly true when the ratio between the requirements and the number of available technicians varies widely across skills.

To look for violated constraints, we solve for each day d and for each subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$ a maximum-weight clique problem in an undirected graph $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$. The graph $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ is composed of:

- a set of vertices $\widetilde{\mathcal{V}}^d_{\bar{\mathcal{S}}}$ such that each vertex maps a job j of set $\mathcal{J}(\bar{x})$ that i) overlaps day d (i.e., $\mathcal{T}_d \cap \{S_j, ..., C_j\} \neq \emptyset$) and ii) requires at least one of the skill in $\bar{\mathcal{S}}$ (i.e., $\mathcal{S}_j \cap \bar{\mathcal{S}} \neq \emptyset$). We denote j_ν the job associated with vertex $\nu \in \widetilde{\mathcal{V}}^d_{\bar{\mathcal{S}}}$. We associate with every vertex ν a weight equal to the number of technicians q_{j_ν} required to perform job j_ν.
- a set of edges $\widetilde{\mathcal{U}}^d_{\bar{\mathcal{S}}}$ where for all vertices $\nu_1, \nu_2 \in \widetilde{\mathcal{V}}^d_{\bar{\mathcal{S}}}$:
  $(\nu_1, \nu_2) \in \widetilde{\mathcal{U}}^d_{\bar{\mathcal{S}}} \Leftrightarrow \nu_1 \neq \nu_2 \wedge \left(\left(S_{j_{\nu_2}} \leq C_{j_{\nu_1}} \wedge S_{j_{\nu_1}} \leq C_{j_{\nu_2}} \right) \vee \sigma_{l_{j_{\nu_1}} l_{j_{\nu_2}}} = 0 \right)$
  There exists an edge between two vertices ν_1 and ν_2 in $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ if and only if a technician cannot be assigned to both jobs j_{ν_1} and j_{ν_2} with regard to constraints [C2] and [C3]. $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ is a kind of sub-graph of graph G used to derive the clique constraints (6.13) in formulation $[SP_2(\bar{x})]$. The only difference comes from the insertion of jobs related to technician unavailability time periods.

Proposition 6.2.2 formally describes the link between the resolution of the technician-to-task assignment sub-problem and the resolution of maximum-weight clique problems.

Proposition 6.2.2. *The technician-to-task assignment sub-problem is feasible for a solution $\bar{x}$ to the restricted master problem $[RMP]$ if for each subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$ of skills and for each day $d \in \mathcal{D}$, the maximum weight of a clique in graph $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ is less than or equal to $|\mathcal{R}_{\bar{\mathcal{S}}}|$.*

Proof. For a fixed day d and a fixed subset of skills $\bar{\mathcal{S}}$, suppose by contradiction that the maximum weight of a clique in graph $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ is strictly greater than $|\mathcal{R}_{\bar{\mathcal{S}}}|$. By construction of $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$, a technician cannot perform more than one job among the jobs whose vertices belong to that clique. Since we need more technicians than those actually available, the technician-to-task assignment sub-problem is infeasible. □

Let us now assume a fixed day $d \in \mathcal{D}$ and a fixed subset of skills $\bar{\mathcal{S}}$. Let us also denote $\mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})$ the set of vertices that belong to the maximum-weight clique of graph $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$. If the total weight of the vertices in $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ is strictly greater than $|\mathcal{R}_{\bar{\mathcal{S}}}|$, the constraint that eliminates $\bar{x}$ is:

$$\sum_{\nu \in \mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})} q_{j_\nu} \leq |\mathcal{R}_{\bar{\mathcal{S}}}| \tag{6.32}$$

The valid constraint (hereafter referred to as *maximum-weight clique* (MWC) cut) that discards a solution $\bar{x}$ to the restricted master problem $[RMP]$ is therefore:

$$\sum_{\substack{j \in \mathcal{J}^{\mathcal{H}}(\bar{x}) \\ s.t.\ \nu_j \in \mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})}} \sum_{p \in \mathcal{P}_{h_j}} q_p x_p \leq |\mathcal{R}_{\bar{\mathcal{S}}}| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ s.t.\ \nu_j \in \mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})}} q_j \tag{6.33}$$

This cut simply states that the number of technicians required by the jobs associated with the vertices of the clique has to be lower than the number of technicians mastering at least a skill in $\bar{\mathcal{S}}$. We derived this cut not only for the maximum-weight clique but also for all the cliques that have a weight greater than $|\mathcal{R}_{\bar{\mathcal{S}}}|$.

It is possible to tighten the MWC cut (6.33) by adding some additional plans on its left side. More precisely, we proceed as follows. First, we consider the sub-graph $G^{sub}(\bar{x})$ of graph G (graph G is used to derive the clique inequalities in the formulation $[SP_2(\bar{x})]$) that includes a vertex for pattern $h \in \mathcal{H}$ if: i) $h \in \mathcal{H} \setminus \mathcal{H}(\bar{x})$ (i.e. $q_h(\bar{x}) = 0$), ii) h overlaps day d, iii) $s_h \in \bar{\mathcal{S}}$, and iv) in the case that a technician is assigned to h, he or she cannot be assigned to any job involved in $\mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})$ with regard to constraints [C2] and [C3]. We then solve a maximum clique problem in sub-graph $G^{sub}(\bar{x})$. Let us denote $\mathcal{H}\left[\mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})\right]$ the set of patterns associated with the vertices that are part of the maximum clique of $G^{sub}(\bar{x})$. Observing that a technician cannot be assigned to more than one of the jobs in $\mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})$ or one of the patterns of $\mathcal{H}\left[\mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})\right]$, we can rewrite the MWC cut (6.33) as follows:

$$\sum_{\substack{j \in \mathcal{J}^{\mathcal{H}}(\bar{x}) \\ s.t.\ \nu_j \in \mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})}} \sum_{p \in \mathcal{P}_{h_j}} q_p x_p + \sum_{h \in \mathcal{H}\left[\mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})\right]} \sum_{p \in \mathcal{P}_h} q_p x_p \leq |\mathcal{R}_{\bar{\mathcal{S}}}| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ s.t.\ \nu_j \in \mathcal{C}^d_{\bar{\mathcal{S}}}(\bar{x})}} q_j \tag{6.34}$$

For efficiency consideration, we reduce the size of the sub-graph $G^{sub}(\bar{x})$ observing that it is sufficient to only consider one vertex for all the patterns that satisfy two conditions: same location and overlapping of the same portion of the day. We point out that this remark also applies for graph $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$.

Last but not least, to avoid overloading our algorithm, we solve the maximum-weight clique problem only if: i) the sum of the weights of the vertices is greater than the number of available technicians and ii) there exists during a particular day at least two jobs that do not overlap and are executed at incompatible locations. Otherwise, the cumulative constraints (6.7) ensure for each day and for each subset of skills the non-existence of a clique with a weight strictly greater than the number of available technicians. We use the algorithms introduced in (Östergård, 2001) and (Östergård, 2002) for solving the maximum clique and maximum-weight clique problems.

Since the approximations to the sub-problem described in Section 6.2.2 and in Section 6.2.3 can be decomposed into a series of small problems, we can potentially identify multiple subsets of plans that cause the infeasibility of the technician-to-task assignment sub-problem. This usually leads to the generation of multiple cuts, which is known to significantly improve the efficiency of a cut generation process. We can also think to run the resolution of those small problems in parallel.

6.2.4 Illustrative examples

For the sake of clarity, we provide here three examples to illustrate how we build the cuts previously described. For the approximations described in 6.2.2 and 6.2.3, the first two examples show that there is no strict dominance of one over the other one (i.e., neither of the MCbM and MWC cuts are the strongest cuts). The third example is meant to illustrate a case where the two previous approximations do not find any cut although the technician-to-task assignment sub-problem is infeasible.

Example 1

This example (referred to as *Example 1*) illustrates thoroughly how we build the different cuts previously described.

We consider in this example a fixed time horizon made up of 8 time periods of identical length ($\mathcal{T} = \{1, 2, ..., 8\}$) and partitioned into two days: time periods 1 to 4 belong to day 1 (i.e., $\mathcal{T}_1 = \{1, 2, 3, 4\}$) and time periods 5 to 8 to day 2 (i.e., $\mathcal{T}_2 = \{5, 6, 7, 8\}$). We have three different locations denoted as l_1, l_2, and l_3 (i.e., $\mathcal{L} = \{l_1, l_2, l_3\}$). Locations l_2 and l_3 cannot be visited by a technician within the same day (i.e., $\sigma_{l_2 l_3} = 0$). We do not define any other daily location-based incompatibilities. We consider 4 tasks to schedule ($\mathcal{I} = \{A, B, C, D\}$), 3 technicians ($\mathcal{R} = \{r_1, r_2, r_3\}$), and 3 skills ($\mathcal{S} = \{s_1, s_2, s_3\}$). The characteristics of the tasks and the technicians are defined in Table 6.1a and in Table 6.1b. For the sake of simplicity, we do not explicitly introduce all the parameters defining an instance of the problem, we introduce only those which are useful for the illustration of the cut generation process.

Table 6.2 shows a given solution to the restricted master problem $[RMP]$ in which no cuts have been previously added. According to the selected plans, the table reports the starting and completion time periods of each task as well as the number of technicians required to perform every task. We refer to this solution using symbol $\bar{x}$. Note that $\bar{x}$ satisfies the cumulative constraints (6.7) of the restricted master problem [1].

Table 6.1 – Data of Example 1.

$\mathcal{I}$	l_i	s_i
A	l_1	s_1
B	l_2	s_1
C	l_1	s_2
D	l_3	s_3

(a) Characteristics of the tasks.

$\mathcal{R}$	$\{s \in \mathcal{S} \mid \zeta_{rs} = 1\}$	unavailability time periods
r_1	$\{s_1\}$	at location l_3 during time period 8
r_2	$\{s_1, s_2, s_3\}$	–
r_3	$\{s_1, s_3\}$	–

(b) Characteristics of the technicians.

Table 6.2 – A solution to the restricted master problem for Example 1.

$\mathcal{I}$	Selected plan p	S_p	C_p	q_p	$\mathcal{R}_p$
A	p_A	2	5	1	$\{r_1, r_2, r_3\}$
B	p_B	4	7	2	$\{r_2, r_3\}$
C	p_C	7	8	1	$\{r_2\}$
D	p_D	1	3	2	$\{r_2, r_3\}$

In the following, symbol u_{r_1} refers to the job associated with the unavailability time period of technician r_1. Observing that u_{r_1} and p_B both overlaps day 2 and are defined at two incompatible locations l_2 and l_3, we can deduce that technician r_1 cannot be assigned to plan p_B (although he or she has the required skill).

Note that in solution $\bar{x}$ we have as many patterns as selected plans, and therefore we introduce as many jobs as plans. Job j_A refers then to plan p_A, job j_B refers then to plan p_B, and so on.

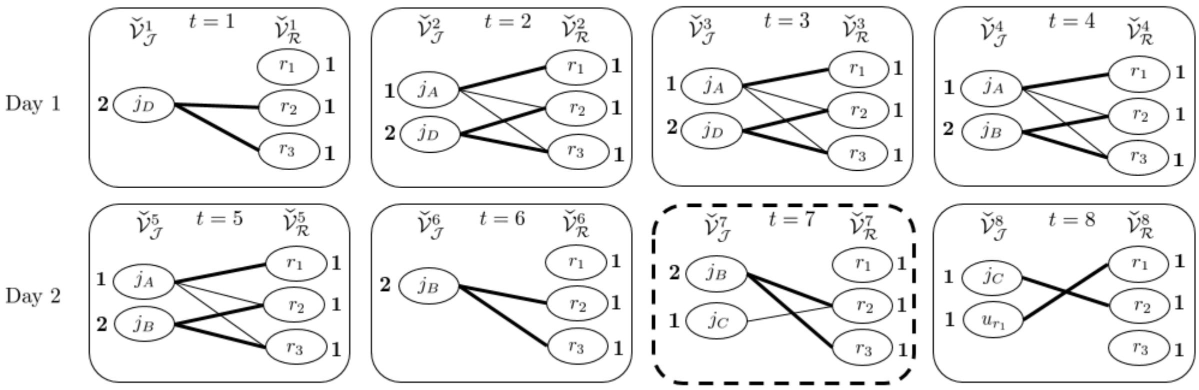
First, let us look at the potential generation of MCbM cuts. Figure 6.2 describes graph $\breve{G}^t(\bar{x})$ for each time period of the planning horizon for Example 1.

For time period $t = 7$, the value of the maximum flow problem in graph $\widehat{G}^7(\bar{x})$ is equal to 2 and so is the maximum cardinality of a b-matching in $G^7(\bar{x})$. Since $q_{j_B} + q_{j_C} = 3$, $\bar{x}$ is an infeasible solution to the whole problem. We then compute the minimum cut in graph $\widehat{G}^7(\bar{x})$ (see Figure 6.3).

From the general expression (6.30), we build the MCbM cut (6.35).

$$2x_{p_B} + 1x_{p_C} \leq 2 \tag{6.35}$$

1. For instance, at time period 7 we have seven cumulative constraints. Plugging in the values of the variables, we obtain $2 \leq 3$, $1 \leq 1$, $0 \leq 2$, $3 \leq 3$, $2 \leq 3$, $1 \leq 2$, $3 \leq 3$ when $\bar{\mathcal{S}}$ is respectively equal to $\{s_1\}$,$\{s_2\}$,$\{s_3\}$,$\{s_1, s_2\}$,$\{s_1, s_3\}$,$\{s_2, s_3\}$,$\{s_1, s_2, s_3\}$



NB: in bold a solution to the maximum cardinality b-matching problem, a rectangular dash-line box means that no b-matching with the desired cardinality can be found

Figure 6.2 – Structure of graphs $\breve{G}^t(\bar{x})$ for Example 1 and different values of $t \in \mathcal{T}$.

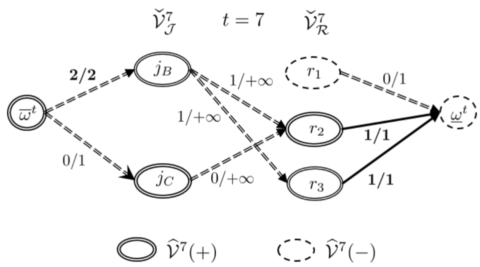
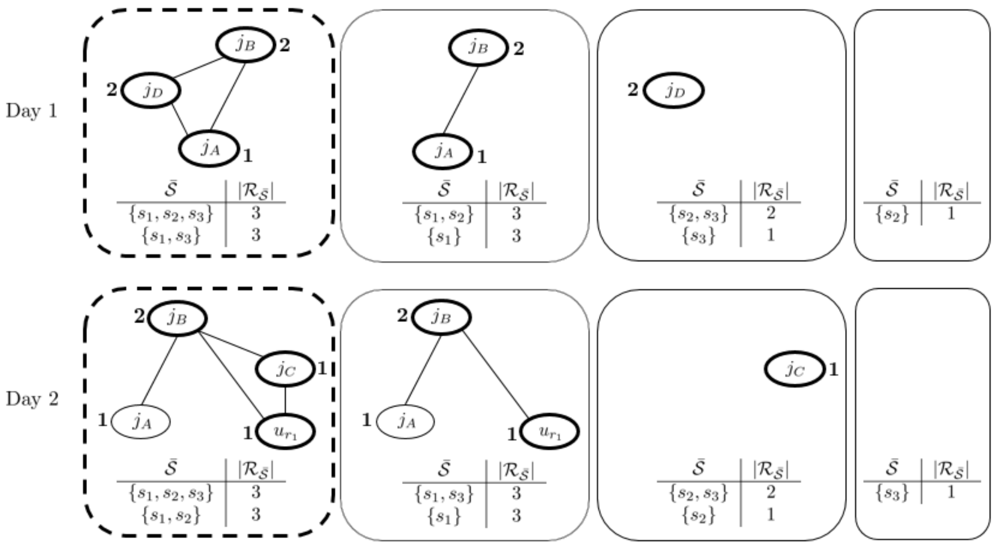


Figure 6.3 – Minimum cut in graph $\widehat{G}^7(\bar{x})$ for Example 1.

Second, let us look at the potential generation of MWC cuts. For illustration purposes, Figure 6.4 depicts $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ for every day $d \in \{\text{day 1, day 2}\}$ and every subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$ of skills.



NB: in bold a solution to the maximum-weight clique problem, a rectangular dash-line box means that the weight of this clique is strictly greater than the maximum allowed

Figure 6.4 – Structure of $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ for Example 1

Looking at Figure 6.4, one can see that the maximum-weight clique is strictly greater than the number of available technicians in 4 different cases. From the general expression (6.33), we build the MWC cuts (6.36) and (6.37).

$$1x_{p_A} + 2x_{p_B} + 1x_{p_C} \leq 3 \quad (6.36)$$
$$2x_{p_B} + 1x_{p_C} \leq 2 \quad (6.37)$$

The cut (6.36) is built from the clique computed either in $\widetilde{G}^1_{\{s_1,s_2,s_3\}}(\bar{x})$ or in $\widetilde{G}^1_{\{s_1,s_3\}}(\bar{x})$. In the same way, the cut (6.37) is built from the clique computed either in $\widetilde{G}^2_{\{s_1,s_2,s_3\}}(\bar{x})$ or in $\widetilde{G}^2_{\{s_1,s_2\}}(\bar{x})$.

Third, let us solve the formulation $[SP_2^{LR}(\bar{x})]$ with a commercial solver. Since its optimum value is strictly greater than zero (equal to 2), we identify the violated BF cut (6.38).

$$2x_{p_B} + x_{p_C} + 2x_{p_D} \leq 3 \quad (6.38)$$

Fourth, let us solve the ILP formulation $[SP_2(\bar{x})]$. As with the linear relaxation, the optimum value is equal to 2. We then generate the CB cut (6.39).

$$x_{p_A} + x_{p_B} + x_{p_C} + x_{p_D} \leq 3 \quad (6.39)$$

Table 6.3 reports the infeasible sections of plans discarded by the MCbM, MWC, BF, and CB cuts. We denote each selection of plans using a four dimensional vector where the first, second, third and fourth coordinate refers to the plan selected for task A, task B, task C, and task D, respectively. Symbol "..." simply means that the infeasibility of the plans selection holds for any executing plan selected at the corresponding coordinate.

Table 6.3 – The infeasible plans selections discarded in Example 1.

Plans selection	CB	BF	MCbM	MWC
(p_A, p_B, p_C, p_D)	✓	✓	✓	✓
$(p_A, p_B, p_C, ...)$			✓	✓
$(p_A, p_B, ..., p_D)$		✓		✓
$(..., p_B, p_C, p_D)$		✓	✓	✓
$(..., p_B, p_C, ...)$			✓	✓
$(..., p_B, ..., p_D)$		✓		✓

For the cut (6.35), notice that we can build a stronger MCbM cut of type (6.31) as described in Section 6.2.2. Indeed, we can add to the left hand side of the cut (6.35) all the patterns overlapping time period 7 to which technician r_1 cannot be assigned (this is the only technician associated with a vertex of set $\check{\mathcal{V}}^t_{\mathcal{R}}(-)$).

For the MWC cuts (6.36) and (6.37), we can build stronger MWC cuts of type (6.34) as described in Section 6.2.3. To strengthen MWC cut (6.36), we consider the sub-graph $G^{sub}(\bar{x})$ of G that includes the vertices linked to: i) patterns at location l_1 overlapping time periods 3 and 4, ii) patterns at location l_2 overlapping time periods 4, and iii) patterns at location l_3 overlapping at least time period 2, 3, or 4. Indeed, one technician cannot be assigned to any of the previous patterns if he or she is assigned to pattern p_A, p_B or p_D. We then solve a maximum clique problem in this sub-graph, and we add to the left hand side of the MWC cut (6.36) all the plans associated with the patterns involved in the maximum clique. We proceed on a similar way for cut (6.37) by considering the sub-graph $G^{sub}(\bar{x})$ of G that includes the vertices linked to: i) patterns at location l_1 overlapping time periods 7 and 8, ii) patterns at location l_2 overlapping at least time period 7 or 8, and iii) patterns at location l_3 overlapping at least time period 8. Indeed, one technician cannot be assigned to any of the previous patterns if he or she is assigned to pattern p_B, p_D or to the job u_{r_1}. Again, we solve a maximum clique problem in this sub-graph, and we add to the left hand side of the MWC cut (6.37) all the plans associated with the patterns involved in the maximum clique. In this example, it is worth noting that when building the sub-graph, we do not pay a special attention to the skill associated with the patterns because the MWC cuts (6.36) and (6.37) have been computed with $\bar{\mathcal{S}} = \mathcal{S}$. Otherwise, only the patterns having their skill in $\bar{\mathcal{S}}$ can be added to the left hand side of the cuts. (since the right hand side of the cuts is based on the total number of technicians mastering at least one skill of $\bar{\mathcal{S}}$).

Example 1 illustrates a case where the approximation described in Section 6.2.3 dominates the approximation described in Section 6.2.2 (i.e., a case where the MWC cuts are stronger than the MCbM cuts)

Example 2

We introduce now a second example (referred to as *Example 2*) to illustrate a case where the approximation described in Section 6.2.2 dominates the approximation described in Section 6.2.3 (i.e., a case where the MCbM cuts are stronger than the MWC cuts)

In this example, we consider a fixed time horizon made up of 4 time periods ($\mathcal{T} = \{1, 2, 3, 4\}$) and partitioned into two days: time periods 1 and 2 belongs to day 1 and time periods 3 and 4 belongs to day 2. We consider 1 location ($\mathcal{L} = \{l\}$), 2 tasks to schedule ($\mathcal{I} = \{E, F\}$), 1 skill ($\mathcal{S} = \{s\}$), and 2 technicians ($\mathcal{R} = \{r_4, r_5\}$). The technician r_2 is unavailable during time periods 1 and 3. The characteristics of the tasks and the technicians are summarized in Table 6.4a and in Table 6.4b. We denote with symbol u_{r_5} the jobs associated with the unavailability time periods of technician r_5.

Table 6.4 – Data of Example 2.

(a) Characteristics of the tasks.

$\mathcal{I}$	l_i	s_i
E	l	s
F	l	s

(b) Characteristics of the technicians.

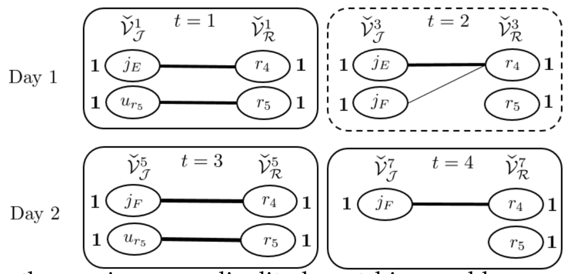
$\mathcal{R}$	$\{s \in \mathcal{S} \mid \lambda_{rs} = 1\}$	unavailability time periods
r_4	$\{s\}$	–
r_5	$\{s\}$	at location l during time periods 1 and 3

Table 6.5 shows a given solution to the restricted master problem $[RMP]$ in which no cuts have been previously added. According to the selected plans, the table reports the starting and completion time periods of each task as well as the number of technicians required to perform every task. Note that $\bar{x}$ satisfies the cumulative constraints (6.7) of the restricted master problem.

Table 6.5 – A solution to the restricted master problem for Example 2.

$\mathcal{I}$	Selected plan p	S_p	C_p	q_p	$\mathcal{R}_p$
E	p_E	1	2	1	$\{r_4\}$
F	p_F	2	4	1	$\{r_4\}$

First, let us look at the potential generation of MCbM cuts. Figure 6.5 describes graph $\breve{G}^t(\bar{x})$ for each time period of the horizon for Example 2.



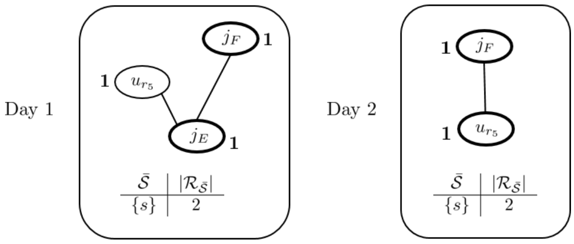
NB: in bold a solution to the maximum cardinality b-matching problem, a rectangular dash-line box means that no b-matching with the desired cardinality can be found

Figure 6.5 – Structure of graphs $\breve{G}^t(\bar{x})$ for Example 2 and different values of $t \in \mathcal{T}$.

Since technician r_2 cannot be assigned to task B because of its personal availability schedule, the maximum cardinality of a b-matching at time period 2 is equal to 1 whereas the tasks scheduled during this time period require a total of two technicians. The following MCbM cut is then produced:

$$x_{p_E} + x_{p_F} \leq 1 \tag{6.40}$$

Second, let us look at the potential generation of MWC cuts. Figure 6.6 depicts $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ for every day $d \in \{$day 1, day 2$\}$.



NB: in bold a solution to the maximum-weight clique problem

Figure 6.6 – Structure of $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ for Example 2

We immediately see that solving maximum-weight clique problems does not enable us to produce any MWC cut for this example. Therefore, Example 2 illustrates a case where the approximation described in Section 6.2.2 dominates the approximation described in Section 6.2.3.

Example 3

This third example (referred to as *Example 3*) is meant to illustrate a case where the problem-specific approximations do not find any cut although the technician-to-task assignment sub-problem is infeasible.

We consider in this example a fixed time horizon of 8 time periods ($\mathcal{T} = \{1, ..., 8\}$) and partitioned into two days: time periods 1 to 4 belongs to day 1 and time periods 5 and 8 belongs to day 2. We consider 1 location ($\mathcal{L} = \{l\}$), 4 tasks to schedule ($\mathcal{I} = \{G, H, I, J\}$), 3 skills ($\mathcal{S} = \{s_6, s_7, s_8\}$), and 2 technicians ($\mathcal{R} = \{r_6, r_7\}$). The characteristics of the tasks and the technicians are summarized in Table 6.6a and in Table 6.6b.

Table 6.6 – Data of Example 3.

(a) Characteristics of the tasks.

$\mathcal{I}$	l_i	s_i
G	l	s_6
H	l	s_6
I	l	s_7
J	l	s_8

(b) Characteristics of the technicians.

$\mathcal{R}$	$\{s \in \mathcal{S} \mid \lambda_{rs} = 1\}$	unavailability time periods
r_6	$\{s_6\}$	–
r_7	$\{s_6, s_7, s_8\}$	–

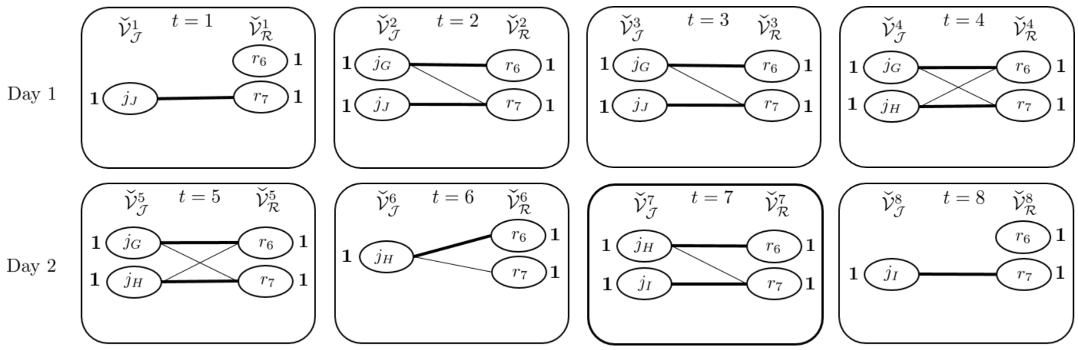
Table 6.7 shows a given solution to the restricted master problem $[RMP]$ in which no cuts have been previously added. According to the selected plans, the table reports the starting and completion time periods of each task as well as the number of technicians required to perform every task. Again, it is easy to verify that $\bar{x}$ satisfies the cumulative constraints (6.7) of the restricted master problem.

Table 6.7 – A solution to the restricted master problem for Example 3.

$\mathcal{I}$	Selected plan p	S_p	C_p	q_p	$\mathcal{R}_p$
G	p_G	2	5	1	$\{r_6, r_7\}$
H	p_H	4	7	1	$\{r_6, r_7\}$
I	p_I	7	8	1	$\{r_7\}$
J	p_J	1	3	1	$\{r_7\}$

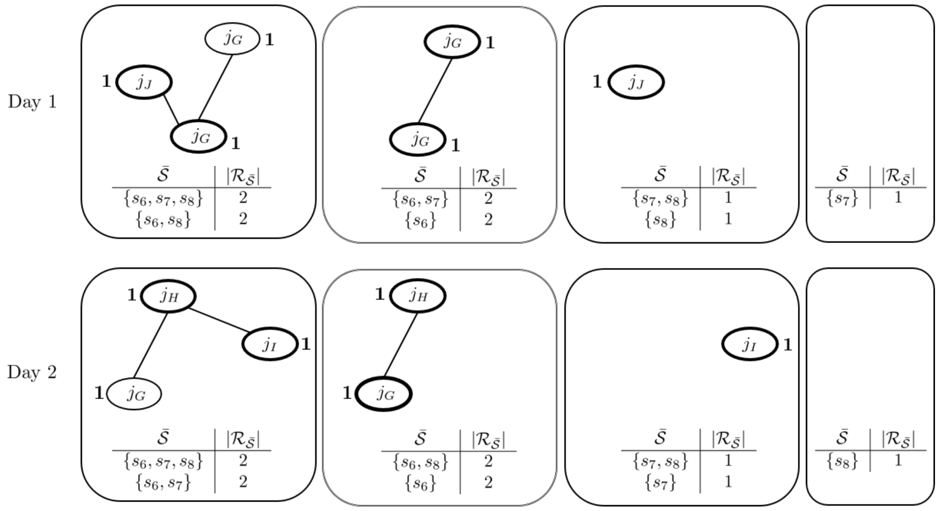
First, let us look at the potential generation of MCbM cuts. Figure 6.7 describes graph $\breve{G}^t(\bar{x})$ for each time period of the horizon for Example 3. We observe that solving maximum cardinality b-matching problems does not enable us to produce any MCbM cut for this example.

Second, let us look at the potential generation of MWC cuts. Figure 6.8 depicts $\widetilde{G}^d_{\bar{\mathcal{S}}}(\bar{x})$ for every day $d \in \{day1, day2\}$. Similarly, no MWC cuts are produced from this approximation.



NB: in bold a solution to the maximum cardinality b-matching problem

Figure 6.7 – Structure of graphs $\breve{G}^t(\bar{x})$ for Example 3 and different values of $t \in \mathcal{T}$.



NB: in bold a solution to the maximum-weight clique problem

Figure 6.8 – Structure of $\widetilde{G}^d_{\mathcal{S}}(\bar{x})$ for Example 3

However, it is easy to see that the technician-to-task assignment sub-problem does not admit any solution. This comes from two observations. First, it is clear that technician r_7 has to perform tasks I and J. Second, tasks G and H cannot be performed by the same technician since they overlap. The same holds for tasks G and J as well as for tasks H and I. The technician-to-task assignment sub-problem therefore does not admit any solution since technician r_7 cannot perform either tasks G, I and J, or tasks H, I and J.

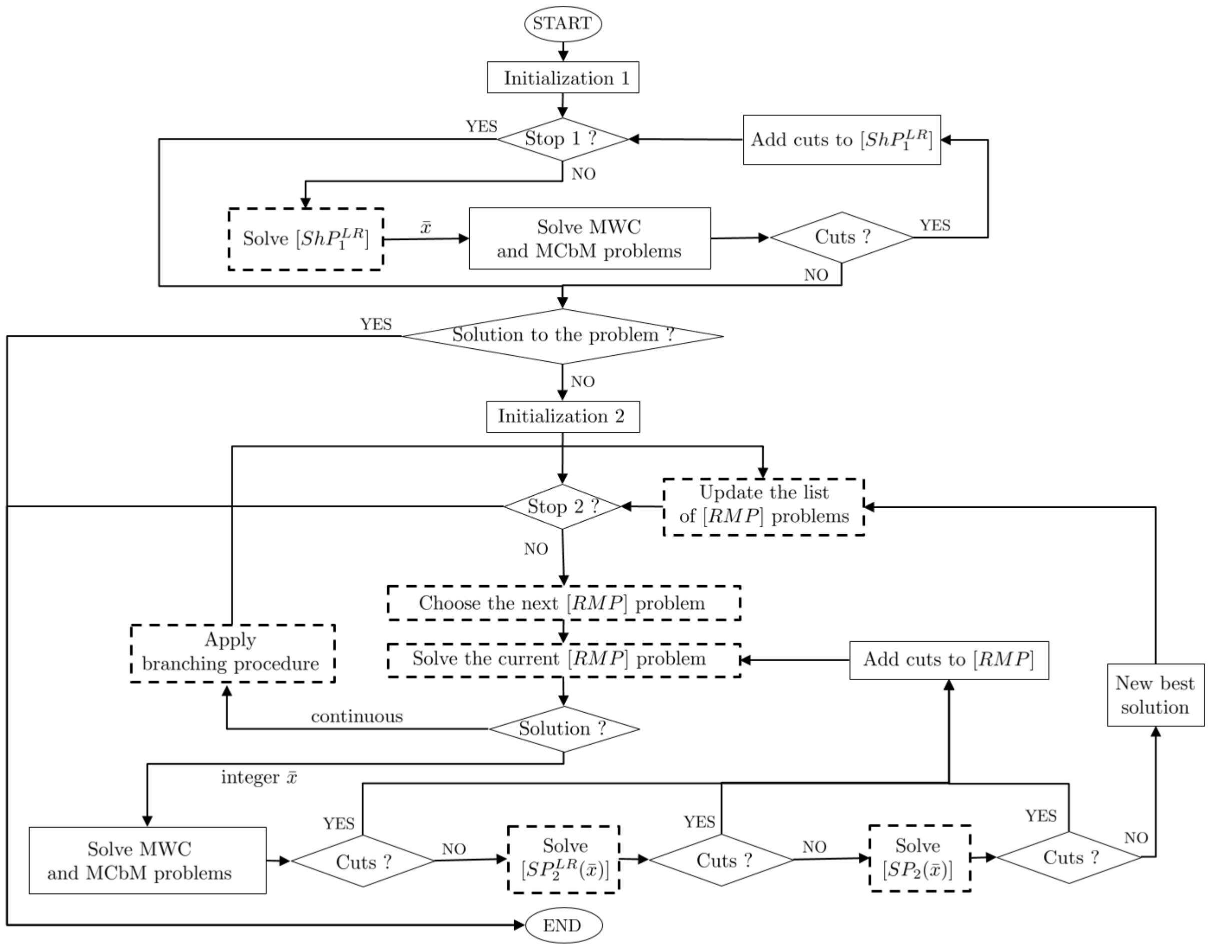
The resolution of formulation $[SP_2^{LR}(\bar{x})]$ gives an optimum value strictly greater than zero (equal to 1). We then identify the violated BF cut (6.41) (which has here the same expression as a CB cut).

$$x_{p_G} + x_{p_H} + x_{p_I} + x_{p_J} \leq 3 \tag{6.41}$$

This example illustrates a case where no MCbM and/or MWC cuts are identified although the technician-to-task assignment sub-problem is infeasible.

6.3 The algorithm: general structure

To efficiently solve the problem while exploiting the decomposition described in section 6.1, one can easily distinguish two different implementation approaches. Indeed, since this decomposition can be seen as a Benders decomposition of the problem, these two different implementations are presented in Section 2.3. Since the formulation of the master problem is a pure ILP model, solving it is very likely to be too time-consuming. Therefore, it seems better suitable to solve the problem with the alternative implementation of the Benders decomposition (see Figure 2.2). Moreover, one drawback of applying the classical implementation (see Figure 2.1) to our problem is that a feasible solution (and optimal) is only obtained at the end, whereas the alternative implementation may provide feasible solutions throughout the resolution of the master problem. We choose to refer to our method as a B&C approach. Although we do not use CP in the method, we choose this terminology since the technician-to-task assignment sub-problem is a feasibility test, and the approximations to this sub-problem yields something rather similar to filtering algorithms. Figure 6.9 outlines the general structure of our two-stage method:



NB: All the steps denoted with a rectangular dash-line box are performed by an ILP solver.

Figure 6.9 – Flow chart of the B&C approach

— Stage 1 (solve a linear relaxation of $[P]$):
The purpose of this first stage is to generate potential useful MCbM and MWC cuts while working with a much easier problem. To this end, we consider $[ShP_1^{LR}]$ the linear relaxation of $[ShP_1]$ ("Initialization 1"). Using a LP solver, we solve this linear relaxation ("Solve $[ShP_1^{LR}]$"). We then solve the approximations to the sub-problem described in Sections (6.2.2) (the maximum cardinality b-matching becomes a fractional maximum cardinality b-matching) and (6.2.3) for the continuous solution $\bar{x}$ of $[ShP_1^{LR}]$ ("Solve MWC and MCbM problems"). If we generate some MCbM and/or MWC cuts, we add them to $[ShP_1^{LR}]$ ("Add cuts to $[ShP^{LR}]$") and we re-optimize this problem. Otherwise, we stop this first stage.
Last but not least, to avoid wasting too much time generating cuts that may not be all useful, we choose arbitrary to stop the resolution ("Stop 1") after the first 100 iterations if it is not stopped sooner ("Cuts ? No"). For efficiency consideration, we also only compute the clique with the maximum-weight in the corresponding approximation to the sub-problem. This comes from

the fact that the graph is larger when considering continuous solutions.

Remark: Note that this stage could have been performed at the root-node of the search tree defined for the second stage. However, our aim is to take advantage of the preprocessing techniques embedded in ILP solvers.

— Stage 2 (Solve the master problem $[P]$):
In the second stage, we first check ("solution to the problem ?") if the solution $\bar{x}^*$ obtained during the previous stage is feasible for $[P]$ (i.e., if $\bar{x}^*$ is integer and leads to a feasible technician-to-task assignment sub-problem). If it is not the case, we solve the master problem by a branch-and-cut method implemented in a commercial solver. We initialize the restricted master problem $[RMP]$ with all the cuts that have been previously generated ("Initialization 2") and we forward it to the ILP solver. As long as the solution computed by the solver is continuous ("Solution ? continuous"), we let it make its own branching decisions to produce integer solutions ("Apply branching procedure"), its own exploration of the search tree ("Choose the next $[RMP]$ problem"), and we let it use its own techniques to compute feasible solutions and generic cuts which help to reduce the list of active nodes ("Update the list of $[RMP]$ problems"). For every integer solution $\bar{x}$ to the current restricted master problem $[RMP]$ ("Solution ? integer"), we check if $\bar{x}$ is feasible regarding the technician-to-task assignment sub-problem. We start by solving the maximum cardinality b-matching and the maximum-weight clique problems ("Solve MWC and MCbM problems"). If it produces at least one MCbM or MWC cut, we discard the current solution $\bar{x}$ by adding the generated cut(s) to $[RMP]$ ("Add cuts to $[RMP]$"). Otherwise, we solve the LP formulation $[SP_2^{LR}(\bar{x})]$ ("Solve $[SP_2^{LR}(\bar{x})]$"). If we identify a violated BF cut of type (6.25), we add it to $[RMP]$ ("Add cuts to $[RMP]$"). Otherwise, we cannot directly conclude to the feasibility of the technician-to-task assignment sub-problem before solving the ILP formulation $[SP_2(\bar{x})]$ ("Solve $[SP_2(\bar{x})]$"). If the solution has a strictly positive cost, we generate a CB cut of type (6.19) and add it to the restricted master problem ("Add cuts to $[RMP]$"). Otherwise, we conclude that $\bar{x}$ is a new feasible solution to our problem ("New best solution"). Note that the branch-and-bound scheme ensures that $\bar{x}$ is strictly better than the best previous solution. We end up this phase ("Stop 2") with the optimal solution to $[P]$ or with a feasible solution if a time limit has been reached. Note that if no solution is found within the time limit, one can always consider the feasible solution in which all the tasks are postponed.

Remark: As described above, the sub-problem is solved at each integer node of the branch-and-bound tree during the second stage. In order to improve the value of the linear relaxation and to potentially speed-up the algorithm (especially in case where integer solutions are scarce), one can also decide to check the approximations to the sub-problem at non-integer-nodes because it may produce cuts that eliminate continuous infeasible solutions. However, adding too many of these cuts to the restricted master problem can decrease the efficiency of the approach, since it is likely to increase the time needed to solve the linear relaxation at each node. In preliminary experiments, we tested this strategy checking the maximum-weight clique and the maximum cardinality b-matching problems at non-integer nodes. We arbitrarily added the cuts whose value of slack variable is greater or equal to 0.5. Additionally, we stopped the search for a cut as soon as the first cut is found. We did not observe any significant improvement of the results and therefore dropped this idea.

6.4 Computational experiments

We implemented our algorithms using *Java 8 (JVM 1.8.0.25)*. We rely on *Gurobi 6.5.1* for solving LP and ILP models. We ran our experiments on a Linux 64 bit-machine, with an Intel(R) Xeon(R) X5675 (3.07Ghz) and 12GB of RAM. We set a 3-hour time limit to solve the different instances (notice that all CPU times are reported in seconds and rounded to the closest integer). In order to assess the quality of our results, we compute the gap with respect to the optimal solution when it is known, or to the best upper bound retrieved by the solver over all our different experiments.

6.4.1 Results of the exact approaches

To quantify the relevance and the contribution of the problem-specific cuts (MCbM and MWC cuts) introduced in Sections 6.2.2 and 6.2.3, we present the computational results of the B&C approach without ($\overline{B\&C}$) and with ($B\&C$) the MCbM and MWC cuts. To be clear, these two approaches only differ on the inclusion of step "Solve MWC and MCbM problems" in the scheme presented in Figure 6.9. For the sake of comparison, we also present the computational results for the direct resolution of the ILP formulation $[P3]$ which seems to give the best results according to Chapter 4.

In Table 6.8, we report the average, over all the instances belonging the same family or sharing a common characteristic, of the gap (Gap), of the solution time (Time), and of the percentage of tasks scheduled in the best solution (%S). We also report the number of optimal solutions found within the time limit (#Opt).

Remark: In order to have a meaningful comparison, the average solution times only takes into account those instances for which an optimal solution has been found within the time limit. Similarly, the average gap and percentage of tasks scheduled takes only into account the instances which are not optimally solved. Indeed, since in our instances, postponing a task is non-profitable and heavily penalized, a large gap is often related to a low percentage of tasks scheduled during the time horizon. This allows a better understanding of the results. Notice that on average 99% of the tasks are scheduled in the optimal or best-known solutions for testbed G1.

First, we observe that the B&C approach outperforms by far the direct resolution of ILP formulations, and this for every family of instances. Indeed we are able to solve to optimality 80% of the instances for testbed G1, in this case, the solution time is importantly reduced. Moreover, the overall average gap when optimality is not reached is small (1.7%).

Second, we can state that the performance of the B&C approach is strongly correlated with the cuts generated from the approximations to the technician-to-task assignment sub-problem. Indeed, including the problem-specific cuts allows us to find the optimal solution on 63 additional instances. On the remaining instances, it also significantly reduces the gap from 4.0%. This highlights the weakness of the Benders cuts and the strength of the problem-specific cuts.

Third, we can draw the same conclusions on the difficulty of the instances than those drawn after having tested the different ILP formulations. The technicians-to-work ratio, the number of time periods per day, and, to a lesser extent, the number of tasks are the most complicating factors. While the impact of the first factor is particularly evident, the difficulty to solve instances with a large number of time periods per day is mainly related to the fact that there are more patterns, which creates more opportunity for a technician to change from one task to another one in the same day. The daily location-based incompatibilities are then more binding. Morerover, although the number of plans is larger when considering more tasks, the number of patterns does not grow proportionally, which results in a moderately more complicated technician-to-task assignment sub-problem.

In Table 6.9, we present a brief description of the average number of cuts generated during the execution of the B&C approach. The average, over all the instances with a common characteristic, of the total number of cuts (#Cuts) is decomposed into the CB, BF, MCbM, and MWC cuts. Detailed results for each family of instances are available in Appendix C.3.

First, we observe that, on average, 90% of the cuts are problem-specific cuts whereas the other 10% are generic cuts. These results naturally show that the approximations are not always able to identify the infeasibility of the technician-to-task assignment sub-problem. However, when $|\mathcal{S}| = 1$, it is noteworthy that solving the maximum cardinality b-matching and maximum-weight clique problems allow almost always to identify the infeasibility of the sub-problem. We observe that we generate more cuts for instances with 3 skills, 4 time periods per day and a tight technicians-to-work ratio. This is due to the largest number of patterns in the first two cases and to the fact that there is less potential configuration to schedule the tasks in the last case. Notice that we never generate CB cuts. Actually, we observe that the optimal solution to the relaxed sub-problem $[SP_2^{LR}(\bar{x})]$ is

Table 6.8 – Detailed computational results on testbed G1 for the different exact approaches.

Family	[P3]				$\overline{\text{B\&C}}$				B&C			
	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time
10_2_1_20_A	-	-	5/5	152	-	-	5/5	2,272	-	-	5/5	**5**
10_2_1_20_B	-	-	5/5	37	-	-	5/5	2	-	-	5/5	**1**
10_2_1_40_A	0.01%	100%	1/5	7	0.70%	100%	2/5	3,379	-	-	**5/5**	7
10_2_1_40_B	-	-	5/5	121	-	-	5/5	8	-	-	5/5	**1**
10_2_3_20_A	-	-	5/5	1,574	1.0%	97%	2/5	2,871	-	-	5/5	**162**
10_2_3_20_B	-	-	5/5	18	-	-	5/5	94	-	-	5/5	**2**
10_2_3_40_A	-	-	5/5	3,996	2.2%	98%	2/5	4,040	-	-	5/5	**17**
10_2_3_40_B	-	-	5/5	295	-	-	5/5	18	-	-	5/5	**1**
20_2_1_40_A	1.4%	98%	2/5	2,858	6.4%	96%	1/5	7,301	-	-	**5/5**	230
20_2_1_40_B	-	-	5/5	2,078	-	-	5/5	695	-	-	5/5	**4**
20_2_1_80_A	334%	20%	0/5	-	8.2%	96%	0/5	-	0.02%	100%	**4/5**	300
20_2_1_80_B	229%	49%	3/5	3,823	0.13%	100%	4/5	580	-	-	**5/5**	5
20_2_3_40_A	1.2%	99%	3/5	5,534	4.6%	95%	1/5	1,136	2.1%	98%	**4/5**	40
20_2_3_40_B	-	-	5/5	155	-	-	5/5	525	-	-	5/5	**3**
20_2_3_80_A	156%	39%	0/5	-	3.3%	98%	0/5	-	-	-	**5/5**	51
20_2_3_80_B	196%	50%	3/5	2,456	0.02%	100%	4/5	163			**5/5**	5
20_4_1_20_A	1.3%	97%	0/5	-	2.1%	95%	0/5	-	2.2%	95%	**3/5**	1,715
20_4_1_20_B	-	-	5/5	405	-	-	5/5	131	-	-	5/5	**2**
20_4_1_40_A	61%	75%	0/5	-	8.6%	93%	-	-	1.2%	98%	**2/5**	1,586
20_4_1_40_B	107%	74%	1/5	1,968	0.3%	100%	4/5	630	-	-	**5/5**	6
20_4_3_20_A	2.4%	95%	4/5	5,005	3.1%	95%	0/5	-	2.0%	95%	**4/5**	237
20_4_3_20_B	-	-	5/5	113	3.1%	95%	4/5	134			5/5	**9**
20_4_3_40_A	5.3%	95%	0/5	-	6.7%	94%	0/5	-	0.85%	98%	**1/5**	8,888
20_4_3_40_B	0.03%	100%	3/5	2,108	0.29%	100%	1/5	1,608			**5/5**	11
40_4_1_40_A	106%	76%	0/5	-	15.7%	89%	0/5	-	**2.1%**	98%	0/5	-
40_4_1_40_B	3.0%	98%	0/5	-	0.7%	100%	0/5	-	-	-	**5/5**	31
40_4_1_80_A	4,948%	0%	0/5	-	16.1%	90%	0/5	-	**1.5%**	99%	0/5	-
40_4_1_80_B	331%	0%	0/5	-	1.8%	99%	0/5	-	-	-	**5/5**	89
40_4_3_40_A	4.6%	96%	0/5	-	14.6%	90%	0/5	-	**1.5%**	99%	0/5	-
40_4_3_40_B	0.84%	99%	2/5	2,118	0.3%	100%	0/5	-	-	-	**5/5**	36
40_4_3_80_A	2,727%	18%	0/5	-	14.3%	90%	0/5	-	**2.3%**	98%	0/5	-
40_4_3_80_B	3,899%	0%	0/5	-	0.96%	100%	0/5	-	-	-	**5/5**	86

Characteristics		[P3]				$\overline{\text{B\&C}}$				B&C			
		Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time
$\lvert\mathcal{S}\rvert =$	1	621%	61%	34/80	1,035	6.7%	95%	36/80	957	1.6%	98%	**64/80**	179
	3	982%	61%	43/80	1,896	4.9%	96%	29/80	722	1.7%	98%	**64/80**	186
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	144%	60%	57/80	1,461	3.9%	97%	51/80	982	1.0%	99%	**78/80**	49
	4	1,014%	61%	20/80	1,757	6.5%	95%	14/80	380	1.7%	98%	**50/80**	390
Type =	A	759%	65%	25/80	2,838	7.8%	94%	13/80	3,106	1.7%	98%	**48/80**	456
	B	802%	53%	52/80	913	0.8%	100%	52/80	288	-	-	**80/80**	18
All		773%	61%	77/160	1,538	5.7%	96%	65/160	852	1.7%	98%	**128/160**	182

most of the time integer although the constraint matrix is not totally unimodular [2]. We can also note that we generate only few MCbM cuts. On the contrary, we generate many MWC cuts. The reason for this is that the restricted master problem has no information about the daily location-based incompatibilities at the beginning of the optimization. It is then more likely that these specific constraints are not satisfied by the solutions to the restricted master problem.

Table 6.9 – Description of the average number of cuts generated in the B&C approach.

Characteristic		#Cuts	CB	Other cuts		
				BF	MCbM	MWC
$\lvert\mathcal{S}\rvert =$	1	102	0	0.3	3	98
	3	224	0	28	10	186
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	56	0	9	3	44
	4	270	0	19	10	241
Type $=$	A	262	0	24	10	229
	B	63	0	4	4	56
All		163	0	14	7	142

Furthermore, we notice that all the components of the B&C approach have a favorable trade-off between their efficiency and the time spent on it. Since the relaxation of the problem considered in the first stage only contains continuous variables and no (or few) cuts, this first stage does not require too much time: on average 1% of the CPU time. Notice also that the limit on the number of iterations during this stage is never reached in our experiments. The results also show that solving the restricted master problem in the B&C approach is the most time-consuming part of the second-stage. Indeed, this component is responsible on average for 99% of the CPU time. This compares with the negligible time spent on solving, for a solution $\bar{x}$ to $[RMP]$, the formulations $[SP_2^{LR}(\bar{x})]$ and $[SP_2(\bar{x})]$ with the commercial solver, or the approximations to the sub-problem.

6.4.2 A cooperative approach

As a second part of our experiments, we tested the use of the CPLNS introduced in Chapter 5 along with the B&C approach. More specifically, the idea is to run the CPLNS in parallel with the algorithm. If the current solution to the CPLNS is better than the best solution found so far, we provide this solution to the solver. If this improves the current lower bound of the ILP solver it may help to prune some nodes in the branch-and-bound tree. This idea comes from the observation that for the large-sized instances the solver has sometimes trouble with finding good quality solutions. Table 6.10 summarizes the results. The last columns report the average gap and the proportion of scheduled tasks of the best solution found by the CPLNS with the same execution time as the B&C approach (i.e., we stop the CPLNS when the B&C approach finds the optimal solution or when it reaches the time limit). More detailed results for each family of instances are available in Appendix C.3. Since no additional instances are solved to optimality, we conclude that running the CPLNS in parallel with the B&C approach has no significant effect on its efficacy (even if the gap is reduced by 0.4% for the instances for which optimality is not reached). Above all, these results allow us to state that, if we are given a 3-hour time limit, the B&C approach significantly outperforms the CPLNS with an average difference of around 2% between the two gaps. It is mainly due to the difficulty of the CPLNS in scheduling some tasks when the availability of the technicians is scarce. It also highlights the fact that the metaheuristic may often be trapped in local optima. However, one may find a smaller gap between the efficiency of the CPLNS and the B&C approach if one imposes another time limit. Lastly, we notice that the characteristics that make the instances difficult to solve by the B&C approach are the same for the CPLNS.

2. We found an instance - not part of testbed G1 - and a solution $\bar{x}$ of $[RMP]$ where the optimal value of $[SP_2^{LR}(\bar{x})]$ is equal to 0 whereas the optimal value of $[SP_2(\bar{x})]$ is equal to 1.

Table 6.10 – Aggregated computational results of the B&C approach coupled with the CPLNS.

Characteristics		B&C			CPLNS	
		Gap	#Opt	Time	Gap[1]	Gap[2]
$\lvert\mathcal{S}\rvert =$	1	1.6%	64/80	183	3.7%	1.2%
	3	1.0%	64/80	195	2.7%	1.0%
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	0.09%	78/80	38	3.7%	0.90%
	4	1.4%	50/80	399	3.2%	1.4%
Type =	A	1.3%	48/80	445	3.2%	1.9%
	B	-	80/80	20	-	0.61%
All		1.3%	128/160	179	3.2%	1.1%

[1] Takes into account the instances where the time limit is reached in the B&C approach.

[2] Takes into account the instances solved to optimality by the B&C approach.

6.5 Conclusions

In this chapter, we have proposed a B&C approach to solve the wind turbine maintenance scheduling problem. This exact method takes advantage of the decomposition of the problem into a task scheduling problem (in which technician considerations have been partially dropped) and a technician-to-task assignment sub-problem. For each selection of plans, we actually check the existence of an assignment of the technicians to the scheduled tasks that copies with the availability of every single technician and that meets the travel limitations imposed on each day. Since the ILP formulation of the sub-problem does not possess the integrity property, we use the concept of combinatorial Benders cuts to invalidate infeasible selection of plans in the restricted master problem, while trying to identify violated classical Benders feasibility cuts beforehand. However the key part of the method comes from the approximations to the technician-to-task assignment sub-problem as a series of maximum cardinality b-matching and maximum-weight clique problems, which help us to build additional cuts. Indeed, according to the experiments that we conducted, these problem-specific cuts proves to be very effective speeding up the convergence of the B&C approach. This latter method finds optimal solutions in short execution times for the large majority of the instances or delivers high-quality integer solutions in those instances in which optimality is not reached. The B&C approach significantly outperforms the direct resolution of ILP models as well as, in a certain context, the CPLNS introduced in Chapter 5.

6.6 Complement: sub-problem and complexity

6.6.1 Equivalence to the L-coloring problem

To assess the complexity of the technician-to-task assignment sub-problem, we prove its equivalence to the L-coloring problem

Definition 6.6.1. ***(L-coloring problem):***
Let $\mathcal{G} = [\mathcal{V}, \mathcal{E}]$ be an undirected graph and assume that each vertex $v \in \mathcal{V}$ is assigned a list $L(v)$ of allowed colors. The L-coloring problem consists in finding a coloring[3] *c such that $c(v) \in L(v)$ for all $v \in \mathcal{V}$. If such a coloring exists, we say that $\mathcal{G}$ is L-colorable.*

First, let us associate a color $color_r$ to every technician $r \in \mathcal{R}$. For a given solution $\bar{x}$ to the restricted master problem $[RMP]$, let us also introduce the undirected graph $\ddot{G}(\bar{x})$ composed of:

— a set of vertices $\ddot{\mathcal{V}}$

 — For each job $j \in \mathcal{J}(\bar{x})$, we add q_j vertices in $\ddot{\mathcal{V}}$. Parameter j_ν denotes the job associated with a vertex $\nu \in \ddot{\mathcal{V}}$ and $\ddot{\mathcal{V}}_j$ the set of vertices associated with job j. Denoting L_ν the set of colors associated with vertex ν, we define $L_\nu = \{color_r\}_{r \in \mathcal{R}_j}$.

3. A coloring of graph $G = [\mathcal{V}, \mathcal{E}]$ is a mapping c: $\mathcal{V} \rightarrow \mathbb{N}$ such that $c(v) \neq c(v')$ for every edge $(v, v') \in \mathcal{E}$

— a set of edges $\ddot{\mathcal{U}}$ defined such that $\forall \nu_1\, \nu_2 \in \ddot{\mathcal{V}}$:
$(\nu_1, \nu_2) \in \ddot{\mathcal{U}} \Leftrightarrow \nu_1 \neq \nu_2 \;\wedge\; \Big(\big(S_{j_{\nu_2}} \leq C_{j_{\nu_1}} \;\wedge\; S_{j_{\nu_1}} \leq C_{j_{\nu_2}} \big) \;\vee\; \sigma_{l_{j_{\nu_1}} l_{j_{\nu_2}}} = 0 \Big)$
(There exists an edge between two vertices ν_1 and ν_2 in $\ddot{G}(\bar{x})$ if and only if a technician cannot be assigned to both jobs j_{ν_1} and j_{ν_2} with regard to constraints [C2] and [C3])

We prove in Proposition 6.6.1 the equivalence between the technician-to-task assignment sub-problem and the L-coloring problem in graph $\ddot{G}(\bar{x})$.

Proposition 6.6.1. *Let $\bar{x}$ be a solution to the restricted master problem $[RMP]$. The technician-to-task assignment sub-problem for $\bar{x}$ is equivalent to the L-coloring problem in graph $\ddot{G}(\bar{x})$.*

Proof. Assume that we know a feasible solution to the technician-to-task assignment sub-problem. This solution directly yields the list $\mathcal{R}_j^{ass}$ of technicians assigned to every job $j \in \mathcal{J}(\bar{x})$ (in that solution). We can then build a solution to the L-coloring problem by iterating through the vertices of $\ddot{G}(\bar{x})$. More specifically, for each vertex $\nu \in \ddot{\mathcal{V}}$, we pick a technician $r \in \mathcal{R}_{j_\nu}^{ass}$ (and remove it from this set) and color the vertex ν with $color_r$. By construction of the graph, we are ensured that for a job j the set $\mathcal{R}_j^{ass}$ becomes empty only when every vertex of $\ddot{\mathcal{V}}_j$ is colored. We are also ensured that the graph is L-colorable since each vertex $\nu \in \ddot{\mathcal{V}}$ picks up an admissible color in L_ν.

Alternatively, assume that we know a solution c to the L-coloring problem in graph $\ddot{G}(\bar{x})$. This solution directly yields the list $\mathcal{R}_j^{ass}$ of technicians assigned to every job $j \in \mathcal{J}(\bar{x})$. More specifically, for each vertex $\nu \in \ddot{\mathcal{V}}$, we add to $\mathcal{R}_j^{ass}$ the technician $r \in \mathcal{R}$ such that $color_r = c(\nu)$. We then induce the underlying assignment of the technicians to the plans selected in the solution to the restricted master problem. By construction of the graph $\ddot{G}(\bar{x})$, the assignments copy with individual technician availability time periods and do not violate the location-based incompatibilities. This produces a feasible solution to the technician-to-task assignment sub-problem. □

Since the graph coloring problem is a specific case of the L-coloring problem, the strong NP-completeness of the former (Jensen and Toft, 2011) implies the strong NP-completeness of the latter. We can therefore state that the technician-to-task assignments sub-problem is NP-complete in the strong sense. It is noteworthy that in the case of interval graphs, the L-coloring problem stays NP-complete (Biro et al., 1992) although the graph coloring problem becomes polynomial.

Hall's condition (see Definition 6.6.2) provides a necessary condition for the existence of a L-coloring in a graph. However, at first sight, the direct use of this condition to derive cuts is computationally intractable as one should solve several maximal independent set problems for each sub-graph of $\ddot{G}(\bar{x})$.

Definition 6.6.2. ***(Hall's condition)***:
Let $\mathcal{G} = [\mathcal{V}, \mathcal{E}]$ be an undirected graph and assume that each vertex $v \in \mathcal{V}$ is assigned a list $L(v)$ of allowed colors. If for each sub-graph $\mathcal{G}^{sub}$ of graph $\mathcal{G}$, we have $\sum_{c \in L(\mathcal{G}^{sub})} \alpha(c, \mathcal{G}^{sub}, L) \geq |\mathcal{G}^{sub}|$ (where $L(\mathcal{G}^{sub}) = \bigcup_{v \in \mathcal{G}^{sub}} L(v)$ and $\alpha(c, \mathcal{G}^{sub}, L)$ denotes the maximum number of independent vertices of $\mathcal{G}^{sub}$ having the color c in their list), then $\mathcal{G}$ is L-colorable.

6.6.2 Special cases of polynomial complexity

We describe below two special cases where the technician-to-task assignment sub-problem is solvable in polynomial time.

— **Assumptions: relaxation of constraints [C3] and [C5]**
The technician-to-task assignment sub-problem is solvable in polynomial time since it is equivalent to a series of maximum cardinality b-matching problems (see Proposition 6.2.1 in Section 6.2.2).

— **Assumptions: $|\mathcal{S}| = 1$; relaxation of constraints [C4]; substitution of constraints [C3] by chaining constraints**

Substituting the constraints [C3] by chaining constraints means that we replace the daily location-based incompatibilities by the following constraints: a technician can work consecutively at locations l and then l' if and only if $\sigma_{ll'} = 1$.

The technician-to-task assignment sub-problem consists in that case in solving a circulation flow problem in graph $GF(\bar{x})$ defined by:

- a set of vertices: $\mathcal{V} = \{\omega\} \cup \mathcal{V}_{start} \cup \mathcal{V}_{end} \cup \{\tau\}$
  - ω: source vertex.
  - τ: sink vertex.
  - $\mathcal{V}_{start}$: a vertex of $\mathcal{V}_{start}$ represents an aggregate node for all the technicians assigned to a job $j \in \mathcal{J}(\bar{x})$. We denote o a vertex of $\mathcal{V}_{start}$ and j_o the job associated with this vertex.
  - $\mathcal{V}_{end}$: a vertex of $\mathcal{V}_{end}$ represents a dispersion node of all the technicians that have performed a job $j \in \mathcal{J}(\bar{x})$. We denote d a vertex of $\mathcal{V}_{end}$ and j_d the job associated with d.
- a set of edges: $(v, v') \in \mathcal{U}$ with minimal capacity $\gamma_{vv'}^{min}$ and maximal capacity $\gamma_{vv'}^{max}$ (if $f_{vv'}$ denotes the flow on edge (v, v'), we must have $\gamma_{vv'}^{min} \leq f_{vv'} \leq \gamma_{vv'}^{max}$). More precisely:
  - $\forall o \in \mathcal{V}_{start}, \forall d \in \mathcal{V}_{end} : (o, d) \in \mathcal{U} \Leftrightarrow j_o = j_d$, $\gamma_{od}^{min} = q_{j_o}$, $\gamma_{od}^{max} = +\infty$
  - $\forall d \in \mathcal{V}_{end}, \forall o \in \mathcal{V}_{start} : (d, o) \in \mathcal{U} \Leftrightarrow start_{j_o} \geq end_{j_d} \wedge (d_{start_{j_o}} \neq d_{end_{j_d}} \vee \sigma_{l_{j_o}, l_{j_d}} = 1)$, $\gamma_{do}^{min} = 0$, $\gamma_{do}^{max} = +\infty$
  - $\forall o \in \mathcal{V}_{start} : (\omega, o) \in \mathcal{U}$, $\gamma_{\omega o}^{min} = 0$, $\gamma_{\omega o}^{max} = +\infty$
  - $\forall d \in \mathcal{V}_{end} : (d, \tau) \in \mathcal{U}$, $\gamma_{d\tau}^{min} = 0$, $\gamma_{d\tau}^{max} = +\infty$
  - $(\tau, \omega) \in \mathcal{U}$, $\gamma_{\tau\omega}^{min} = 0$, $\gamma_{\tau,\omega}^{max} = |\mathcal{R}|$

Figure 6.10 shows the general structure of the graph $GF(\bar{x})$.

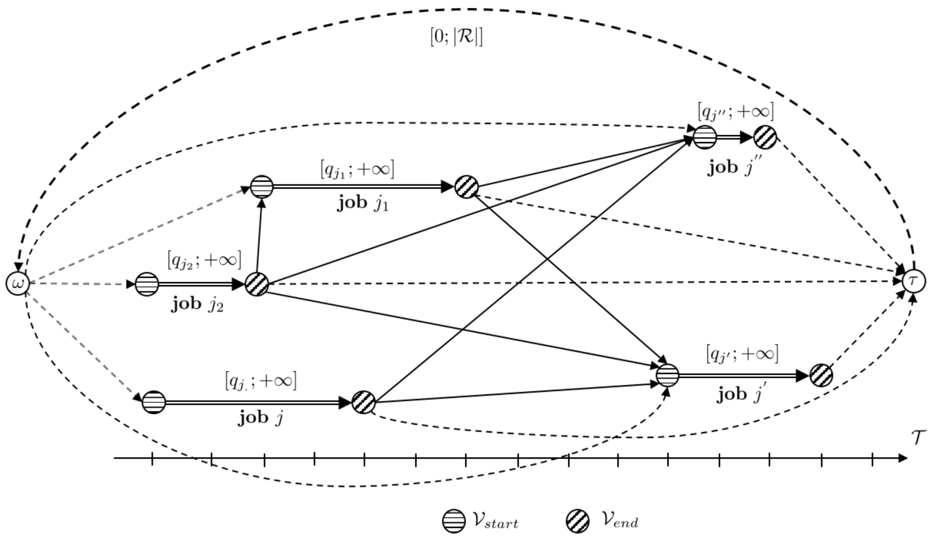


Figure 6.10 – Structure of graph $GF(\bar{x})$

Solving this circulation flow problem is equivalent to solving a maximum flow problem in a modified graph where edges from the source and the sink are added to remove the lower bound on the flow associated with some edges. The technician-to-task assignment sub-problem is therefore solvable in polynomial time.

Chapter 7

Maximizing availability vs maximizing revenue: a short comparative study

When good quality wind forecasts are unavailable (usually when the length of the planning horizon increases), maximizing the revenue generated by the electricity production of the turbines is not longer possible. As an alternative objective, we can maximize the availability of the turbines (i.e., minimize their total downtime). This objective is actually the most common criterion used by maintenance schedulers, since the large majority of the maintenance contracts are based on the annual availability factor (AF) of the turbines. Nonetheless, it should be observed that a short downtime for the wind turbines does not necessarily mean a negligible lost electricity production (although these two elements are inextricably linked), and therefore minimizing the former is not equivalent to minimizing the latter. This chapter aims to quantify this difference.

This alternative objective of maximizing the availability of the turbines requires only a minor adjustment to the problem defined in Chapter 3, and it can be easily handled by the models and the solutions methods defined in Chapters 4 and 6. Indeed, one needs to modify only the value of the parameters used in the objective function. Let δ be the number of hours in a time period of $\mathcal{T}$. Since time periods in $\mathcal{T}$ have identical length, we set each parameter g_w^t to δ for every wind turbine w. Let $\widetilde{\delta}^d$ be the number of hours in the rest time period following day d. For every day d and every turbine w, parameter $\widetilde{g}_w^d$ takes the value of $\widetilde{\delta}^d$.

We conducted experiments only on the type B instances of testbed G1 since we have been able to compute the optimal solution for each of them. Tables 7.1 and 7.2 summarize the results of the ILP formulation $[P3]$ and the B&C approach with the two different objective functions: maximizing the revenue generated by the wind electricity production (max R) and maximizing the availability of the turbines (max A). For the direct resolution of the ILP formulation $[P3]$, more instances are solved to optimality within the 3-hour time limit when maximizing the availability of the turbines. For the B&C approach, results appear to be independent of the selected objective function. First, one should notice that the problem contains far more symmetries when maximizing the availability of the turbines. Indeed, the value of the objective function is almost only based on the duration of each task but not on the time periods during which these tasks are scheduled. This also makes the problem a bit easier to solve because a lot of maintenance plans have the same objective value. Although directly solving the ILP formulation provides an acceptable solution approach, it is still significantly outperformed by the B&C approach. Last but not least, the difficulty of the instances seem to be related to the same characteristics for the two different objective functions (the number of time periods per day and, to a lesser extent, the number of tasks). We did not conduct our experiments on the type A instances (part of testbed G1), but it is highly likely that they will still remain harder to solve. Indeed, the difficulty of solving these instances is though to be related to the tightness of the technicians-to-work ratio which does not depend at all on the selected objective function.

Table 7.1 – Detailed computational results on testbed G1 (type B) for the ILP formulation $[P3]$ according to the selected objective function.

Family	**[P3] - max R**				**[P3] - max A**			
	Gap	**%S**	**#Opt**	**Time**	**Gap**	**%S**	**#Opt**	**Time**
10_2_1_20_B			5/5	47			5/5	1
10_2_1_40_B			5/5	93			5/5	10
10_2_3_20_B			5/5	23			5/5	1
10_2_3_40_B			5/5	417			5/5	13
20_2_1_40_B			5/5	2,120			5/5	33
20_2_1_80_B	456%	0%	4/5	4,584			5/5	946
20_2_3_40_B			5/5	143			5/5	16
20_2_3_80_B	7,848%	49%	3/5	3,982			5/5	220
20_4_1_20_B			5/5	358			5/5	11
20_4_1_40_B	103%	60%					5/5	244
20_4_3_20_B			5/5	111			5/5	9
20_4_3_40_B	1.4%	99%	3/5	2,513			5/5	374
40_4_1_40_B	57%	79%					5/5	2,457
40_4_1_80_B	314%	0%			172%	75%		
40_4_3_40_B	2.9%	97%	2/5	2,920			5/5	3,560
40_4_3_80_B	10,333%	0%			149%	61%	2/5	5,469

Characteristic		**[P3] - max R**				**[P3] - max A**			
		Gap	**%S**	**#Opt**	**Time**	**Gap**	**%S**	**#Opt**	**Time**
$\lvert\mathcal{S}\rvert =$	1	176%	43%	24/40	1,309	172%	75%	35/40	529
	3	5,614%	49%	28/40	1,028	149%	61%	37/40	862
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	5,384%	33%	37/40	1,202			40/40	155
	4	2,162%	47%	15/40	1,048	163%	70%	32/40	1,382
All		2,507%	46%	52/80	1,158	163%	70%	72/80	700

Table 7.2 – Detailed computational results on testbed G1 (type B) for the B&C approach according to the selected objective function.

Family	**B&C - max R**		**B&C - max A**	
	#Opt	**Time**	**#Opt**	**Time**
10_2_1_20_B	5/5	9	5/5	1
10_2_1_40_B	5/5	2	5/5	2
10_2_3_20_B	5/5	3	5/5	1
10_2_3_40_B	5/5	2	5/5	2
20_2_1_40_B	5/5	4	5/5	7
20_2_1_80_B	5/5	6	5/5	15
20_2_3_40_B	5/5	4	5/5	8
20_2_3_80_B	5/5	6	5/5	19
20_4_1_20_B	5/5	2	5/5	4
20_4_1_40_B	5/5	7	5/5	11
20_4_3_20_B	5/5	15	5/5	5
20_4_3_40_B	5/5	13	5/5	17
40_4_1_40_B	5/5	27	5/5	36
40_4_1_80_B	5/5	173	5/5	68
40_4_3_40_B	5/5	38	5/5	68
40_4_3_80_B	5/5	72	5/5	174

Characteristic		**B&C - max R**		**B&C - max A**	
		#Opt	**Time**	**#Opt**	**Time**
$\lvert\mathcal{S}\rvert =$	1	40/40	27	40/40	18
	3	40/40	19	40/40	37
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	40/40	3	40/40	7
	4	40/40	43	40/40	48
All		80/80	23	80/80	27

More importantly, as a second part of our tests, Table 7.3 compares the optimal solutions obtained considering the two objectives functions separately. More precisely, let us denote mp^A and mp^R the maintenance plans obtained when maximizing the availability of the turbines and when maximizing the revenue generated by their electricity production. We then compute for each instance two metrics, ΔA and ΔR, defined as follows:

$$\Delta A = \frac{A(mp^A) - A(mp^R)}{A(mp^R)} \qquad \Delta R = \frac{R(mp^A) - R(mp^R)}{R(mp^R)}$$

where $A(.)$ and $R(.)$ are functions giving the total availability of the turbines and the total revenue generated by the wind electricity production in a maintenance plan. Metrics ΔA and ΔR give the relative deviation of the solutions obtained using one objective function when evaluated against the other.

From Table 7.3, we observe that the maintenance plans obtained by maximizing the revenue yield a total availability for the turbines that is close to the optimal one (a gap of 0.20% on average). On the opposite, the maintenance plans obtained by maximizing the availability of the turbines yield on average a total revenue that is quite far from the optimal (a gap of 5.76% on average). In that case, notice that it is likely that many symmetric solutions exist, which can make this value smaller or larger if one considers alternative solutions. However, these results highlight the fact that solving the problem with this alternative objective does not guarantee the highest revenue. Although highly suspected, this short comparative study proves the validity and the relevance of maximizing the revenue when solving the short-term maintenance scheduling problem in the onshore wind power industry.

Table 7.3 – Comparison of the optimal solutions for the two objective functions (testbed G1, type B).

Family	**ΔA**	**ΔR**
10_2_1_20_B	0.07%	9.89%
10_2_1_40_B	0.09%	6.76%
10_2_3_20_B	0.19%	8.91%
10_2_3_40_B	0.00%	10.45%
20_2_1_40_B	0.68%	4.09%
20_2_1_80_B	0.35%	3.30%
20_2_3_40_B	0.25%	4.03%
20_2_3_80_B	0.34%	3.98%
20_4_1_20_B	0.21%	8.46%
20_4_1_40_B	0.10%	5.93%
20_4_3_20_B	0.23%	7.60%
20_4_3_40_B	0.11%	5.77%
40_4_1_40_B	0.05%	3.28%
40_4_1_80_B	0.23%	3.00%
40_4_3_40_B	0.18%	3.75%
40_4_3_80_B	0.16%	2.99%

Characteristic		**ΔA**	**ΔR**
$\lvert\mathcal{S}\rvert =$	1	0.22%	5.59%
	3	0.18%	5.94%
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} =$	2	0.25%	6.43%
	4	0.16%	5.10%
All		0.20%	5.76%

Chapter 8

A robust approach to tackle the stochastic problem

In the problem described in Chapter 3, we assume perfect knowledge of the wind speed during the whole time horizon. We develop several solution methods to address this deterministic problem in Chapters 4, 5, and 6. In practice however, wind speed forecasts are subject to a certain degree of uncertainty. In this chapter, we propose a robust optimization approach to tackle this uncertainty. We introduce a budgeted uncertainty set with additional constraints to deal with the possible spatial and time-wise correlation of the wind speed. We solve the robust problem using a cutting plane method built on top of the decomposition technique designed in Chapter 6 for the deterministic version of the problem. We report computational results on the type B instances of testbed G1.

The chapter is organized as follows. Section 8.1 provides a short overview on wind uncertainty and, more precisely, on wind prediction techniques. This section provides the background needed to understand the choice of working with robust optimization as a valid modeling paradigm to our problem. In order to take into consideration the wind uncertainty, Section 8.2 presents additional elements in the definition of our maintenance scheduling problem. Section 8.3 describes how we model this uncertainty. Section 8.4 defines the robust counterpart of the problem and Section 8.5 presents the solution method to solve it. Section 8.6 reports computational experiments. Section 8.7 concludes this chapter.

8.1 Introduction

Dealing with a short-term maintenance scheduling problem rising in the onshore industry, we aim to find a maintenance plan that maximizes the revenue generated by the electricity production of the turbines while ensuring their safe operation. To avoid computing a maintenance plan that becomes infeasible or/and whose revenue is not optimal when the wind speed deviates from its forecasted values, we need to explicitly address this uncertainty when solving the problem.

To specifically deal with uncertainty, one can distinguish two main methodologies: stochastic programming and robust optimization (see Section 2.5 for more details).

To define the most suitable approach to our problem, we first examine how wind speed can be modeled. In practice, the frequency of wind speeds at a particular wind farm are modeled using different probability distributions. Probably the most commonly-used distribution is the Weibull distribution. Another distribution that is often used by wind power companies is the Raleigh distribution. Indeed, since the shape factor of the Weibull distribution is usually close to 2, a Raleigh distribution with the same scale factor as the Weibull distribution is an acceptable approximation often employed by wind power systems companies. Gamma and Lognormal distributions are also sometimes considered. As an alternative to these univariate distributions, more complex distributions have been proposed. For example, multivariate distributions have been produced to jointly fit wind speed and wind direction. For the purpose of accuracy, Zhang et al. (2013) derived a multivariate and multimodal wind distribution for wind speed, wind direction, and air density.

One of the primary uses of finding an accurate distribution to model the frequency of the wind speed is to estimate the number of hours per year that certain wind speeds are likely to be observed, a key input to estimate the total power output of a wind turbine per year. This information is vital when assessing the wind resource potential when looking for potential for candidates areas for building wind farms. However these wind speed frequency distributions are ineffective for wind speed prediction in a short-term horizon because they neglect time-wise correlations, as well as as some likely spatial correlations.

Wind predictions techniques are based on: physical models, statistical models, artificial neural-network, fuzzy-logic, and hybrid (Wang et al., 2011; Lawan et al., 2014). Schematically, the physical models use numerical weather prediction models (based for example on kinematic equations) to provide wind forecasts with a particular spatial resolution, and they then use some topological information to refine these forecasts. They take into account multiple meteorological variables such as temperature, pressure, and humidity. On the contrary, statistical models work with historical data along with on-line measures. The prediction techniques then rely on multivariate time series, autoregressive–moving average models, and Gaussian copulas (the list is not exhaustive). The application of artificial neural networks or fuzzy logic is also very popular since they can handle very easily non linear relations among the inputs. Hybrid models have also been developed to take advantage of the strength of multiple approaches. We can classify the predictions into two categories according to their outputs: a single value (*point forecasts*) or an interval (*interval forecasts*). The former are largely developed whereas the latter are juts beginning to receive more attention (Qin et al., 2015).

Meanwhile, scenario forecasting of wind power generation is very suitable for decision-making. The meteorological community obtains scenarios, called ensemble forecasts or ensemble-based time-trajectories, from the perturbation of initial conditions and parameters in their numerical weather prediction models. However, linking probabilities to these scenarios seems quite unrealistic. Indeed, even a simple equiprobability assumption among the scenarios may be not verified in practice. The generation of wind speed scenarios have recently received increasing attention (Morales et al., 2010; Pinson and Girard, 2012). We refer the reader to (Zhang et al., 2014) for a survey on these techniques.

Despite a significant amount of work devoted to prediction techniques and scenario generation, they suffer from some drawbacks: curse of dimensionality when estimating the parameters, need of an extensive amount of data, and difficulty to characterize the interdependence structure of the wind speed.

Taking into account the difficulty to fit probability distributions to wind speed or to generate valid scenarios and their probability of occurrence, we decided to apply robust optimization to our problem. Indeed, robust optimization requires limited information on the uncertainty. We can then rely on the previous wind predictions techniques to provide us with the required data. Moreover, a robust approach suits well to our problem as we aim to take risk-averse decisions. We actually aim: i) to guarantee that the maintenance plan can be executed independently of the actual wind speed and ii) to build a maintenance plan yielding reasonably good revenue even if the wind deviates from its nominal value. These two concerns are even stronger if we take into account that a maintenance plan is executed only once.

8.2 Additional elements in the problem definition

In order to take into consideration the wind uncertainty, we provide some additional elements in the definition of the problem described in Chapter 3. For the sake of convenience, we introduce the totally ordered set $\mathcal{T}^+$ formed by the union of the totally ordered set $\mathcal{T}$ and the set of rest time periods that occur between each day. More specifically, the set is built by adding every rest time period at its natural position in the time horizon $\mathcal{T}$(see Figure 8.1). Introducing $\mathcal{T}^+$ allows us to simplify the tracking of wind turbine availability. It also allows us to avoid overloading the chapter with new notation. The set $\mathcal{T}$ is still used for some constraints, since the tasks can only be performed during this particular time periods and the technicians constraints are also only related to those time periods.

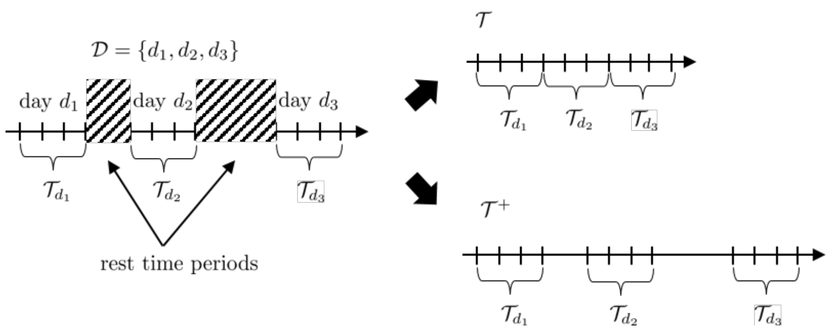


Figure 8.1 – Construction of $\mathcal{T}^+$ from $\mathcal{T}$ and $\mathcal{D}$

To model uncertainty we need to introduce additional parameters. We denote Υ_i the maximal wind speed allowed to perform task $i \in \mathcal{I}$. Note that this parameter is used in the deterministic problem to build vector $\ddot{\vartheta}_i$, which contains the information on the possibility of executing every task i. We also introduce function $g_w^t(\phi)$ which gives the revenue that may be generated by turbine $w \in \mathcal{W}$ during time period $t \in \mathcal{T}$ if the wind speed is equal to $\phi \geq 0$. More specifically, we assume that $g_w^t(\phi)$ is equal to the product of four elements: the selling price of 1kWh of wind power, the nominal power P_w (in kWh) of turbine w, the number of hours $hours(t)$ in time period t, and the capacity factor $CF(\phi)$ of w associated with the wind speed ϕ.

$$g_w^t(\phi) = 0.08 \cdot P_w \cdot hours(t) \cdot CF(\phi) \tag{8.1}$$

Last but not least, we make the following assumption.

Assumption 8.2.1. *Thereafter, we assume that the wind speed at a particular location during a particular time period is equal for all the turbines.*

8.3 Definition of the uncertainty considered in the problem

Uncertainty on the wind speed impacts the revenue generated by the turbines and can jeopardize the execution of some tasks due to safety concerns. We choose here to work with row-wise uncertainty (see Section 2.5). More specifically, we aim to protect independently each task against a potential cancellation due to strong wind, and we want to take a risk averse decision regarding the revenue by maximizing its value in the wort-case.

Every robust approach is based on the definition of the set of potential values for all the uncertain parameters. In the remainder of this section, we define two different type of uncertainty sets whether we take into account the potential correlations in the wind speed or not. Notice that Assumption 8.2.1 makes sufficient to only consider one wind speed for every time period and every location when modeling the uncertainty set.

8.3.1 Non-correlated uncertainty set

In this section, we represent the uncertainty of the problem using an adaptation of the polyhedral approach introduced by Bertsimas and Sim (2003) – see Section 2.5 for more details. We denote ϕ_{tl} the uncertain wind speed at location $l \in \mathcal{L}$ during time period $t \in \mathcal{T}^+$. Each uncertain parameter ϕ_{tl} belongs to the interval $[\widehat{\phi}_{tl} - \phi_{tl}^-, \widehat{\phi}_{tl} + \phi_{tl}^+]$ with ϕ_{tl}^-, $\phi_{tl}^+ \geq 0$. It is noteworthy that we do not make any assumptions on the probability distribution of each uncertain parameter, except that we assume that $\widehat{\phi}_{tl}$ represents its nominal value. To avoid overprotecting the system, we control the level of robustness of the solution by bounding independently the number (or amount) of changes from the nominal values among all the uncertain parameters for each time period $t \in \mathcal{T}^+$ by Γ_t. Each parameter Γ_t is a so-called *uncertainty budget*. The value of Γ_t is fixed for each time period by decision-makers according to their trade off preferences between robustness and performance (in our case the revenue). Notice that we choose to bound the deviations by time period to prevent inappropriate cases where all the deviations happen at the beginning of the time horizon. The deviations are more likely to be spread out over the time horizon. The uncertainty set is defined by:

$$\Phi^{NC}(\Gamma) = \{\phi \,|\exists(\delta^-, \delta^+) \in \Delta^{NC}(\Gamma),\ \phi_{tl} = \widehat{\phi}_{tl} - \delta^-_{tl}\phi^-_{tl} + \delta^+_{tl}\phi^+_{tl} \ \wedge \ \phi_{tl} \geq 0 \ \ t \in \mathcal{T}^+, l \in \mathcal{L}\} \tag{8.2}$$

and

$$\Delta^{NC}(\Gamma) = \left\{ \ (\delta^-, \delta^+) \ \middle| \ \begin{array}{l} \delta^-_{tl} \geq 0,\ \delta^+_{tl} \geq 0 \ \ t \in \mathcal{T}^+, l \in \mathcal{L}, \\ \|\delta^+_{t.} + \delta^-_{t.}\|_1 \leq \Gamma_t \wedge \|\delta^+_{t.} + \delta^-_{t.}\|_\infty \leq 1 \ \ t \in \mathcal{T}^+ \end{array} \right\} \tag{8.3}$$

One may observe that if $\exists t \in \widehat{\mathcal{T}}\ s.t.\ \Gamma_t < |\mathcal{L}|$ this excludes those cases where every uncertain parameter simultaneously takes a value far from its nominal value. However, they can be presumed rare from a practical perspective. If for every time period t, $\Gamma_t = |\mathcal{L}|$, we go back to the approach of Soyster (1973). In that case, we can just remove all the plans that are not feasible in the worst-case scenario (i.e. we fix $x_p = 0, \forall p \in \mathcal{P}^0, \forall t \in \mathcal{T}\ s.t.\ S_p \leq t \leq C_p, \widehat{\phi}_{tl_p} + \phi^+_{tl_p} > \Upsilon_{i_p}$) and we simply associate with each time period t of availability of a turbine w the revenue $\min\left(g^t_w(\widehat{\phi}_{tl_w} - \phi^-_{tl_w}), g^t_w(\widehat{\phi}_{tl_w} + \phi^+_{tl_w})\right)$ generated when one considers the worst-case scenario for the wind speed [1]. This approach is known to be overly conservative. To assess the conservatism of our method, we show in our experiments how our robust solutions evolves when varying the value of the vector Γ.

8.3.2 Correlated uncertainty set

When defining the uncertainty set $\Phi^{NC}(\Gamma)$, we implicitly assume that the wind speed at a location during a particular time period is independent on the past wind speed and the wind observed at the other locations. More realistically, it is very likely that some correlations exist between the wind speed at locations that are very closed to each other. Moreover, there can be some correlations between the wind speed observed during time period t and that observed during previous time periods. We arbitrarily assume that the wind speed at location l during time period t depends on the wind speed at locations that are within a distance of l lower than D^{max}, and on the wind speed at l during the k time periods previous to t. As a result, we do no longer work directly with intervals for every wind speed. We now consider two new positive variables $\ddot{\delta}_{tl}$ and $\ddot{\epsilon}_{tl}$ to describe the deviations of the wind speeds from their nominal values. We then define the correlated uncertainty set in a way that is largely inspired from the developments of Lorca and Sun (2014), who addressed the uncertainty of the wind power for the economic dispatch of power systems.

$$\Phi^C(\Gamma) = \{\phi \,|\exists((\ddot{\delta}, \ddot{\epsilon}) \in \Delta^C(\Gamma),\ \phi_{tl} = \widehat{\phi}_{tl} + \ddot{\delta}_{tl} \ \ \wedge \ \ \phi_{tl} \geq 0 \ \ t \in \mathcal{T}^+, l \in \mathcal{L}, \ \} \tag{8.4}$$

and

$$\Delta^C(\Gamma) = \left\{ \begin{array}{l} (\ddot{\delta}, \ddot{\epsilon}) \,| \\ \ddot{\delta}_{tl} = \sum\limits_{t'=\max(0,t-k)}^{t-1} \alpha_{tt'}\ddot{\delta}_{t'l} + \sum\limits_{l' \in \mathcal{L}} \beta_{ll'}\left(-\delta^-_{tl}\ddot{\epsilon}^-_{tl'} + \delta^+_{tl}\ddot{\epsilon}^+_{tl'}\right) \ \ t \in \mathcal{T}^+, l \in \mathcal{L}, \\ \|\ddot{\epsilon}^-_{t.} + \ddot{\epsilon}^+_{t.}\|_1 \leq \Gamma_t,\ \|\ddot{\epsilon}^-_{t.} + \ddot{\epsilon}^+_{t.}\|_\infty \leq 1 \ \ t \in \mathcal{T}^+\} \end{array} \right\} \tag{8.5}$$

1. According to the CF function represented in Figure 8.2, the minimum production of a turbine $w \in \mathcal{W}$ during time period $t \in \mathcal{T}$ can only occur at the lower bound and upper bound of the interval describing all the possible wind speed at location l_w during time period t.

with $\beta_{ll'} = \dfrac{\max(0, D^{max} - D_{ll'})}{\sum_{l' \in \mathcal{L}} \max(0, D^{max} - D_{ll'})}$ ($D_{ll'}$ denotes the distance between the locations l and l') and $\alpha_{tt'} < 1$ denote the coefficient of time correlation between the deviation of the wind speed during time periods t' and t ($t > t'$).

In the following sections, we use $\Phi(\Gamma)$ to refer either to $\Phi^{NC}(\Gamma)$ or $\Phi^{C}(\Gamma)$. We point out when these latter sets require more individual attention.

8.4 Robust counterpart of the problem

Solving the problem using a robust approach and considering row-wise uncertainty implies that: i) we want to build a maintenance plan that is associated with the largest revenue in the worst case scenario (defined by the uncertainty set) of the wind speed and ii) we want to ensure that every task (except those set to be postponed) can be effectively performed for all the realizations of the uncertainty in $\Phi(\Gamma)$. Indeed, from a practical perspective, it is quite irrelevant to schedule a task during time periods where there is a probability that the wind speed will not allow its execution. Although it may be seen as a conservative decision, it is noteworthy that the two worst-case scenarios for the uncertainty (minimal value of the revenue and the impossibility of execution of some tasks) are essentially opposed. Indeed, the revenue is almost always minimal when the wind speed is lower than its nominal value, whereas some tasks cannot be performed anymore only when the wind speed is higher than its nominal value. This means that generally we do not impact the revenue by preventing tasks to be scheduled during time periods where the safety limit can be exceeded.

We choose to express the robust counterpart of the problem based on the deterministic formulation $[P2]$ described in Chapter 4. One can notice that the revenue is based on the CF function which is not linear. In order to formulate the problem using ILP, we approximate the CF function with a piecewise linear function – denoted $\widetilde{CF}(.)$ – on the interval $[0; \Upsilon^{max}]$ where Υ^{max} is a maximal wind speed that cannot be reached in all the locations we consider in our problem (see Figure 8.2). We use 9 points for this approximation and we denote $\dot{\phi}_k$ the wind speed associated with the k-th point. We have $\dot{\phi}_0 = 0$ and as $\dot{\phi}_9 = \Upsilon^{max}$. For every segment between points k and $k+1$, we also denote a_k the slope (i.e., $a_k = \dfrac{CF(\dot{\phi}_{k+1}) - CF(\dot{\phi}_k)}{\dot{\phi}_{k+1} - \dot{\phi}_k}$) and as b_k the y-intercept (i.e., $b_k = CF(\dot{\phi}_k) - a_k\dot{\phi}_k$). Note that the function is convex on the interval $[\phi_0; \phi_5]$, concave on the interval $[\phi_5; \phi_7[$, and again convex on the interval $[\phi_7; \phi_9]$. We have the wind-speeds VWI, WR and WCO equal to $\dot{\phi}_2$,$\dot{\phi}_6$, and $\dot{\phi}_7$, respectively.

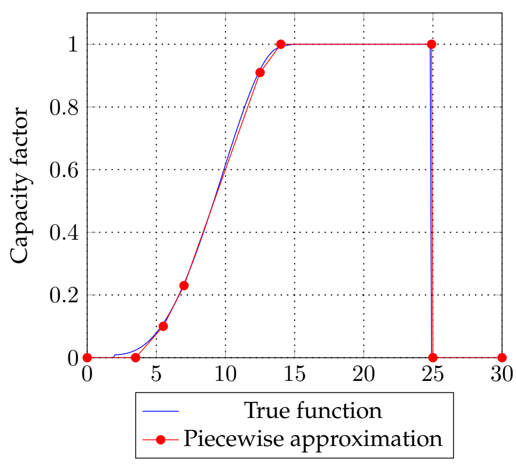


Figure 8.2 – Capacity factor of a wind turbine according to the wind speed

The robust counterpart $[RP(\Gamma)]$ of the problem (hereafter referred to simply as the robust problem) reads:

$$[RP(\Gamma)] \quad \max - \sum_{p \in \mathcal{P}} o_p x_p + \min_{\phi \in \Phi_\Gamma} 0.08 \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \widetilde{CF}(\phi_{tl}) \sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \tag{8.6}$$

subject to:

$$\phi_{l_i}^t \sum_{p \in \mathcal{P}_i} a_p^t x_p \leq \vartheta_{i_p}^t \Upsilon_i \;\; \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \forall \phi \in \Phi_\Gamma, \tag{8.7}$$

$$\sum_{p \in \mathcal{P}_i} x_p = 1 \;\; \forall i \in \mathcal{I}, \tag{8.8}$$

$$\sum_{i \in \mathcal{B}} \sum_{p \in \mathcal{P}_i} a_p^t x_p \leq 1 \;\; \forall \mathcal{B} \in ov(\mathcal{I}), \forall t \in \mathcal{T}, \tag{8.9}$$

$$f_w^t + \sum_{p \in \mathcal{P}_i} b_{wp} a_p^t x_p \leq 1 \;\; \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}^+, \tag{8.10}$$

$$\sum_{i \in \mathcal{I} | s_i \in \bar{\mathcal{S}}} \sum_{p \in \mathcal{P}_i} a_p^t q_p x_p \leq |R_{\bar{\mathcal{S}}}^t| \;\; \forall t \in \mathcal{T}, \forall \bar{\mathcal{S}} \subset \mathcal{S}, \tag{8.11}$$

$$\sum_{r \in \mathcal{R}_h} y_{rh} = \sum_{p \in P_h} q_p x_p \;\; \forall h \in \mathcal{H}, \tag{8.12}$$

$$\sum_{h \in \mathcal{H}_l} a_h^t y_{rh} \leq \rho_r^t v_{rl}^t \;\; \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}, \tag{8.13}$$

$$\sum_{l \in \mathcal{L}} v_{rl}^t = 1 \;\; \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \tag{8.14}$$

$$v_{rl_r^t}^t = 1 \;\; \forall r \in \mathcal{R}, \forall t \in \mathcal{T} s.t. \rho_r^t = 0, \tag{8.15}$$

$$v_{rl}^t + \sum_{l' \in \mathcal{L} | \sigma_{ll'} = 0} v_{rl'}^{t'} \leq 1$$
$$\forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \forall (t, t') \in \mathcal{T}_d \times \mathcal{T}_d, t \neq t', \forall l \in \mathcal{L}, \tag{8.16}$$

$$x_p \in \{0, 1\} \;\; \forall p \in \mathcal{P}, \tag{8.17}$$

$$f_w^t \in \{0, 1\} \;\; \forall w \in \mathcal{W}, \forall t \in \mathcal{T}^+, \tag{8.18}$$

$$y_{rh} \in \{0, 1\} \; \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h \tag{8.19}$$

$$v_{rl}^t \in \{0, 1\} \;\; \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T} \tag{8.20}$$

It is worth noting that for producing consistent decisions we only make use of one uncertainty set for the whole problem. The objective in (8.6) is defined as the difference between the minimal revenue that can be generated by the wind turbines according to Φ_Γ and the penalties related to the postponement of some tasks. Constraints (8.7) ensure that each scheduled task can be perform for any wind speed in the uncertainty set Φ_Γ. They also impose that the maintenance plan meets the external restrictions (e.g., reliability concerns, contract commitments). Although the other constraints have been defined first in Section 4.2, we describe them again for the sake of clarity. Constraints (8.8) ensure that at least one plan involving each task is selected (i.e., each task is either executed or postponed). Constraints (8.9) are the non-overlapping constraints. Constraints (8.10) couple electricity production variables to maintenance plan variables. We have the cumulative scheduling constraints (8.11). Constraints (8.12) ensure that the technician requirements are fulfilled. Constraints (8.13) couple the locations of the technicians to the tasks they perform. Constraints (8.14), (8.15), and (8.16) are used to handle the daily location-based incompatibilities and the availability calendars of the technicians. Finally, constraints (8.17)-(8.20) state the binary nature of the decision variables.

8.5 Solution method

Since the CF function is not linear, we cannot reformulate the robust problem using duality theory to generate a linear problem equivalent to $[RP(\Gamma)]$ (see Section 2.5). As an alternative, a cutting plane method appears to be an appropriate method to solve the robust problem.

8.5.1 Problem reformulation

First, let us start by presenting a reformulation of the robust counterpart of the problem more adapted to a cutting plane approach. To this end, we associate with each time period of availability of a wind turbine the maximum revenue that it can generate. From that, we modify the objective function such that its value is always an upper bound on the true objective value. We denote $\widetilde{CF}_{tl}^{max} = \max_{\phi_{tl} \in \Phi(\Gamma)} CF(\phi_{tl})$ the maximum capacity factor that can be reached at location l during time period t. Notice that setting $\widetilde{CF}_{tl}^{max} = CF\left(\hat{\phi}_{tl}\right)$ is also a valid approach since, in the worst-case, the revenue is at most equal to the revenue considering the nominal value of the wind. We then introduce variable θ representing the difference between the maximum revenue that can be generated according to a fixed maintenance plan and the true value of the revenue in the worst case scenario. The robust problem can be rewritten as follows:

$$[RP(\Gamma)] \quad \max - \sum_{p \in \mathcal{P}} o_p x_p + 0.08 \left(\sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \widetilde{CF}_{tl}^{max} \sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t - \theta \right) \tag{8.21}$$

subject to:

$$\max_{\phi_{tl_i} \in \Phi_\Gamma} \phi_{l_i}^t \sum_{p \in \mathcal{P}_i} a_p^t x_p \leq \vartheta_{i_p}^t \Upsilon_i \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{8.22}$$

$$\theta \geq \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \widetilde{CF}(\phi_{tl}) \right) \left(\sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \right) \quad \forall \phi \in \Phi_\Gamma, \tag{8.23}$$

$$\theta \geq 0 \tag{8.24}$$

(8.8), (8.9), (8.10), (8.11), (8.12), (8.13), (8.14), (8.15), (8.16), (8.17), (8.18), (8.19), (8.20)

Applying a cutting plane approach to problem $[RP(\Gamma)]$ implies separating constraints (8.22) and (8.23). Let us explain how we achieve that goal.

First, given a maintenance plan $\bar{x}_p$ (hereafter referred to as $\bar{x}$), we check if all the tasks (except those set to be postponed) can be effectively performed for all the realizations of the uncertainty in Φ_Γ. For each task $i \in \mathcal{I}$, we consider that a task cannot be performed if there exists one particular execution time period of i during which the wind exceeds the safety limit Υ_i. For a task i executed during time period t, the worst-case scenario is obtained by maximizing over the uncertainty set Φ_Γ the wind speed ϕ_{tl_i} and by comparing the resulting value to Υ_i. Indeed, we can just rewrite the constraints (8.22) as follows:

$$\max_{\phi_{tl_i} \in \Phi_\Gamma} \phi_{l_i}^t \sum_{p \in \mathcal{P}_i} a_p^t x_p \leq \vartheta_{i_p}^t \Upsilon_i \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \tag{8.25}$$

Note that constraints (8.25) are equivalent to setting $x_p = 0$ for every plan p for which the following clause holds:

$$\exists t \in \{S_p, ..., C_p\}, \max_{\phi_{tl_p} \in \Phi_\Gamma} \phi_{l_p}^t > \vartheta_{i_p}^t \Upsilon_{i_p} \tag{8.26}$$

A prepossessing technique can therefore achieve the goal of ensuring that the selected maintenance plan can be performed for any wind speed in the uncertainty set. We apply this process and, therefore, eliminate the need of separating constraints (8.22).

Second, given an availability of the wind turbines $\bar{f}_w^t$ (hereafter referred to as $\bar{f}$), we compute the minimum revenue generated by the electricity production of the wind turbines that can be reached according to the uncertainty set. To model the piecewise linear approximation $\widetilde{CF}(.)$ of the function $CF(.)$, we use a multiple choice model as an alternative to the widespread convex combination model. Although these two models have the same LP relaxation and lead to the same bounds (Croxton et al., 2003), the former seems to work better in practice according to Vielma et al. (2010). We introduce binary variable y_{tl}^k taking the value 1 if and only if the wind speed at location l during time period t is within the interval $[\dot{\phi}_k; \dot{\phi}_{k+1}[$ (i.e., lies on the k-th segment). We also introduce for every location l and every time period t the continuous variable ϕ_{tl}^k taking a value equal to the wind speed if and only if the latter lies on the k-th segment. The inner minimization problem $[SRP(\bar{f}, \Gamma)]$ reads:

$$[SRP(\bar{f}, \Gamma)] \quad \min_{\phi \in \Phi_\Gamma} \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \left(\sum_{k=1}^{n} a_k \phi_{tl}^k + b_k y_{tl}^k \right) \left(\sum_{w \in \mathcal{W}_l} P_w hours(t) \bar{f}_w^t \right) \tag{8.27}$$

subject to:

$$\phi_{tl} = \sum_{k=1}^{n-1} \phi_{tl}^k \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}^+, \tag{8.28}$$

$$\dot{\phi}_k y_{tl}^k \leq \phi_{tl}^k \leq \dot{\phi}_{k+1} y_{tl}^k \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}^+, \forall k \in \{1, ..., n-1\}, \tag{8.29}$$

$$\sum_{k=1}^{n-1} y_{tl}^k \leq 1 \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}^+, \tag{8.30}$$

$$\phi_{tl}^k \geq 0 \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}^+, \forall k \in \{1, ..., n\}, \tag{8.31}$$

$$y_{tl}^k \in \{0, 1\} \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}^+, \forall k \in \{1, ..., n\} \tag{8.32}$$

It is worth noting that we only have to consider the revenue function in the interval $[\min_{\phi \in \Phi_\Gamma} \phi_{tl}, \max_{\phi \in \Phi_\Gamma} \phi_{tl}]$ for each location l and each time period t. This can significantly reduce the number of points to consider for the piecewise linear approximation to the CF function. For the non-correlated uncertainty set Φ_Γ^{NC}, such interval is equal to $[\widehat{\phi}_{tl} - \min(1, \Gamma_t)\phi_{tl}^-, \widehat{\phi}_{tl} + \min(1, \Gamma_t)\phi_{tl}^+]$. For the correlated uncertainty set Φ_Γ^C, the expression of the interval is less straightforward, but it can also be computed beforehand.

We denote $\bar{\phi}_{lk}^t$ and $\bar{y}_{lk}^t$ the value of ϕ_{tl}^k and y_{tl}^k in the optimal solution to $[SRP(\bar{f}, \Gamma)]$. Let also $\bar{\theta}$ be the current value of the variable θ in $[RP(\Gamma)]$. The revenue generated by the electricity production of the wind turbines is overestimated if we have:

$$\bar{\theta} < \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k \bar{\phi}_{tl}^k + b_k \bar{y}_{tl}^k \right) \right) \left(\sum_{w \in \mathcal{W}_l} P_w hours(t) \bar{f}_w^t \right)$$

In that case, we introduce the *robustness cut* (8.33).

$$\theta \geq \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k \bar{\phi}_{lk}^t + b_k \bar{y}_{lk}^t \right) \right) \left(\sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \right) \tag{8.33}$$

The sub-problem $[SRP(\bar{f}, \Gamma)]$ always admits a feasible solution and is bounded. We denote $D(\Gamma)$ the finite set of its extreme points for a given Γ. The robust problem can finally be reformulated as the following master problem $[RP_*(\Gamma)]$.

$$[RP_*(\Gamma)] \quad \max - \sum_{p \in \mathcal{P}} o_p x_p + 0.08 \left(\sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \widetilde{CF}_{tl}^{max} \sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t - \theta \right)$$

subject to:

$$\theta \geq \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k \bar{\phi}_{tl}^k + b_k \bar{y}_{tl}^k \right) \right) \left(\sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \right) \quad \forall (\bar{\phi}, \bar{y}) \in D(\Gamma) \tag{8.34}$$

$$x_p = 0 \quad \forall p \in \mathcal{P}, \exists t \in \{S_p, ..., C_p\} \ s.t. \max_{\phi_{tl_p} \in \Phi_\Gamma} \phi_{l_p}^t > \vartheta_{i_p}^t \Upsilon_{i_p} \tag{8.35}$$

(8.8), (8.9), (8.10), (8.11), (8.12), (8.13), (8.14), (8.15), (8.16), (8.17), (8.18), (8.19), (8.20)

This formulation is independent of the type of uncertainty set that we consider. However, if we consider the non-correlated uncertainty set Φ_Γ^{NC}, one can slightly modify the formulation of the robust problem. Indeed, one can observe that the sub-problem $[SRP(f, \Gamma)]$ can be decomposed in $|\mathcal{T}^+|$

independent sub-problems, one per time period. Denoting $[SRP_t(f,\Gamma)]$ the sub-problem restricted to time period $t \in \mathcal{T}^+$, we have $opt([SRP(f,\Gamma)]) = \sum_{t\in\mathcal{T}^+} opt([SRP_t(f,\Gamma)])$. We therefore replace the variable θ in the robust counterpart of the problem by one variable θ_t for every time period t. For every time period t, let $\bar{\theta}_t$ be the current value of variable θ_t in $[RP_*(\Gamma)]$; the revenue generated by the electricity production of the wind turbines during t are overestimated if:

$$\bar{\theta}_t < \sum_{l\in\mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k\bar{\phi}_{tl}^k + b_k\bar{y}_{tl}^k\right)\right)\left(\sum_{w\in\mathcal{W}_l} P_w hours(t)\bar{f}_w^t\right)$$

The robustness cut then reads:

$$\theta_t \geq \sum_{l\in\mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k\bar{\phi}_{tl}^k + b_k\bar{y}_{tl}^k\right)\right)\left(\sum_{w\in\mathcal{W}_l} P_w \cdot hours(t)\cdot f_w^t\right) \tag{8.36}$$

Let us denote $D_t(\Gamma)$ the finite set of extreme points of $[SRP_t(\bar{f},\Gamma)]$. If we consider non-correlated uncertainty, the robust master problem can be reformulated as the following master problem:

$$[RP_*(\Gamma)] \quad \max -\sum_{p\in\mathcal{P}} o_p x_p + 0.08\left(\sum_{t\in\mathcal{T}^+}\sum_{l\in\mathcal{L}}\widetilde{CF}_{tl}^{max}\sum_{w\in\mathcal{W}_l} P_w\cdot hours(t)\cdot f_w^t - \sum_{t\in\mathcal{T}^+}\theta_t\right)$$

subject to:

$$\theta_t \geq \sum_{l\in\mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k\bar{\phi}_{tl}^k + b_k\bar{y}_{tl}^k\right)\right)\left(\sum_{w\in\mathcal{W}_l} P_w \cdot hours(t)\cdot f_w^t\right) \quad \forall t\in\mathcal{T}^+, \forall(\bar{\phi},\bar{y})\in D_t(\Gamma) \tag{8.37}$$

$$x_p = 0 \quad \forall p\in\mathcal{P}, \exists t\in\{S_p,...,C_p\}\ s.t.\ \widehat{\phi}_{l_p}^t + \max(1,\Gamma_t) > \vartheta_{i_p}^t\Upsilon_{i_p} \tag{8.38}$$

(8.8), (8.9), (8.10), (8.11), (8.12), (8.13), (8.14), (8.15), (8.16), (8.17), (8.18), (8.19), (8.20)

8.5.2 General scheme of the solution method

To efficiently solve $[RP_*(\Gamma)]$ we rely on the decomposition approach built to solve the deterministic problem in Chapter 6. Replacing the constraints (8.13), (8.14), (8.15), (8.16), (8.17), (8.18), (8.19), and (8.20) by the combinatorial Benders cuts (6.19), the robust problem can be rewritten as follows:

$$[RP_*(\Gamma)] \quad \max -\sum_{p\in\mathcal{P}} o_p x_p + 0.08\sum_{t\in\mathcal{T}^+}\sum_{l\in\mathcal{L}}\widetilde{CF}_{tl}^{max}\sum_{w\in\mathcal{W}_l} P_w\cdot hours(t)\cdot f_w^t - \theta$$

subject to:

$$\theta \geq \sum_{t\in\mathcal{T}^+}\sum_{l\in\mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k\bar{\phi}_{tl}^k + b_k\bar{y}_{tl}^k\right)\right)\left(\sum_{w\in\mathcal{W}_l} P_w \cdot hours(t)\cdot f_w^t\right) \quad \forall(\bar{\phi},\bar{y})\in D(\Gamma) \tag{8.37}$$

$$x_p = 0 \quad \forall p\in\mathcal{P}, \exists t\in\{S_p,...,C_p\}\ s.t.\ \max_{\phi_{tl_p}\in\Phi_\Gamma}\phi_{l_p}^t > \vartheta_{i_p}^t\Upsilon_{i_p} \tag{8.38}$$

(8.8), (8.9), (8.10), (8.11), (8.17), (8.18),

$$\sum_{p\in\mathcal{P}(\bar{x})} x_p \leq |\mathcal{I}| - 1 \quad \forall\bar{x}\in\bar{\mathcal{F}} \tag{6.19}$$

We initialize a restricted master problem $[RRP_*(\Gamma)]$ with no robustness cuts (8.34) (if considering Φ_Γ^C) or (8.37) (if considering Φ_Γ^{NC}) and no combinatorial Benders cuts (6.19). We then apply the same scheme, with some small adaptations, as the one presented in Figure 6.9 for the B&C approach. At each integer node $(\bar{x},\bar{f})$, we now check two sub-problems. First, we check the existence of a feasible technician-to-task assignment according to the selection of plans $\bar{x}$ in the same way as for the deterministic problem. Second, we solve the sub-problem $[SRP(\bar{f},\Gamma)]$ with a MILP solver. The non-convexity of the $\widetilde{CF}$ function and the complexity of the uncertainty set $\Phi^C(\Gamma)$ makes it difficult to propose an alternative solution method as efficient as the direct resolution of the $MILP$ formulation. If the technician-to-task assignment sub-problem is feasible and the revenue is correctly estimated (i.e., no robustness cuts (8.34) – if considering Φ_Γ^C – or (8.37) – if considering Φ_Γ^{NC} – need to be introduce in the restricted master problem). Otherwise, we add all the generated cuts. To speed-up the process, we can solve in parallel the two sub-problems. Similar to our approach for the deterministic problem, we start the method by iteratively solving a linear relaxation of the scheduling problem until the technician-to-task assignment sub-problem does not produce any cuts. During this first stage, we also solve the sub-problem $[SRP(\bar{f}, \Gamma)]$ at each iteration.

8.5.3 Alternative robust approach

One drawback of the robust approach described up to this point is that the optimization only takes the worst-case scenario into consideration for the computation of the revenue (see the discussion in Section 2.5). Let us consider two maintenance plans x^A and x^B with an negligible difference in terms of minimal revenue (we assume that the minimal revenue is larger for x^A) but a significant difference for the revenue when considering the nominal wind speed (we assume that the average revenue is larger for x^B). One may consider maintenance plan x^B to be preferable to maintenance plan x^A since it seems that it is more "probable" that selecting x^B will generate more revenue. However, the previous approach would select the maintenance plan x^A since the objective is to maximize the revenue in the worst-case.

One possible way to address this issue is to maximize the revenue based on the nominal wind speed while ensuring that in the worst-case scenario, the revenue is larger than a specified value LB (this value should be set by the decision-maker). Fixing this value is not trivial and requires a considerable expertise. Notice that a LB that is too large (i.e., LB is equal to a value not reachable for any feasible maintenance plan) can lead to infeasibility. We refer to this approach as the LB robust approach (as an alternative to the baseline robust approach previously described). From the robust formulation described in Section 8.5.1, the LB robust problem reads:

$$[RP_*^{LB}(\Gamma)] \quad \max -\sum_{p\in\mathcal{P}} o_p x_p + 0.08 \sum_{t\in\mathcal{T}^+} \sum_{l\in\mathcal{L}} CF(\hat{\phi}_{tl}) \sum_{w\in\mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \tag{8.39}$$

subject to: (8.40)

$$-\sum_{p\in\mathcal{P}} o_p x_p + 0.08 \sum_{t\in\mathcal{T}^+} \sum_{l\in\mathcal{L}} \left(\sum_{k=1}^{n} a_k \bar{\phi}_{tl}^k + b_k \bar{y}_{tl}^k \right) \left(\sum_{w\in\mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \right) \geq LB \quad \forall (\bar{\phi}, \bar{y}) \in D(\Gamma) \tag{8.41}$$

$$x_p = 0 \quad \forall p \in \mathcal{P}, \exists t \in \{S_p, ..., C_p\}\ s.t. \max_{\phi_{tl_p}\in\Phi_\Gamma} \phi_{l_p}^t > \vartheta_{i_p}^t \Upsilon_{i_p} \tag{8.38}$$

(8.8), (8.9), (8.10), (8.11), (8.17), (8.18),

$$\sum_{p\in\mathcal{P}(\bar{x})} x_p \leq |\mathcal{I}| - 1 \quad \forall \bar{x} \in \bar{\mathcal{F}} \tag{6.19}$$

The objective in (8.39) is defined as the difference between the nominal revenue generated by the wind turbines and the penalties related to the postponement of some tasks. Constraints (8.7) ensure that the worst-case revenue (according to the uncertainty set Φ_Γ) yielded by the maintenance plan is at least equal to LB. We separate these constraints when solving this alternative robust problem.

8.6 Computational experiments

We implemented our algorithms using *Java 8 (JVM 1.8.0.25)*. We rely on *Gurobi 6.5.1* for solving LP and ILP models. We ran our experiments on a Linux 64 bit-machine, with an Intel(R) Xeon(R) X5675 (3.07Ghz) and 12GB of RAM. We set a 3-hour time limit to solve the different instances (notice that all CPU times are reported in seconds and rounded to the closest integer). In order to assess the quality of our results, we compute the gap with respect to the optimal solution when it is known, or to the best upper bound retrieved by the solver.

We decided to conduct our experiments only on the instances that solved to optimality when considering the deterministic problem. Thus, we only considered the type B instances of testbed G1.

Since we work with randomly generated instances, we have no information on the distribution of the wind speed. We only assume that the nominal value of the wind speed at each location during each time time period is equal to the wind speed in the deterministic problem. However, we need to define the uncertainty set, that is, to assign a value to parameters ϕ_{tl}^- and ϕ_{tl}^+ for every location $l \in \mathcal{L}$ and every time period $t \in \mathcal{T}^+$ and to fix the value of the vector Γ. We choose to fix the values of ϕ_{tl}^- and ϕ_{tl}^+ according to the number of hours between the beginning of time period t and the beginning of the time horizon. Since the accuracy of wind predictions degrade over time, the larger the difference is the larger is the value of the coefficient. In contrast, we do not make any distinction between locations as for a time period the parameters are all set equal (in practice, it is not necessarily true as it is difficult to estimate the wind speed in complex terrain). Note that these parameters can be determined by using historical data. In our case, we consider symmetric values for these parameters. For the first time period, we fix a reference value of $\phi_{1l}^- = -0.25$ and $\phi_{1l}^- = 0.25$ for every location l. Then, every 12 hours, we increase this value by 0.125. For illustration purpose, for every location $l \in \mathcal{L}$ parameters $\phi_{t^*l}^-$ and $\phi_{t^*l}^+$ are equal to $\pm 1.25\ m.s^{-1}$ if t^* is the time period corresponding to the beginning of the time horizon plus 5 days. If we consider the uncorrelated uncertainty set $\Phi(\Gamma)$ and a nominal value of $\phi_{t^*l} = 5\ m.s^{-1}$, this means that the range of possible values for the wind speed is $[3.75\ m.s^{-1}; 6.25\ m.s^{-1}]$. Figure 8.3 illustrates on two different examples the possible range of values for the wind speed at a particular location during the time horizon.

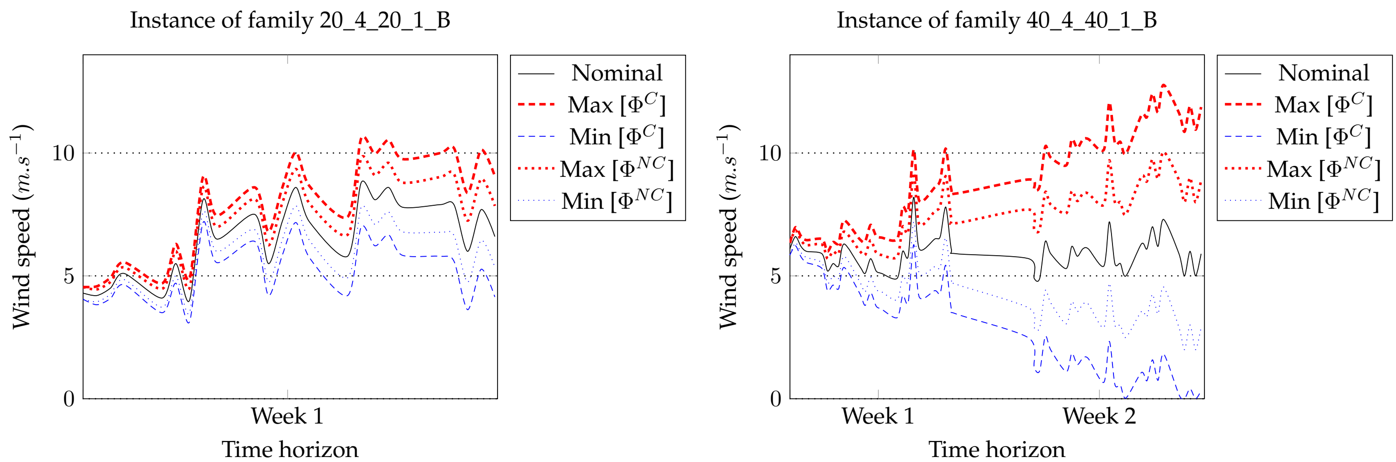


Figure 8.3 – Two examples of the deviations allowed for the wind speed at a particular location according to the uncertainty set

We also need to define parameter Δ^{MAX} for the spatial correlation and parameters $\alpha_{tt'}$ for the time correlation of the wind speed. Since we do not have historical data that we can use to estimate these parameters (e.g., using time series models), we set Δ^{MAX} to an arbitrary value of 15 kms. Moreover, we consider a time correlation of maximum 5 hours. In our case, this means that the wind speed during a time period depends on the deviations observed in, at most, the last two time periods. More specifically, if the number of hours between the beginning of a time period t and the beginning of the time period $t-1$ is larger or equal to 5 hours, we set the parameters $\alpha_{tt-1} = 0.5$ and $\alpha_{tt'} = 0$ if $t' \neq t-1$. If the number of hours is strictly less than 5 hours, we set the parameters $\alpha_{tt-1} = 0.375$, $\alpha_{tt-2} = 0.125$, and $\alpha_{tt'} = 0$ if $t' \neq t-1$ and $t' \neq t-2$.

For the budget vector Γ, we consider different values defined as a proportion of the number of uncertain parameters $\{0, 0.1|\mathcal{L}|, 0.2|\mathcal{L}|, 0.3|\mathcal{L}|, 0.4|\mathcal{L}|, 0.5|\mathcal{L}|, |\mathcal{L}|\}$. For simplicity purpose, once the proportion is chosen, we set all the uncertainty budgets Γ_t equal to the same value. This proportion thus represents the percentage of total deviation from the nominal wind speed allowed during each time period.

8.6.1 Baseline robust approach

We present in this part the computational results for the robust approach. In Tables 8.1 and 8.2, we report the average gap (Gap) over all the instances which are not optimally solved and belonging to the same family or sharing a common characteristic. For the instances with a proven optimum, we report the solution time (Time), the time spent solving the robust sub-problem $[SRP(\bar{f}, \Gamma)]$ (Time.SPR), the number of optimal solutions found within the time limit (#Opt), and the number of robustness cuts generated (#Cuts). For all the instances, we also report the percentage of plans (%rm) that have been removed during the preprocessing stage to ensure the execution of each task for any wind speed in the uncertainty set. First, we observe that we are not able to reach optimality for some of the instances independently of the value of the budget vector Γ and the type of uncertainty set. Nonetheless, this only concerns few instances and the gap is then very small. Moreover, it seems that solving the problem using the uncorrelated uncertainty set $\Phi^{NC}(\Gamma)$ is easier than using the correlated uncertainty set $\Phi^{C}(\Gamma)$. Indeed, we are able to compute more optimal solutions and the solution time is shorter. This partially comes from the fact that adding the disaggregated cuts (8.37) provides more information to the restricted master problem than adding the cuts (8.34). Solving the robust problem is also more time-consuming as the value of the budget grows. We point out here that for a budget proportion set to 100% (all the deviations of the wind speed are allowed), the solution to the sub-problem is simply found when the wind speed ϕ_{tl} at each time period and each location takes the value of $\min(\widehat{\phi}_{tl} - \phi^-_{tl}, \widehat{\phi}_{tl} + \phi^+_{tl})$. Furthermore, to guarantee that each scheduled task can be performed for all the realizations of the wind speed in the uncertainty set, we remove on average 12% of the plans in $\mathcal{P}$. We noticed in our experiments that starting from a budget value of 10% we do not almost always remove any plans by increasing this value. This is explained by the fact that the wind speed at a particular time period t and a particular location l can take the value of $\widehat{\phi}_{tl} + \phi^+_{tl}$ even for small value of the budget proportion. Although this can be thought as a very conservative management, we usually only remove plans that are linked with time periods where the nominal wind speed is already high. These plans are unlikely to be part of any optimal solution. However, when the time horizon spreads over two weeks, the wind speed may vary considerably and it should be pointed out that it is probable that, if the wind speed is around $7m.s^{-1}$, only a few maintenance operations are planned at the end of the time horizon in the optimal solution. Nonetheless, this does make sense in practice.

Table 8.1 – Detailed computational results for the robust approach when considering the correlated uncertainty set $\mathbf{\Phi^C}$.

Characteristic	Γ	Gap	#Opt	Time	Time.SPR	#Cuts	%rm
1 week	**10%**		40/40	23	8	65	1%
	20%	0.00%	39/40	37	9	67	2%
	30%	0.10%	39/40	171	14	100	2%
	40%	0.10%	39/40	275	17	116	2%
	50%	0.15%	38/40	180	17	110	2%
	100%	0.10%	38/40	6	1	30	2%
2 weeks	**10%**	0.03%	34/40	539	216	92	21%
	20%	0.13%	22/40	450	139	120	22%
	30%	0.23%	18/40	784	100	148	22%
	40%	0.24%	20/40	1009	117	164	22%
	50%	0.10%	24/40	1097	81	125	22%
	100%	0.01%	33/40	237	1	33	22%
All	**10%**	0.03%	74/80	260	104	77	11%
	20%	0.13%	61/80	186	56	86	12%
	30%	0.22%	57/80	364	41	115	12%
	40%	0.23%	59/80	524	51	133	12%
	50%	0.14%	62/80	535	42	116	12%
	100%	0.03%	72/80	112	1	31	12%

In Figure 8.4, we show the evolution of the minimal revenue $opt([RP(\Gamma])$ according to the type of uncertainty set and the budget vector. The curve is a decreasing convex function where we observe a quick decrease of $opt([RP(\Gamma])$ with small values of Γ followed by a slow decrease. This comes from the modelization of the uncertainty, since when Γ increases the most influential deviations of the

Table 8.2 – Aggregated computational results for the robust approach.

Uncertainty	Γ	Gap	#Opt	Time	Time.SPR	#Cuts	%rm
Φ^C	**10%**	0.03%	74/80	260	104	77	11%
	20%	0.13%	61/80	186	56	86	12%
	30%	0.22%	57/80	364	41	115	12%
	40%	0.23%	59/80	524	51	133	12%
	50%	0.14%	62/80	535	42	116	12%
	100%	0.03%	72/80	112	1	31	12%
Φ^{NC}	**10%**	-	80/80	72	15	115	1%
	20%	0.00%	78/80	106	14	138	1%
	30%	0.04%	75/80	221	13	152	1%
	40%	0.00%	73/80	290	13	154	1%
	50%	0.04%	72/80	439	12	164	1%
	100%	0.00%	78/80	184	1	94	1%

wind speed on the revenue are chosen first.

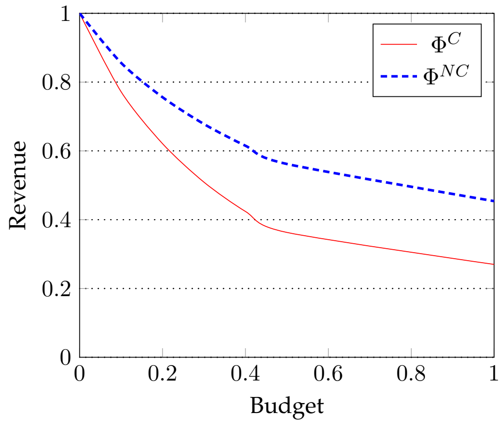


Figure 8.4 – Evolution of the minimal revenue according to the uncertainty budget vector

After evaluating the capacity of the robust approach to find optimal solutions and its performance, we now estimate the quality of the robust solutions. To this end, we perform two different experiments. Notice that we only present here the aggregated results over all the instances since their characteristics do not seem to have a particular influence. We refer to the maintenance plans obtained when considering $\Phi^C(\Gamma)$ and $\Phi^{NC}(\Gamma)$ as the robust correlated and non-correlated maintenance plans. We also refer to the maintenance plan obtained when solving the deterministic problem as the deterministic maintenance plan.

First, we performed simulations using a Monte-Carlo scheme. Notice that by construction of the uncertainty sets, the coordinates of the uncertain vector $\delta = \delta^+ - \delta^-$ – used in the definition of $\Phi^{NC}(\Gamma)$ – are all assumed to be independents. The same holds for the uncertain vector $\ddot{\epsilon} = \ddot{\epsilon}^+ - \ddot{\epsilon}^-$ used for the definition of $\Phi^C(\Gamma)$. We separately considered two different probability distributions for $\ddot{\epsilon}$: uniform $U(-1;1)$ and truncated Gaussian distributions $\mathcal{N}(0,1)$ on the interval $[-1;1]$. From the chosen distribution, we created a sample of 5,000 realizations (i.e., scenarios) for $\ddot{\epsilon}$. We then evaluated the robust solutions on each of these realizations.

We report in Table 8.3 a 95% confidence interval for the proportion of wind speed scenarios (in the uncertainty set) for which the deterministic (line “Uncertainty=D" - "$\Gamma = 0\%$), the robust correlated (lines “Uncertainty= $\Phi^C(\Gamma)$), and the robust non-correlated (lines “Uncertainty= $\Phi^{NC}(\Gamma)$) maintenance plans are feasible (i.e., for every task i the wind speed during the time periods it is executed is strictly less than the maximal authorized value Υ_i). Let us focus on the simulation performed considering the correlated uncertainty set (columns 3 and 5 associated with the symbol $\Phi^C(\Gamma)$). We observe that almost all, if not all, the robust correlated maintenance plans for different budget vectors are feasible for any realization of the wind speed. This is not surprising, since the robust approach aims to take risk-averse decisions. On the contrary, the deterministic maintenance plan as well as the non-correlated maintenance plans are infeasible in 1% of the realizations. Second, if we focus on the simulation performed considering the non-correlated uncertainty set (columns 4 and 6 associated with the symbol $\Phi^{NC}(\Gamma)$), only the deterministic maintenance plans are likely to be infeasible. This can be explained by the fact that we are much more conservative when considering correlated uncertainties. Indeed, it is clear that if we evaluate the correlated maintenance plans considering the non-correlated uncertainty set $\Phi^{NC}(\Gamma)$ they are always feasible. One should also note that small values of Γ seem more appropriate to avoid overconservatism.

Table 8.3 – 95% confidence intervals for the proportion of wind speed scenarios for which the maintenance plan is feasible

			Distribution			
			Uniform		**Gaussian**	
	Uncertainty	**Γ**	$\mathbf{\Phi^{C}}$	$\mathbf{\Phi^{NC}}$	$\mathbf{\Phi^{C}}$	$\mathbf{\Phi^{NC}}$
Solutions	**D**	**0%**	[98.5% ; 98.7%]	[99.2% ; 99.3%]	[98.8% ; 99.0%]	[99.4% ; 99.5%]
	$\mathbf{\Phi^{C}}$	**10%**	[99.9% ; 100%]	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]
		20%	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]
		30%	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]
		40%	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]
		50%	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]
		100%	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]	[100% ; 100%]
	$\mathbf{\Phi^{NC}}$	**10%**	[99.2% ; 99.4%]	[100% ; 100%]	[99.4% ; 99.6%]	[100% ; 100%]
		20%	[99.1% ; 99.4%]	[100% ; 100%]	[99.3% ; 99.5%]	[100% ; 100%]
		30%	[99.0% ; 99.3%]	[100% ; 100%]	[99.3% ; 99.5%]	[100% ; 100%]
		40%	[98.9% ; 99.2%]	[100% ; 100%]	[99.3% ; 99.5%]	[100% ; 100%]
		50%	[99.0% ; 99.2%]	[100% ; 100%]	[99.3% ; 99.6%]	[100% ; 100%]
		100%	[99.0% ; 99.3%]	[100% ; 100%]	[99.4% ; 99.6%]	[100% ; 100%]

D: deterministic case

The simulation also enables to capture the average value of the revenue as well as its variance. Figure 8.5 depicts the box plots of the distribution of the revenue (considered only when the maintenance plan is feasible for the associated realization) for different budget vectors and for the two different type of uncertainty sets. Notice that the ends of the whiskers represent the 1st decile and the 9th decile. This box plot aggregates the results over all the instances (by taking mean values for the average, the quartiles, the deciles, and the extreme values).

As for the previous table, the results are almost identical independently of the chosen distribution. We therefore do not make explicit references to a distribution in the following analysis. The distribution of the revenue does not appear to be particularly impacted by the solutions we consider even if the deterministic maintenance plan is generally better. Obviously, we observe that the maintenance plans computed considering large budget proportions generally leads to the smallest revenue. The percentage difference between the revenue yielded by deterministic maintenance plans and robust maintenance plans computed using a budget proportion of 50% is less than 1% and around 2% for maintenance plans computed considering the wort-case (i.e., 100%). We also notice that non-correlated maintenance plans usually yield larger revenue, which confirms that the associated solutions are less conservative than the ones associated with correlated maintenance plans. However, as reported in Table 8.3, non-correlated maintenance plans are not always feasible. The robust approach then seems to provide an interesting trade-off between feasibility and minimal revenue in the worst-case.

Second, one should be aware of the fact that each maintenance plan is actually optimal regarding a particular uncertainty set; it is protected against any feasibility issue if the wind speed lies in this uncertainty set, and the associated revenue is maximal when one considers the wort-case (according to this uncertainty set) for the wind speed. For different type of uncertainty sets (correlated or non-correlated) and budget vectors Γ, we therefore observe that the robust approach produces different maintenance plans. Since it is not easy for decision-makers to fix the value of the budget according to their trade off preferences between robustness and performance, we would like to assess the quality of the maintenance plans when considering a different uncertainty set (i.e. different type and/or different budget vector). For each maintenance plan obtained for a particular uncertainty set, we therefore computed the minimal revenue and we tested its feasibility in the worst-case if one considers an alternative uncertainty set. Table 8.4 reports the percentage of instances with a feasible

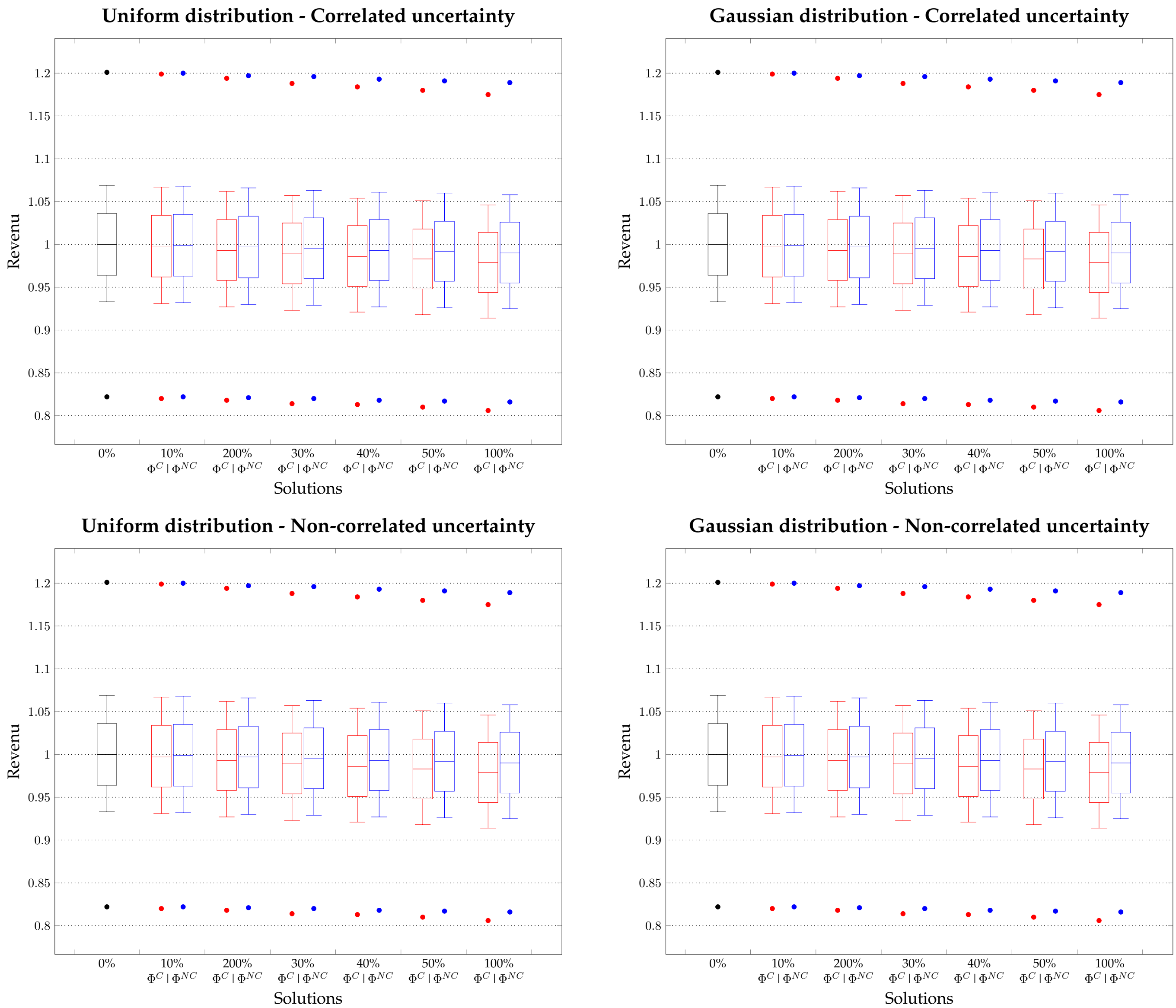


Figure 8.5 – Box plots of the distribution of the revenue in the simulation according to different distributuions and uncertainty sets

maintenance plan. Let us consider a maintenance plan and a particular uncertainty set. For each task, we checked if for any of its execution time periods the wind speed can be above the safety limit according to the selected uncertainty set. Table 8.5 reports the average gap to the optimal minimum revenue computed over all the instances with a feasible maintenance plan. Again, let us consider a maintenance plan and a particular uncertainty set. We computed the minimal revenue according to the selected uncertainty set that is yielded by this maintenance plan. We then computed the gap (percentage change) between this value and the maximum value (for the minimal revenue) that can be reached. This intends to help a decision maker to find a desire trade-off between revenue and guarantee of feasibility.

If we evaluate the maintenance plans assuming correlated uncertainties, we observe that the deterministic and the robust non-correlated maintenance plans are not feasible for half of the instances. On the opposite, the correlated maintenance plans are always feasible. If the wind speed deviates from its nominal value, this underlines the contribution of using the appropriate robust approach. Indeed, considering non-correlated uncertainties when correlation exists yields feasibility issues. In contrast, the opposite does not hold. Regarding the minimal revenue, we observe that the deterministic maintenance plan behaves quite well when the wind speed is uncertain, even if its quality degrades when the uncertainty increases. More generally, we observe that for a maintenance plan optimal for a budget of Γ^* (and a certain type of uncertainty set), the more we restrict (by considering values of Γ lower than Γ^*) or expand (by considering values of Γ larger than Γ^*) the uncertainty the larger is the gap to the optimal minimum revenue. Fixing the budget proportion around 20% seems to provide the best results for the robust correlated maintenance plans. This is in line with the observation of Bertsimas and Sim (2004) that suggested to fix the uncertainty budget in the order of $\sqrt{|\mathcal{L}|}$ to avoid overconservatism.

Table 8.4 – Percentage of instances with a feasible maintenance plan for different uncertainty sets and different budget vectors

			Worst-case evaluation												
	Uncertainty		**D**	$\mathbf{\Phi^C}$						$\mathbf{\Phi^{NC}}$					
		$\mathbf{\Gamma}$	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**100%**	**10%**	**20%**	**30%**	**40%**	**50%**	**100%**
Solutions	**D**	**0%**	100%	53%	51%	51%	51%	51%	51%	95%	95%	95%	95%	95%	95%
	$\mathbf{\Phi^C}$	**10%**	100%	100%	96%	96%	96%	96%	96%	100%	100%	100%	100%	99%	100%
		20%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		30%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		40%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		50%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	$\mathbf{\Phi^{NC}}$	**10%**	100%	51%	49%	49%	49%	49%	49%	100%	100%	100%	100%	100%	100%
		20%	100%	50%	49%	49%	49%	49%	49%	100%	100%	100%	100%	100%	100%
		30%	100%	50%	48%	48%	48%	48%	48%	100%	100%	100%	100%	100%	100%
		40%	100%	49%	48%	48%	48%	48%	48%	100%	100%	100%	100%	100%	100%
		50%	100%	49%	46%	46%	46%	46%	46%	100%	100%	100%	100%	100%	100%
		100%	100%	48%	46%	46%	46%	46%	46%	100%	100%	100%	100%	100%	100%

D: deterministic case

Table 8.5 – Average gap to the optimal niminum revenue for different uncertainty sets and different budget vectors

		Worst-case evaluation												
Uncertainty		D	Φ^C						Φ^{NC}					
	Γ	0%	10%	20%	30%	40%	50%	100%	10%	20%	30%	40%	50%	100%
Solutions D	0%	0.0%	0.1%	0.3%	0.5%	0.7%	0.8%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%
Φ^C	10%	0.3%	0.0%	0.1%	0.4%	0.7%	1.0%	1.2%	0.2%	0.2%	0.3%	0.5%	0.6%	0.8%
	20%	0.6%	0.3%	0.0%	0.2%	0.4%	0.8%	1.0%	0.5%	0.4%	0.4%	0.5%	0.6%	1.0%
	30%	0.9%	0.6%	0.3%	0.0%	0.2%	0.5%	0.9%	0.9%	0.8%	0.7%	0.6%	0.6%	1.0%
	40%	1.1%	1.0%	0.7%	0.3%	0.0%	0.2%	0.7%	1.1%	1.1%	1.0%	0.9%	0.7%	1.0%
	50%	1.3%	1.2%	1.0%	0.7%	0.2%	0.0%	0.5%	1.4%	1.3%	1.3%	1.2%	1.0%	0.9%
	100%	1.8%	1.8%	1.7%	1.6%	1.3%	1.1%	0.0%	1.9%	1.9%	1.9%	1.8%	1.7%	0.9%
Φ^{NC}	10%	0.1%	0.1%	0.2%	0.3%	0.5%	0.6%	1.0%	0.0%	0.1%	0.2%	0.3%	0.4%	0.5%
	20%	0.2%	0.0%	0.1%	0.2%	0.4%	0.5%	0.9%	0.1%	0.0%	0.1%	0.2%	0.3%	0.5%
	30%	0.4%	0.1%	0.1%	0.1%	0.2%	0.3%	0.8%	0.2%	0.1%	0.0%	0.1%	0.2%	0.5%
	40%	0.5%	0.2%	0.2%	0.1%	0.2%	0.2%	0.7%	0.4%	0.3%	0.1%	0.0%	0.1%	0.4%
	50%	0.6%	0.2%	0.2%	0.2%	0.1%	0.2%	0.6%	0.5%	0.4%	0.3%	0.1%	0.0%	0.3%
	100%	0.6%	0.3%	0.4%	0.4%	0.4%	0.4%	0.4%	0.7%	0.7%	0.7%	0.7%	0.6%	0.0%

D: deterministic case

8.6.2 Alternative robust approach

We also tested the LB robust approach described in Section 8.5.3. We present the computational results in Table 8.6. For each instance, we set the value LB according to the solution computed by the CPLNS presented in Chapter 5 after 15 seconds (the average gap for the CPLNS is equal to 0.5% for the deterministic version of the problem if the time limit is set to 15 seconds). More specifically, we set LB to the minimal revenue according to the uncertainty set that can be generated when one executes the maintenance plan retrieved by the CPLNS. We are indeed interested in maintenance plans that yield a minimal revenue larger than LB. Contrary to the approach presented above, all the instances are optimally solved in less than 2 minutes. This may be linked to the fact that adding the robustness cuts (8.41) does not directly influence the objective function. The method may then converge faster. Nonetheless, since the low number of robustness cuts generated may indicate that the value LB is not very binding, only some additional experiments with higher values of LB could assess this latter statement. Otherwise, we globally reach the same conclusions as above on the impact of the chosen uncertainty set.

Table 8.6 – Aggregated computational results on testbed G1 (type B) for the robust approach.

Uncertainty	Γ	#Opt	Time	Time.SPR	#Cuts	%rm
Φ^C	10%	80/80	88	47	11%	1
	20%	80/80	78	39	12%	3
	30%	80/80	69	29	12%	7
	40%	80/80	72	26	12%	5
	50%	80/80	65	22	12%	11
	100%	80/80	37	1	12%	1
Φ^{NC}	10%	80/80	46	6	1%	1
	20%	80/80	47	6	1%	1
	30%	80/80	45	5	1%	3
	40%	80/80	48	5	1%	5
	50%	80/80	48	5	1%	4
	100%	80/80	38	1	1%	1

For each solution, we computed the minimal revenue and we tested its feasibility in the worst-case if one considers an alternative uncertainty set to the one for which the solution is optimal. Table 8.7 reports the percentage of instances with a feasible maintenance plan and Table 8.8 reports the average gap to the optimal minimum revenue computed over all the instances with a feasible maintenance plan. The results we obtained are very similar to the one previously obtained. If we compare the solutions computed using this LB robust approach and the baseline approach, we observe that the former solutions are always feasible and yield a minimum revenue that is, in the worst-case, not far from the one reached for the latter solutions. However, solutions given by the LB robust approach appear to be less conservative as the minimal revenue seems slightly better for small deviations of the wind speed.

Table 8.7 – Percentage of instances with a feasible maintenance plan according to the uncertainty considered

			Worst-case evaluation												
	Uncertainty		**D**	Φ^C						Φ^{NC}					
		Gamma	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**100%**	**10%**	**20%**	**30%**	**40%**	**50%**	**100%**
Solutions	Φ^C	**10%**	100%	100%	96%	96%	96%	96%	96%	100%	100%	100%	100%	100%	100%
		20%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		30%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		40%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		50%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
		100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Φ^C	**10%**	100%	53%	51%	51%	51%	51%	51%	100%	100%	100%	100%	100%	100%
		20%	100%	53%	51%	51%	51%	51%	51%	100%	100%	100%	100%	100%	100%
		30%	100%	53%	51%	51%	51%	51%	51%	100%	100%	100%	100%	100%	100%
		40%	100%	50%	49%	49%	49%	49%	49%	100%	100%	100%	100%	100%	100%
		50%	100%	53%	50%	50%	50%	50%	50%	100%	100%	100%	100%	100%	100%
		100%	100%	53%	51%	51%	51%	51%	51%	100%	100%	100%	100%	100%	100%

D: deterministic case

Table 8.8 – Average gap to the optimal niminum revenue according to the uncertainty considered computed over all the instances with a feasible maintenance plan

			Worst-case evaluation												
	Uncertainty		**D**	Φ^C						Φ^{NC}					
		Γ	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**100%**	**10%**	**20%**	**30%**	**40%**	**50%**	**100%**
Solutions	Φ^C	**10%**	0.1%	0.1%	0.3%	0.6%	0.8%	1.1%	1.4%	0.2%	0.3%	0.4%	0.5%	0.7%	0.8%
		20%	0.2%	0.2%	0.3%	0.6%	0.9%	1.1%	1.4%	0.2%	0.4%	0.5%	0.6%	0.8%	0.9%
		30%	0.2%	0.2%	0.3%	0.6%	0.8%	1.1%	1.4%	0.2%	0.3%	0.5%	0.6%	0.7%	0.9%
		40%	0.1%	0.2%	0.3%	0.6%	0.9%	1.1%	1.4%	0.2%	0.3%	0.5%	0.6%	0.7%	0.9%
		50%	0.2%	0.2%	0.3%	0.6%	0.8%	1.1%	1.3%	0.2%	0.3%	0.5%	0.6%	0.7%	0.9%
		100%	0.2%	0.2%	0.4%	0.6%	0.9%	1.1%	1.4%	0.2%	0.4%	0.5%	0.6%	0.8%	0.9%
	Φ^{NC}	**10%**	0.0%	0.1%	0.3%	0.5%	0.6%	0.8%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%
		20%	0.0%	0.1%	0.3%	0.5%	0.6%	0.8%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%
		30%	0.0%	0.1%	0.3%	0.4%	0.6%	0.7%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%
		40%	0.0%	0.1%	0.3%	0.4%	0.6%	0.7%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%
		50%	0.0%	0.1%	0.3%	0.4%	0.6%	0.7%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%
		100%	0.0%	0.1%	0.3%	0.5%	0.6%	0.7%	1.1%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%

D: deterministic case

8.7 Conclusions

In this chapter, we have proposed a robust approach to take into account the uncertainty related to the wind speed in our maintenance scheduling problem. We have considered row-wise uncertainty. More precisely, we produce maintenance plans that are protected against any feasibility issues. In other words, we are ensured that all the maintenance tasks can be performed according to the safety limit imposed in relation to the wind speed. We also maximize the minimum revenue – according to the uncertainty set – generated when implementing the maintenance plan. We have introduced two different type of uncertainty sets: a set that takes into account the spatial and time-wise correlations of the wind and another set where no correlations are considered.

To tackle the robust problem, we have implemented a method built on top of the decomposition approach presented in Chapter 6. For each selection of plans, along with checking the existence of a technician-to-task assignment, we compute the minimal revenue that can be generated, and we introduce robustness cuts in the restricted master problem if this revenue has been overestimated.

According to the experiments that we conducted, the robust problem is more difficult to solve than its deterministic version. Indeed, we are not able to solve to optimality all the instances. However the gap with respect to upper bounds is almost negligible. We have shown the relevance of computing robust solutions, as we can face feasibility issues if we only take into account the nominal values of the wind speed. Although the revenue is usually slightly smaller for robust solutions, these latter provide an adequate trade-off between revenue and guarantee of feasibility. We also studied the impact of taking into account correlations or not and the impact of the budget vector Γ on the quality of the solution. Not considering correlations when they exist does not produce adequate solutions, whereas the opposite case has only a slight impact on the revenue. This study could allow decision-makers to take the right decisions according to which kind of maintenance plan they are interested in.

We have also introduced an alternative approach that consists in maximizing the nominal revenue while bounding the minimal revenue in the worst-case. We have introduced different robustness cuts to discard selection of plans for which the minimal revenue is not large enough. Although we have conducted limited tests, this robust problem appears to be as easy to solve as the deterministic problem (we find optimal solutions in short computation times). Moreover, the solutions seem less conservative than using the previous approach. They produce a maintenance plan almost always, if not always, feasible while barely penalizing the revenue. Extensive computational experiments should be conducted to confirm this. An interesting perspective could also be to iteratively solve the robust problem while increasing the bound on the minimal revenue.

As a perspective to avoid overconservatism when ensuring the feasibility of the maintenance plans, one could consider an alternative uncertainty budget to remove the plans in which the safety limit can be exceeded (this approach is discussed by Poss (2013) where the author considered a different uncertainty budget for each constraint). Moreover, since a time period usually covers multiple hours, the wind speed is likely to variate inside a time period. Using the average wind speed does not guarantee the right estimation of the wind production since the CF function is not linear. For the sake of precision, it is possible to increase the number of time points during which we evaluate the wind speed. However, one disadvantage of this solution is the corresponding linear increase of the number and of constraints required to model the robust problem. Nonetheless, this could be necessary to ensure more consistent decision-making. Additionally, we could try to adopt more refined strategies to take into consideration the spatial and space-time correlations faced in the wind speed.

Dealing with column-wise uncertainty could be also investigated, but strong challenges arise to define relevant recourse actions. Indeed, one can also think to allow some minor changes in the schedule when the wind speeds are revealed, especially if the wind becomes too strong. As we may have to reschedule some tasks, the corresponding technicians assignments may could be revised. Depending on when the uncertainty is realized, one can naturally think of two different courses of action. If the actual speed of the wind is known a few moments before a technician moves to a location to execute a task, then it would be possible to re-assign the technician to a different task (aiming to speeding up its execution). On the other hand, If the actual speed of the wind is only revealed upon the technician's arrival at the location, it would be hardly possible to assign the technician to another task. In both cases, the canceled tasks may be re-schedule latter. Both approaches are close to reactive scheduling or dynamic scheduling since we work with a short-term horizon. It is worth noting that typical two-stage stochastic approaches are not well-adapted to our problem. Indeed, it is implausible to assume that the actual wind speeds for each location and time period are disclosed at the beginning of the planing horizon. Therefore, if there is too much wind to execute a given task at period $t \in \mathcal{T}^+$, we may want to reschedule it between $t+1$ and $|\mathcal{T}|$. The recourse action must then be taken before the realization of the uncertainty for the remaining periods. A multi-stage approach (e.g., one with one stage per day) would then suit the problem better. Solving multi-stage problems, however, requires complicated techniques (e.g., nested methods and scenario decomposition approaches based on a scenario tree).

8.8 Complement: combining column generation with the decomposition

As a perspective to reduce the solution time for the robust problem, one can think to combine a column generation process (see Section 2.4 for more details) for the plans of $\mathcal{P}$ with the decomposition approach. Indeed, although the complete enumeration of the set $\mathcal{P}$ is possible (as shown in our experiments) and not so large, it has a huge impact on the solution time. It may then be interesting to generate them dynamically. Although we investigate this challenging alternative approach from a theoretical point of view, at the writing of these lines we had not implemented and experimented this approach.

In this new method, at each node of the branch-and-bound tree, we now solve the linear relaxation of the restricted master problem $[RRP_*(\Gamma)]$ with a column generation algorithm. This approach could also be used to solve the deterministic problem, taking into account that on some large-sized instances, our B&C either reports large solution times or struggles proving optimality in the 3-hour time limit.

First, one should notice that the cuts potentially introduced after checking the technician-to-task assignment sub-problem are expressed using the variables x_p. In order to make the pricing problem easier to model (and to solve), we define two new kinds of decision variables. First, we introduce the decision variables z_h related to the number of selected plans that are associated with pattern $h \in \mathcal{H}$. We therefore have $z_h = \sum_{p \in \mathcal{P}_h} x_p \ \forall h \in \mathcal{H}$. To derive combinatorial Benders (CB) cuts, we express each variable z_h using $|\mathcal{P}_h|$ binary variables z_h^k that take value 1 if at least $k \in \{1,..,|\mathcal{P}_h|\}$ plans from set $\mathcal{P}_h$ are selected, and assume value 0 otherwise. Second, we also introduce the variables $\hat{z}_h$ to represent the number of technicians required to be assigned to pattern $h \in \mathcal{H}$. Observing that the Benders feasibility (BF), MCbM, and MWC cuts always include for each pattern $h \in \mathcal{H}$ either none or all the plans of $\mathcal{P}_h$, we can replace the terms $\sum_{p \in \mathcal{P}} q_p x_p$ by $\hat{z}_{h_p}$ in each of these cuts. In order to apply column generation, we reformulate the master problem $[RP_*(\Gamma)]$ as follows:

$$[RP_*(\Gamma)] \quad \max - \sum_{p \in \mathcal{P}} o_p x_p + \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \widetilde{CF}_{tl}^{max} \sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t - \theta \tag{8.42}$$

subject to:

$$\theta \geq \sum_{t \in \mathcal{T}^+} \sum_{l \in \mathcal{L}} \left(\widetilde{CF}_{tl}^{max} - \left(\sum_{k=1}^{n} a_k \bar{\phi}_{tl}^k + b_k \bar{y}_{tl}^k \right) \right) \left(\sum_{w \in \mathcal{W}_l} P_w \cdot hours(t) \cdot f_w^t \right) \quad \forall (\bar{\phi}, \bar{y}) \in D(\Gamma) \tag{8.37}$$

$$\sum_{p \in \mathcal{P}_i} x_p = 1 \quad \forall i \in \mathcal{I}, \tag{8.43}$$

$$\sum_{i \in \mathcal{B}} \sum_{p \in \mathcal{P}_i} a_p^t x_p \leq 1 \quad \forall \mathcal{B} \in ov(\mathcal{I}), \forall t \in \mathcal{T}, \tag{8.44}$$

$$f_w^t + \sum_{p \in \mathcal{P}_i} b_{wp} a_p^t x_p \leq 1 \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}^+, \tag{8.45}$$

$$\sum_{i \in \mathcal{I} | s_i \in \bar{\mathcal{S}}} \sum_{p \in \mathcal{P}_i} a_p^t q_p x_p \leq |R_{\bar{\mathcal{S}}}^t| \quad \forall t \in \mathcal{T}, \forall \bar{\mathcal{S}} \subseteq \mathcal{S}, \tag{8.46}$$

$$\hat{z}_h = \sum_{p \in \mathcal{P}_h} q_p x_p \quad \forall h \in \mathcal{H}, \tag{8.47}$$

$$z_h = \sum_{p \in \mathcal{P}_h} x_p \quad \forall h \in \mathcal{H}, \tag{8.48}$$

$$z_h = \sum_{k=1}^{k=|\mathcal{P}_h|} z_h^k \quad \forall h \in \mathcal{H}, \tag{8.49}$$

$$z_h^{k+1} \leq z_h^k \quad \forall h \in \mathcal{H}, \forall k \in \{1,..,|\mathcal{P}_h| - 1\}, \tag{8.50}$$

$$\sum_{h \in \mathcal{H}} \sum_{k \in \{1,...,|\mathcal{P}_h|\} s.t. \bar{z}_h^k = 1} z_h^k \leq |\mathcal{I}| - 1 \quad \forall \bar{z} \in \bar{\mathcal{F}}, \tag{8.51}$$

$$\theta \geq 0 \tag{8.52}$$

$$x_p \in \{0,1\} \quad \forall p \in \mathcal{P}, \tag{8.53}$$

$$\hat{z}_h \geq 0 \text{ and integer} \quad \forall h \in \mathcal{H}, \tag{8.54}$$

$$z_h \geq 0 \text{ and integer} \quad \forall h \in \mathcal{H}, \tag{8.55}$$

$$z_h^k \in \{0,1\} \quad \forall h \in \mathcal{H}, \forall k \in \{1,..,|\mathcal{P}_h|\}, \tag{8.56}$$

$$f_w^t \in \{0,1\} \quad \forall w \in \mathcal{W}, \forall t \in \mathcal{T}^+, \tag{8.57}$$

Constraints (8.37) are the robustness cuts introduced in this chapter. We assume here that the set $\mathcal{P}$ is built taking into consideration the constraints (8.38). Constraints (8.49) ensure that the sum of binary variables z_h^k is equal to variable z_h for each pattern $h \in \mathcal{H}$. Constraints (8.50) are symmetry constraints to guarantee a unique representation of each variable z_h using the variables z_h^k. Constraints (8.51) are the new combinatorial Benders cuts.

Again, we denote $[RRP_*(\Gamma)]$ as a restricted master problem containing none or a reduced number of robustness, CB, BF, MCbM, and MWC cuts. We introduce $[RRP_*(\Gamma, \bar{\mathcal{P}})]$ as the problem $[RRP_*(\Gamma)]$ restricted to a subset $\bar{\mathcal{P}}$ of plans $\mathcal{P}$. We denote $[RRP_*^{LR}(\Gamma, \bar{\mathcal{P}})]$ its linear relaxation.

Let us now associate in $[RRP_*^{LR}(\Gamma, \bar{\mathcal{P}})]$ the dual variables π_i, ι_β^t, φ_{wi}^t, $\varrho_{\bar{\mathcal{S}}}^t$, $\hat{\varsigma}_h$ and ς_h to, respectively, constraints (8.43), (8.44), (8.45), (8.46), (8.47), and (8.48). The reduced cost $RC(p)$ associated with column (plan) $p \in \mathcal{P}$ is equal to:

$$RC(p) = -c_p - \pi_{i_p} + q_p \hat{\varsigma}_{h_p} + \varsigma_{h_p} - \sum_{t \in \mathcal{T}} a_p^t \left(\sum_{\substack{\beta \in ov(\mathcal{I}) \\ s.t.\ i_p \in \beta}} \iota_\beta^t + q_p \sum_{\substack{\bar{\mathcal{S}} \subseteq \mathcal{S} \\ s.t.\ s_p \in \bar{\mathcal{S}}}} \varrho_{\bar{\mathcal{S}}}^t \right) - \sum_{t \in \mathcal{T}^+} \sum_{w \in \mathcal{W}} a_p^t b_{wp} \varphi_{wi_p}^t \tag{8.58}$$

Since the set $\mathcal{P}$ can be computed prior to the optimization, we can simply loop over the set $\mathcal{P} \setminus \bar{\mathcal{P}}$ to find columns (plans) with a positive reduced cost. The computational complexity of the pricing problem is then up to $O\left(|\mathcal{T}| \sum_{i \in \mathcal{I}} |\mathcal{M}_i|\right)$.

The general scheme of a method combining column generation with the decomposition of the problem differs in several aspects from the one introduced in Section 8.5.2. At first, we can initialize the set $\bar{\mathcal{P}}$ from solutions given by the CPLNS in a short solution time. Henceforth, at each node (non necessarily integer), we solve the current restricted master problem $[RRP_*^{LR}(\Gamma)]$ using a column generation algorithm. At each iteration of this algorithm, as long as we find plans with a positive reduced cost, we successively solve $[RRP_*^{LR}(\Gamma, \bar{\mathcal{P}})]$ and the pricing problem by looping over the set $\mathcal{P} \setminus \bar{\mathcal{P}}$ (where $\bar{\mathcal{P}}$ is the current set of plans considering in the restricted master problem). Let us denote $\bar{x}$ and $\bar{f}$ the optimal value of variables x and f at a node in the search tree. If the current node is integer (i.e. if the solution of $[RRP_*^{LR}(\Gamma)]$ is integer), we check the feasibility of the technician-to-task assignment sub-problem for $\bar{x}$, and we check if the revenue is not overestimated by solving the robust sub-problem $[SRP(\bar{f}, \Gamma)]$. We solve the current restricted master problem (using the column generation algorithm) as long as one of the two sub-problems leads to the introduction of some cuts in the model and the solution we obtain is integer. If the current node is or becomes continuous (i.e. if the solution of the current restricted master problem is continuous), we apply the following branching strategy. The enumerative strategy we propose is a binary branching scheme preserving the structure of the pricing sub-problem and explored using a depth-first search. It is based on forcing/forbidding execution time periods for tasks. More precisely, at each continuous node, we adopt the following procedure. We select the plan p_1 with the highest fractional value in the LP relaxation. Clearly $p_1 = \arg\min_{p \in \bar{P}} \left(\frac{1}{2} - (\bar{x}_p - \lfloor \bar{x}_p \rfloor) \right)$. We denote i^* the task involved in plan p_1 (i.e., $i^* = i_{p_1}$). Then constraint (8.43) for i^* implies that there exists at least another plan in $\bar{P}_{i^*}$ with a fractional value. We take the plan $p_2 \in \bar{P}_{i^*}$ with the second highest fractional value $\bar{x}_{p_2}$. Then, there is obviously at least one time period $t^* \in \mathcal{T}$ for which $a_{p_1}^{t^*} \neq a_{p_2}^{t^*}$. We therefore create two branches, one forcing task i^* to be scheduled during time period t^*, the other one forbidding i^* to be scheduled during time period t^*. Concretely, we remove from $\bar{P}_{i^*}$ all the plans p such that $a_p^{t*} = 0$ for the first branch, whereas we remove from $\bar{P}_{i^*}$ all the plans p such that $a_p^{t*} = 1$ for the second branch. We can simply set the variables corresponding to those plans to 0. Note that the maximal depth for the search tree is $|\mathcal{I}||\mathcal{T}|$ compared to $|\mathcal{P}|$ if we only branch on the x variables. Notice also that this branching strategy ensures that variables f_w^t, $\widetilde{f}_w^d$, $\hat{z}_h$, and z_h are all integer. However one may also need to branch on the z_h^k variables if we have introduced CB cuts. Actually, we can observe that variables z_h, $\hat{z}_h$, and z_h^k need only to be generated for a pattern h once a plan of $\mathcal{P}_h$ is involved in a cut generated when checking the technician-to-task assignment sub-problem. We may also want to solve the robust sub-problem $[SRP(\bar{f}, \Gamma)]$ at non-integer nodes to improve the quality of the upper bound provided by the linear relaxation, which may help to reduce the size of the search tree.

If one solve the deterministic problem with this approach, the main difference comes from the fact that only the technician-to-task sub-problem is checked at each integer node.

Since no implementation and experiments have been made to test the previous ideas, a more insightful study would be required to assess the efficacy of the whole process.

General conclusion and perspectives

This research has explored the optimization of maintenance scheduling in the electricity industry.

First, we have reviewed the OR literature on this topic. We have observed that the maintenance scheduling problem may be defined in a wide variety of ways. We have studied the problem in both regulated and deregulated power systems. We have discussed network considerations, fuel management, and data uncertainty. One of the contribution of this work has been the introduction of a multidimensional classification of the references to make easier to researchers working on this topic identifying the problem they are working on and what have been the main methods used for solving it. From this detailed analysis, we have identified a certain lack of OR studies addressing the new challenges presented by the growing renewable energy sector.

Second, to respond to the need previously identified, we have defined and tackled a new and challenging maintenance scheduling problem faced by the onshore wind power industry. Some of the special features of this problem are the presence of alternative execution modes for each task and an individual management of a multi-skilled workforce through a space-time tracking imposing technician assignments to comply with daily location-based incompatibilities. Addressing the problem on a short-term horizon, the objective is to find a maintenance plan that maximizes the revenue generated by the electricity production of the turbines. This original objective is actually rather far from the classical scheduling concerns as it directly links the revenue to the time periods during which the maintenance operations are performed. We have proved that the problem is strongly NP-hard.

Facing some difficulties to obtain exploitable and realistic data for the problem, we have nonetheless managed to generate a testbed with input from wind forecasting and maintenance scheduling software companies. We have conducted all our experiments on this dataset.

We have first proposed four integer linear programming formulations of the problem. Although the direct resolution of these models is computationally intractable for most of the instances, we have found that the mathematical formulations based on the prior generation of all the feasible schedules of each task seem to produce the best results.

As a heuristic solution method, we have developed a constraint programming-large neighborhood search approach. This method is based on the successive resolution of a CP formulation of the problem in which we have previously fixed the variables associated with some tasks. We have built several destruction operators (to choose the tasks associated with the variables we free up) either specifically conceived for the problem or adapted from the literature. Moreover, to effectively repair solutions we have designed several branching strategies for the resolution of the CP model. We have demonstrated the efficiency of the proposed method as the CPLNS shows in most cases a reduced average gap with respect to the optimal solutions if known, or to the best upper bounds otherwise. This method has been the core of an optimization tool that we developed for the company WPred as part of our collaboration.

In order to develop an appropriate exact method, we have decomposed the problem into a master problem and a satellite sub-problem. The master problem is concerned with fixing the starting time and execution mode of each task while maximizing the revenue generated by the electricity production of the turbines. From a solution to this latter problem, the sub-problem checks the existence of a task-technician assignment coping with the individual unavailability time periods of the technicians and taking into account the daily location-based incompatibilities. On this basis, we have proposed a coordination procedure relying on a branch-and-check technique. More specifically, while solving a restricted master problem with a branch-and-bound algorithm, we check at each integer node the feasibility of the sub-problem. As a priority, we discard infeasible selection of plans using problem-specific cuts. These cuts are based on approximating the sub-problem as a series of maximum cardinality b-matching and maximum-weight clique problems. Since this step only allows to assess a potential infeasibility of the sub-problem but not to prove it, we also use generic cuts: Benders feasibility cuts and combinatorial Benders cuts as the integer linear programming formulation of the sub-problem does not possess the integrality property. In the experiments we conducted, the problem-specific cuts proved to be key speeding up the convergence of the B&C approach. We have found optimal solutions in a short solution time for 80% of the instances and high-quality integer solutions for the remaining instances where the 3-hour time limit is reached.

Lastly, we have proposed a robust approach to tackle the uncertainty of the wind speed. The method aims to take risk-averse decisions. We protect maintenance plans against feasibility issues in cases of strong wind. We also maximize the revenue yielded by a maintenance plan in the worst-case scenario for the wind speed (usually when it is significantly lower than predicted). We have introduced an uncertainty set which takes into consideration the possible spatial and time-wise correlation of the wind speed. To solve the robust counterpart of the problem, we have adapted the branch-and-check approach previously described. More specifically, at each integer node, we introduce robustness cuts if the minimal revenue is overestimated in the restricted master problem. Our computational experiments have demonstrated that the robust problem is more difficult to solve than its deterministic version, but the method computes near-optimal solutions in short execution times. We have shown that using robust maintenance plans avoids feasibility issues while barely penalizing the revenue. To allow decision-makers to take the right decisions according to which kind of maintenance plans they are interested in, we have studied the impact of taking into account correlations or not and the impact of the value of the budget on the quality of the robust solutions. We have also presented an alternative robust approach in which we aim to ensure that the solution performs decently in the worst-case but very good in the nominal case. Preliminary results have suggested that the resulting solutions may be less conservative.

We identify several perspectives to improve this thesis work.

First, to improve the efficiency of the branch-and-check approach (especially on the largest instances), we could further investigate the equivalence of the technician-to-task assignment sub-problem and the L-coloring problem in order to derive new problem-specific cuts. To this purpose, exploring the practical use of Hall's condition could be an idea. We could also implement specific branching strategies to explore the search tree more efficiently while reducing its size. This could improve the quality of the upper bound as it sometimes barely progresses despite the processing of a large number of nodes.

As discussed at the end of Chapter 8, a potential improvement of the decomposition approach (used to exactly solve the deterministic problem and the robust problem) could be to solve the LP relaxation of the restricted master problem at each node using column generation. It would then be necessary to express the cuts (generated when the technician-to-task assignment sub-problem is infeasible) with alternative variables in order to simplify the resolution of the pricing problem. A slow convergence could be an issue, but only computational experiments could assess the performance of this combined approach.

As another interesting perspective of work, we could try to avoid the individual assignment of the technicians to the tasks in the CP model since it is the main reason of the symmetry issues faced by the method. To this end, one could investigate the creation of one or several global constraint(s) based on the use of the approximations to the technician-to-task assignment sub-problem mentioned above (b-matching problem, maximum-weight clique problem, L-coloring problem) as filtering algorithms. However, it presents important implementation challenges to ensure a fast propagation phase.

To ensure more consistent decision-making in the robust approach, we could increase the number of time periods during which we evaluate the wind speed. Indeed, since the CF function is not linear, the shorter (in number of hours) are the time periods the better we estimate the wind output while working with the average wind speed. Nonetheless, slow convergence issues may arise as incrementing the number of time periods corresponds to a linear increase of the number and of constraints required to model the robust problem. Additionally, we could try to adopt more elaborate strategies to take into consideration the spatial and space-time correlations faced in the wind speed, since our strategy is rather simple. If we would manage to have a relevant set of scenarios for the wind speed (maybe from a further collaboration with WPred), we could also investigate the approach of Roy (2010).

Last, we could extend the definition of the problem. For instance, we could investigate the case of technicians working different shifts. Two different approaches could be considered: either one allows tasks to be initiated by some technicians and finished by other technicians (since tasks can last more than the duration of a shift) or one restricts tasks to be performed by technicians working the same shift. From a practical perspective, the first proposition does not seem advisable in this industry. The second proposition is compatible – with slight adjustments – with the solution method presented in Chapters 6 and 8. Actually, every plan (i.e., every feasible schedule for a task) would be associated with a single shift (we would not create a plan that overlaps two different shifts). The technician-to-task assignment sub-problem could then be solved independently for each shift. The method presented in Chapter 5 would require a broader change to take multiple shifts into consideration as the duration of a task would depend on the number of days it overlaps.

Bibliography

Abirami, M., Ganesan, S., Subramanian, S., and Anandhakumar, R. (2014). Source and transmission line maintenance outage scheduling in a power system using teaching learning based optimization algorithm. *Applied Soft Computing*, 21:72–83. 25, 28, 31, 35, 38

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders Decomposition for Production Routing Under Demand Uncertainty. *Operations Research*, 63(4):851–867. 46

Agarwal, R. and Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196. 46

Al-Khamis, T., Vemuri, S., Lemonidis, L., and Yellen, J. (1992). Unit maintenance scheduling with fuel constraints. *IEEE Transactions on Power Systems*, 7(2):933–939. 25, 30, 33, 38

Anghinolfi, D., Gambardella, L., and Montemanni, R. (2012). A matheuristic algorithm for a large-scale energy management problem. In Van Lirkov, I., Margenov, S., and Waśniewsk, J., editors, *Large-Scale Scientific Computing*, volume 7116, pages 173–181. Springer Berlin Heidelberg. 25, 30, 31, 35, 38

Ardestani-Jaafari, A. and Delage, E. (2016). Robust optimization of sums of piecewise linear functions with application to inventory problems. *Operations Research*, 64(2):474–494. 55

Babonneau, F., Klopfenstein, O., Ouorou, A., and Vial, J.-P. (2013). Robust Capacity Assignment Solutions for Telecommunications Networks with Uncertain Demands. *Networks*, 62(4):255–272. 53

Babonneau, F., Vial, J.-P., and Apparigliato, R. (2009). Robust optimization for environmental and energy planning. In *Uncertainty and Environmental Decision Making*, pages 79–126. Springer. 53

Badri, A. and Niazi, A. (2012). Preventive generation maintenance scheduling considering system reliability and energy purchase in restructured power systems. *Journal of Basic and Applied Scientific Research*, 2(12):12773–12786. 26, 27, 28, 30, 31, 32, 38

Baptiste, P., Le Pape, C., and Nuijten, W. (1999). Satisfiability tests and time-bound adjustments for cumulative scheduling problems. *Annals of Operations research*, 92:305–333. 66

Baptiste, P., Le Pape, C., and Nuijten, W. (2001). *Constraint-based scheduling: applying constraint programming to scheduling problems*. Kluwer Academic Publishers. 79

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329. 50

Barot, H. and Bhattacharya, K. (2008). Security coordinated maintenance scheduling in deregulation based on genco contribution to unserved energy. *IEEE Transactions on Power Systems*, 23(4):1871–1882. 26, 27, 28, 32, 38

Baskar, S., Subbaraj, P., Rao, M., and Tamilselvi, S. (2003). Genetic algorithms solution to generator maintenance scheduling with modified genetic operators. *Generation, Transmission and Distribution, IEE Proceedings*, 150(1):56–60. 24, 25, 29, 31, 34, 38

Ben-Tal, A., den Hertog, D., and Vial, J. (2015). Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2):265–299. 54

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press. 51, 55

Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376. 55

Ben-Tal, A. and Nemirovski, A. (2002). Robust optimization–methodology and applications. *Mathematical Programming*, 92(3):453–480. 53

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252. 45

Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and Applications of Robust Optimization. *SIAM review*, 53(3):464–501. 55

Bertsimas, D., Dunning, I., and Lubin, M. (2016). Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217. 53

Bertsimas, D., Pachamanova, D., and Sim, M. (2004). Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516. 54

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical programming*, 98(1-3):49–71. 52, 125

Bertsimas, D. and Sim, M. (2004). The Price of robustness. *Operations Research*, 52(March 2016):35–53. 52, 138

Bertsimas, D. and Thiele, A. (2006). A Robust Optimization Approach to Inventory Theory. *Operations Research*, 54(1):150–168. 53

Billinton, R. and Abdulwhab, A. (2003). Short-term generating unit maintenance scheduling in a deregulated power system using a probabilistic approach. *Generation, Transmission and Distribution, IEE Proceedings*, 150(4). 26, 29, 31, 38

Billionnet, A., Costa, M., and Poirion, P. (2014). 2-stage robust MILP with continuous recourse variables. *Discrete Applied Mathematics*, 170:21–32. 55

Biro, M., Hujter, M., and Tuza, Z. (1992). Precoloring extension. i. interval graphs. *Discrete Mathematics*, 100(1):267–279. 116

Bisanovic, S., Hajro, M., and Dlakic, M. (2011). A profit-based maintenance scheduling of thermal power units in electricity market. *International Journal of Electrical and Electronics Engineering*, 5:156–164. 26, 27, 32, 38

Botton, Q., Fortz, B., Gouveia, L., and Poss, M. (2013). Benders decomposition for the hop-constrainted survivable network design problem. *INFORMS journal on computing*, 25(1):13–26. 46

Bozorgi, A., Pedram, M., and Yousefi, G. (2016). Unit Maintenance Scheduling: A robust model, based on fuzzy cost factors and peak loads. *International Journal of Electrical Power and Energy Systems*, 79:142–149. 26, 27, 29, 30, 36, 38

Brandt, F., Bauer, R., Völker, M., and Cardeneo, A. (2013). A constraint programming-based approach to a large-scale energy management problem with varied constraints. *Journal of Scheduling*, 16(6):629–648. 25, 30, 31, 36, 38

Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577. 74

Budai, G., Dekker, R., and Nicolai, R. (2008). Maintenance and Production: A Review of Planning Models. In *Complex System Maintenance Handbook*, Springer Series in Reliability Engineering, pages 321–324. Springer London. 61

Buljubasic, M. and Gavranovic, H. (2012). Orchestrating constrained programming and local search to solve a large scale energy management problem. In *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 371–378. 25, 30, 31, 36, 38

Burke, E. and Smith, A. (2000). Hybrid evolutionary techniques for the maintenance scheduling problem. *IEEE Transactions on Power Systems*, 15(1). 25, 34, 35, 36, 38

Canto, S. (2008). Application of Benders' decomposition to power plant preventive maintenance scheduling. *European Journal of Operational Research*, 184:759–777. 25, 30, 33, 38

Canto, S. and Rubio-Romero, J. (2013). A model for the preventive maintenance scheduling of power plants including wind farms. *Reliability Engineering & System Safety*, 119:67–75. 24, 25, 30, 32, 38

Carlos, S., Sánchez, A., Martorell, S., and Marton, I. (2013). Onshore wind farms maintenance optimization using a stochastic model. *Mathematical and Computer Modelling*, 57(7-8):1884–1890. 32

Charest, M. and Ferland, J. (1993). Preventive maintenance scheduling of power generating units. *Annals of Operations Research*, 41:185–206. 25, 38

Chattopadhyay, D. (1998). A practical maintenance scheduling program mathematical model and case study. *IEEE Transactions on Power Systems*, 13(4):1475–1480. 25, 28, 30, 31, 38

Chattopadhyay, D. (2004a). A game theoretic model for strategic maintenance and dispatch decisions. *IEEE Transactions on Power Systems*, 19(4):2014–2021. 26, 27, 36, 38

Chattopadhyay, D. (2004b). Life-cycle maintenance management of generating units in a competitive environment. *IEEE Transactions on Power Systems*, 19(2):1181–1189. 26, 27, 29, 30, 31, 38

Chen, L. and Toyoda, J. (1991). Optimal generating unit maintenance scheduling for multi-area system with network constraints. *IEEE Transactions on Power Systems*, 6(3):1168–1174. 24, 25, 28, 32, 38

Codato, G. and Fischetti, M. (2006). Combinatorial Benders' Cuts for Mixed-Integer Linear Programming. *Operations Research*, 54(4):756–766. 48

Conejo, A., Garcia-Bertrand, R., and Diaz-Salazar, M. (2005). Generation maintenance scheduling in restructured power systems. *IEEE Transactions on Power Systems*, 20(2):984–992. 26, 27, 32, 38

Cordeau, J.-F., Laporte, G., Pasin, F., and Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4):393–409. 83

Cordeau, J.-F., Pasin, F., and Solomon, M. (2006). An integrated model for logistics network design. *Annals of operations research*, 144(1):59–82. 46

Croxton, K., Gendron, B., and Magnanti, T. (2003). A Comparison of Mixed-Integer Programming Models for nonconvex Piecewise Linear Cost Minimization Problems. *Management Science*, 49(3):1268–1273. 129

Dahal, K. (2004). A review of maintenance scheduling approaches in deregulated power systems. In *Proceedings of the International Conference on Power Systems (ICPS 2004), Kathmandu, Nepal*, pages 565–570. 22

Dahal, K., Al-Arfaj, K., and Paudyal, K. (2015). Modelling generator maintenance scheduling costs in deregulated power markets. *European Journal of Operational Research*, 240(2):551–561. 27

Dahal, K., Aldridge, C., and McDonald, J. (1999). Generator maintenance scheduling using a genetic algorithm with a fuzzy evaluation function. *Fuzzy Sets and Systems*, 102(1):21–29. 24, 25, 29, 30, 31, 34, 38

Dahal, K. and Chakpitak, N. (2007). Generator maintenance scheduling in power systems using metaheuristic-based hybrid approaches. *Electric Power Systems Research*, 77(7):771–779. 24, 25, 31, 34, 35, 38

Dahal, K. and McDonald, J. (1997). Generator maintenance scheduling of electric power systems using genetic algorithms with integer representations. In *Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA 1997.*, number 446, pages 456–461. 24, 25, 34, 38

De Camargo, R., De Miranda, G., and Ferreira, R. (2011). A hybrid Outer-Approximation/Benders Decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, 39(5):329–337. 46

De Reyck, B., Demeulemeester, E., and Herroelen, W. (1998). Local search methods for the discrete time/resource trade-off problem in project networks. *Naval Research Logistics (NRL)*, 55(6):553–578. 62

Desaulniers, G., Desrosiers, J., and Solomon, M. (2006). *Column generation*, volume 5. Springer Science & Business Media. 50

Digalakis, J. and Margaritis, K. (2002). A multipopulation cultural algorithm for the electrical generator scheduling problem. *Mathematics and Computers in Simulation*, 60(3–5):293–301. 25, 35, 38

Ding, F., Tian, Z., and Jin, T. (2013). Maintenance modeling and optimization for wind turbine systems: A review. In *2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, pages 569–575. 32, 62

Doyle, E. (2004). On the application of stochastic models in nuclear power plant maintenance. *European Journal of Operational Research*, 154(3):673–690. 11

Ekpenyong, U., Zhang, J., and Xia, X. (2012). An improved robust model for generator maintenance scheduling. *Electric Power Systems Research*, 92:29–36. 24, 25, 30, 31, 34, 35, 38

El-Amin, I., Duffuaa, S., and Abbas, M. (2000). A tabu search algorithm for maintenance scheduling of generating units. *Electric Power Systems Research*, 54:91–99. 24, 25, 36, 38

El-Sharkh, M. (2014). Clonal selection algorithm for power generators maintenance scheduling. *International Journal of Electrical Power & Energy Systems*, 57:73–78. 25, 31, 35, 38

El-Sharkh, M., El-Keib, A., and Chen, H. (2003). A fuzzy evolutionary programming-based solution methodology for security-constrained generation maintenance scheduling. *Electric Power Systems Research*, 67(1):67–72. 25, 28, 29, 30, 31, 34, 38

Elyas, S., Akbari Foroud, A., and Chitsaz, H. (2013). A novel method for maintenance scheduling of generating units considering the demand side. *International Journal of Electrical Power & Energy Systems*, 51:201–212. 26, 27, 31, 35, 38

Eshraghnia, R., Modir Shanechi, M., and Rajabi Mashhadi, H. (2006). A new approach for maintenance scheduling of generating units in power market. In *International Conference on Probabilistic Methods Applied to Power Systems*, pages 9–15. 26, 27, 31, 34, 38

European Commission (2011). World and european energy and environment transition outlook. Retrieved from `https://ec.europa.eu/research/social-sciences/pdf/policy_reviews/publication-weto-t_en.pdf`. 9, 21

Fattahi, M., Mahootchi, M., Mosadegh, H., and Fallahi, F. (2014). A new approach for maintenance scheduling of generating units in electrical power systems based on their operational hours. *Computers & Operations Research*. 25, 31, 35, 36, 38

Feng, C., Wang, X., and Li, F. (2009). Optimal maintenance scheduling of power producers considering unexpected unit failure. *IET Generation, Transmission & Distribution*, 3(5):460–471. 26, 27, 29, 31, 33, 38

Fetanat, A. and Shafipour, G. (2011). Generation maintenance scheduling in power systems using ant colony optimization for continuous domains based 0–1 integer programming. *Expert Systems with Applications*, 38(8):9729–9735. 24, 25, 38

Fischetti, M. and Monaci, M. (2012). Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3):239–273. 53

Focacci, F., Lodi, A., and Milano, M. (1999). Cost-based domain filtering. In *International Conference on Principles and Practice of Constraint Programming*, pages 189–203. Springer. 43

Foong, W., Simpson, A., Maier, H., and Stolp, S. (2007). Ant colony optimization for power plant maintenance scheduling optimization—a five-station hydropower system. *Annals of Operations Research*, 159(1):433–450. 24, 25, 36, 38

Fourcade, F., Johnson, E., Bara, M., and Cortey-Dumont, P. (1997). Optimizing nuclear power plant refueling with mixed-integer programming. *European Journal of Operational Research*, 97(2):269–280. 25, 30, 31, 32, 39

Frost, D. and Dechter, R. (1998). Optimizing with constraints: A case study in scheduling maintenance of electric power units. In *Lecture Notes in Computer Science Volume 1520*, pages 469–488. 25, 36, 39

Fu, Y., Shahidehpour, M., and Li, Z. (2007). Security-constrained optimal coordination of generation and transmission maintenance outage scheduling. *IEEE Transactions on Power Systems*, 22(3):1302–1313. 25, 26, 27, 28, 31, 33, 39

Gabrel, V., Lacroix, M., Murat, C., and Remli, N. (2014a). Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164(PART 1):100–111. 55

Gabrel, V., Murat, C., and Thiele, A. (2014b). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483. 55

Gabrel, V., Murat, C., and Wu, L. (2013). New models for the robust shortest path problem: Complexity, resolution and generalization. *Annals of Operations Research*, 207(1):97–120. 54

Gardi, F. and Nouioua, K. (2011). Local search for mixed-integer nonlinear optimization: A methodology and an application. In *EvoCop: 11th European conference on evolutionary computation in combinatorial optimisation, 27-29 April 2011 Torino, Italy*, pages 167–178. 25, 30, 31, 33, 39

Geetha, T. and Shanti Swarup, K. (2009). Coordinated preventive maintenance scheduling of GENCO and TRANSCO in restructured power systems. *International Journal of Electrical Power & Energy Systems*, 31(10):626–638. 26, 27, 28, 29, 31, 33, 39

Gendron, B., Scutellà, M., Garroppo, R., Nencioni, G., and Tavanti, L. (2014). A Branch-and-Benders-Cut Method for Nonlinear Power Design in Green Wireless Local Area Networks. Technical Report CIRRELT-2014-42, CIRRELT. 46

Godskesen, S., Jensen, T., Kjeldsen, N., and Larsen, R. (2013). Solving a real-life, large-scale energy management problem. *Journal of Scheduling*, 16(6):567–583. 25, 30, 31, 35, 36, 39

Gorge, A., Lisser, A., and Zorgati, R. (2012). Stochastic nuclear outages semidefinite relaxations. *Computational Management Science*, 9(3):363–379. 25, 30, 31, 33, 39

Gualandi, S. and Malucelli, F. (2009). Constraint programming-based column generation. *4OR*, 7(2):113–137. 43

Han, S.-M., Chung, K.-H., and Kim, B. (2011). ISO coordination of generator maintenance scheduling in competitive electricity markets using simulated annealing. *Journal of Electrical Engineering & Technology*, 6(4):431–438. 26, 27, 29, 35, 39

Harvey, W. and Ginsberg, M. (1995). Limited discrepancy search. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 607–613. 43

Hooker, J. (2000). *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley. 43

Hooker, J. and Osorio, M. (1999). Mixed logical-linear programming. *Discrete Applied Mathematics*, 96:395–442. 43

Hooker, J. N. and Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60. 48

Huang, S.-J. (1997). Generator maintenance scheduling: A fuzzy system approach with genetic enhancement. *Electric Power Systems Research*, 41(3):233–239. 26, 29, 30, 32, 34, 39

Irawan, C., Ouelhadj, D., Jones, D., Stålhane, M., and Sperstad, I. (2016). Optimisation of maintenance routing and scheduling for offshore wind farms. *European Journal of Operational Research*. 32, 62

Jain, V. and Grossmann, I. (2001). Algorithms for hybrid milp/cp models for a class of optimization problems. *INFORMS Journal on computing*, 13(4):258–276. 43

Jensen, T. and Toft, B. (2011). *Graph coloring problems*, volume 39. John Wiley & Sons. 116

Jiang, R., Wang, J., and Guan, Y. (2012). Robust unit commitment with wind power and pumped storage hydro. *IEEE Transactions on Power Systems*, 27(2):800–810. 55

Jost, V. and Savourey, D. (2013). A 0–1 integer linear programming approach to schedule outages of nuclear power plants. *Journal of Scheduling*, 16(6):551–566. 25, 30, 31, 33, 39

Kelley, Jr, J. (1960). The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712. 46

Khalid, A. and Ioannis, K. (2012). A survey of generator maintenance scheduling techniques. *Global Journal of Researches in Engineering*, 12(1). 22

Khemmoudj, M., Porcheron, M., and Bennaceur, H. (2006). When constraint programming and local search solve the scheduling problem of Electricité de France nuclear power plant outages. In Benhamou, F., editor, *Principles and Practice of Constraint Programming - CP 2006*, volume 4204 of *Lecture Notes in Computer Science*, pages 271–283. Springer Berlin Heidelberg. 25, 30, 31, 36, 39

Kim, J.-H., Park, J.-B., Park, J.-K., and Chun, Y.-H. (2005). Generating unit maintenance scheduling under competitive market environments. *International Journal of Electrical Power & Energy Systems*, 27(3):189–194. 26, 27, 36, 39

Kouvelis, P. and Yu, G. (1997). *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media. 54

Kovács, A., Erdős, G., Viharos, Z., and Monostori, L. (2011). A system for the detailed scheduling of wind farm maintenance. *CIRP Annals - Manufacturing Technology*, 60(1):497–501. 32, 62

Kralj, B. and Petrović, R. (1988). Optimal preventive maintenance scheduling of thermal generating units in power systems - A survey of problem formulations and solution methods. *European Journal of Operational Research*, 35:1–15. 22

Kralj, B. and Petrovic, R. (1995). A multiobjective optimization approach to thermal generating units maintenance scheduling. *European Journal of Operational Research*, 84(2):481–493. 25, 26, 33, 39

Langdon, W. and Treleaven, P. (1997). Scheduling maintenance of electrical power transmission networks using genetic programming. In Warwick, K., Ekwue, A., and Aggarwal, R., editors, *Artificial Intelligence Techniques in Power Systems, chapter 10*, pages 220–237. IEE. 28, 34, 39

Laporte, G. and Louveaux, F. V. (1993). The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133 – 142. 48

Latify, M., Seifi, H., and Mashhadi, H. (2013). An integrated model for generation maintenance coordination in a restructured power system involving gas network constraints and uncertainties. *International Journal of Electrical Power and Energy Systems*, 46(1):425–440. 26, 27, 30, 32, 39

Lawan, S., Abidin, W., Chai, W., Baharun, A., and Masri, T. (2014). Different Models of Wind Speed Prediction; A Comprehensive Review. *International Journal of Scientific & Engineering Research*, 5(1):1760–1768. 124

Lecoutre, C., Sais, L., and Vidal, V. (2007). Nogood Recording from Restarts. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 131–136. 43

Leou, R.-C. (2006). A new method for unit maintenance scheduling considering reliability and operation expense. *International Journal of Electrical Power & Energy Systems*, 28(7):471–481. 25, 28, 34, 35, 39

Lorca, A. and Sun, X. (2014). Adaptive Robust Optimization With Dynamic Uncertainty Sets for Multi-Period Economic Dispatch Under Significant Wind. *IEEE Transactions on Power Systems*, pages 1–16. 55, 126

Lusby, R., Muller, L., and Petersen, B. (2013). A solution approach based on Benders decomposition for the preventive maintenance scheduling problem of a stochastic large-scale energy system. *Journal of Scheduling*, 16(6):605–628. 25, 30, 31, 39

Lv, C., Wang, J., and Sun, P. (2012). Short-term transmission maintenance scheduling based on the Benders decomposition. In *Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific*, pages 1–5. 28, 29, 33, 39

Magnanti, T. and Wong, R. (1981). Accelerating Benders Decomposition : Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, 29(3):464–484. 48

Malapert, A., Guéret, C., and Rousseau, L.-M. (2012). A constraint programming approach for a batch processing problem with non-identical job sizes. *European Journal of Operational Research*, 221(3):533 – 545. 79

Marwali, M. and Shahidehpour, S. (1998). A deterministic approach to generation and transmission maintenance scheduling with network constraints. *Electric Power Systems Research*, 47(2):101–113. 25, 28, 31, 33, 39

Marwali, M. and Shahidehpour, S. (1999a). A probabilistic approach to generation maintenance scheduler with network constraints. *International Journal of Electrical Power & Energy Systems*, 21(8):533–545. 25, 26, 27, 28, 29, 31, 33, 39

Marwali, M. and Shahidehpour, S. (1999b). Long-term transmission and generation maintenance scheduling with network, fuel and emission constraints. *IEEE Transactions on Power Systems*, 14(3):1160–1165. 26, 27, 28, 29, 30, 31, 33, 39

Marwali, M. and Shahidehpour, S. (2000a). Coordination between long-term and short-term generation scheduling with network constraints. *IEEE Transactions on Power Systems*, 15(3):1161–1167. 25, 26, 27, 28, 29, 30, 31, 33, 39

Marwali, M. and Shahidehpour, S. (2000b). Short-term transmission line maintenance scheduling in a deregulated system. *IEEE Transactions on Power Systems*, 15(3):1117–1124. 26, 27, 28, 31, 39

McDaniel, D. and Devine, M. (1977). A Modified Benders' Partitioning Algorithm for Mixed Integer Programming. *Management Science*, 24(3):312–319. 48

Min, C., Kim, M., Park, J., and Yoon, Y. (2013). Game-theory-based generation maintenance scheduling in electricity markets. *Energy*, 55:310–318. 26, 27, 36, 39

Minoux, M. (2011). On 2-stage robust LP with RHS uncertainty: Complexity results and applications. *Journal of Global Optimization*, 49(3):521–537. 55

Mohanta, D., Sadhu, P., and Chakrabarti, R. (2004). Fuzzy reliability evaluation of captive power plant maintenance scheduling incorporating uncertain forced outage rate and load representation. *Electric Power Systems Research*, 72(1):73–84. 24, 25, 29, 30, 34, 35, 39

Mohanta, D., Sadhu, P., and Chakrabarti, R. (2007). Deterministic and stochastic approach for safety and reliability optimization of captive power plant maintenance scheduling using GA/SA-based hybrid techniques: A comparison of results. *Reliability Engineering & System Safety*, 92(2):187–199. 24, 25, 29, 34, 35, 39

Mollahassani-pour, M., Abdollahi, A., and Rashidinejad, M. (2014). Application of a novel cost reduction index to preventive maintenance scheduling. *International Journal of Electrical Power & Energy Systems*, 56:235–240. 25, 31, 32, 39

Morales, J., Minguez, R., and Conejo, A. (2010). A methodology to generate statistically dependent wind speed scenarios. *Applied Energy*, 87(3):843–855. 124

Muñoz-Moro, L. and Ramos, A. (1999). Goal programming approach to maintenance scheduling of generating units in large scale power systems. *IEEE Transactions on Power Systems*, 14(3):1021–1028. 26, 30, 31, 33, 39

Mytakidis, T. and Vlachos, A. (2008). Maintenance scheduling by using the bi-criterion algorithm of preferential anti-pheromones. *Leonardo Journal of Sciences*, (12):143–164. 25, 36, 39

Naoum-Sawaya, J. and Elhedhli, S. (2010). An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210(1):33–55. 46

Östergård, P. (2001). A new algorithm for the maximum-weight clique problem. *Nordic Journal of Computing*, 8(4):424–436. 103

Östergård, P. (2002). A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1):197–207. 103

Papadakos, N. (2008). Practical enhancements to the Magnanti–Wong method. *Operations Research Letters*, 36(4):444–449. 48

Pinson, P. and Girard, R. (2012). Evaluating the quality of scenarios of short-term wind power generation. *Applied Energy*, 96:12–20. 124

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435. 44, 45

Porcheron, M., Gorge, A., Juan, O., Simovic, T., and Dereu, G. (2010). Challenge ROADEF/EURO 2010: A large-scale energy management problem with varied constraints. 31

Poss, M. (2013). Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11:75–92. 141

Prud'homme, C., Fages, J.-G., and Lorca, X. (2014). *Choco3 Documentation*. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S. 86

Qin, S., Liu, F., Wang, J., and Song, Y. (2015). Interval forecasts of a novelty hybrid model for wind speeds. *Energy Reports*, 1:8–16. 124

Reihani, E., Sarikhani, A., and Davodi, M. (2012). Reliability based generator maintenance scheduling using hybrid evolutionary approach. *International Journal of Electrical Power & Energy Systems*, 42(1):434–439. 24, 25, 29, 34, 39

Reliability Test System Task Force of the Application of Probability Methods Subcommittee (1979). IEEE reliability test system. *IEEE Transactions on Power Apparatus and Systems*, PAS-98(6):2047–2054. 31

Reliability Test System Task Force of the Application of Probability Methods Subcommittee (1999). The IEEE reliability test system - 1996. *IEEE Transactions on Power Systems*, 14(3):1010–1020. 31

Rodriguez, J. (2007). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B: Methodological*, 41(2):231–245. 79

Ropke, S. and Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472. 44

Rossi, F., Van Beek, P., and Walsh, T. (2006). *Handbook of constraint programming*. Elsevier. 43

Roy, B. (2010). Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200(3):629–638. 51, 54, 55, 147

Rozenknop, A., Wolfler Calvo, R., Alfandari, L., Chemla, D., and Létocart, L. (2013). Solving the electricity production planning problem by a column generation based heuristic. *Journal of Scheduling*, 16(6):585–604. 25, 30, 31, 33, 39

Sadykov, R. (2008). A branch-and-check algorithm for minimizing the weighted number of late jobs on a single machine with release dates. *European Journal of Operational Research*, 189(3):1284–1304. 47

Saharidis, G. and Ierapetritou, M. (2010). Improving benders decomposition using maximum feasible subsystem (MFS) cut generation strategy. *Computers & Chemical Engineering*, 34(8):1237–1245. 48

Saraiva, J., Pereira, M., Mendes, V., and Sousa, J. (2011). A simulated annealing based approach to solve the generator maintenance scheduling problem. *Electric Power Systems Research*, 81(7):1283–1291. 25, 35, 39

Satoh, T. and Nara, K. (1991). Maintenance scheduling by using simulated annealing method. *IEEE Transactions on Power Systems*, 6(2):850–857. 25, 40

Schlünz, E. and van Vuuren, J. (2013). An investigation into the effectiveness of simulated annealing as a solution approach for the generator maintenance scheduling problem. *International Journal of Electrical Power & Energy Systems*, 53:166–174. 25, 31, 40

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons. 42

Shabanzadeh, M. and Fattahi, M. (2015). Generation maintenace scheduling via robust optimization. In *Electrical Engineering (ICEE), 2015 23rd Iranian Conference on*, pages 1504–1509. 26, 27, 30, 33, 40

Shahidehopour, M. and Fu, Y. (2005). Benders decomposition: applying benders decomposition to power systems. *Power and Energy Magazine, IEEE*, 3(2):20–21. 46

Shahidehpour, M., Yamin, H., and Li, Z. (2002). *Market Operations in Electric Power Systems*. Wiley-IEEE Press. 22

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. *Principles and Practice of Constraint Programming-CP 1998*, pages 417–431. 44

Silva, E. (2000). Generation maintenance scheduling considering transmission constraints. *IEEE Transactions on Power Systems*, 15(2):838–843. 25, 28, 33, 40

Silva, E. and Morozowski, M. (1995). Transmission constrained maintenance scheduling of generating units: A stochastic programming approach. *IEEE Transactions on Power Systems*, 10(2):695–701. 25, 28, 29, 33, 40

Souyris, S., Cortés, C., Ordóñez, F., and Weintraub, A. (2012). A robust optimization approach to dispatching technicians under stochastic service times. *Optimization Letters*. 55

Soyster, A. (1973). Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157. 51, 52, 55, 126

Suresh, K. and Kumarappan, N. (2013). Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem. *Swarm and Evolutionary Computation*, 9:69–89. 25, 29, 31, 34, 35, 40

The Global Wind Energy Council (2016). Global wind report annual market update 2015. Retrieved from `http://www.gwec.net/wp-content/uploads/vip/GWEC-Global-Wind-2015-Report_April-2016_22_04.pdf`. 11, 12, 61

Thorsteinsson, E. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In Walsh, T., editor, *Principles and Practice of Constraint Programming — CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin Heidelberg. 43, 47

Vielma, J., Ahmed, S., and Nemhauser, G. (2010). Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions. *Operations Research*, 58(2):303–315. 129

Volkanovski, A. and Mavko, B. (2008). Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system. *Reliability Engineering & System Safety*, 93(6):757–767. 25, 29, 34, 40

Wang, X., Guo, P., and Huang, X. (2011). A review of wind power forecasting models. *Energy Procedia*, 12:770–778. 124

Wang, Y. and Handschin, E. (2000). A new genetic algorithm for preventive unit maintenance scheduling of power systems. *Journal of Electrical Power & Energy Systems*, 22(5):343–348. 25, 34, 40

Wright, S. (1997). *Primal-dual interior-point methods*. Siam. 42

Wu, L., Shahidehpour, M., and Li, T. (2008). GENCO's risk-based maintenance outage scheduling. *IEEE Transactions on Power Systems*, 23(1):127–136. 26, 27, 28, 30, 33, 40

Yamayee, Z. (1982). Maintenance scheduling: Description, literature survey, and interface with overall operations scheduling. *IEEE Transactions on Power Apparatus and Systems*, 75(8):2770–2779. 22, 32

Yamayee, Z., Sidenblad, K., and Yoshimura, M. (1983). A computationally efficient optimal maintenance scheduling method. *IEEE Transactions on Power Apparatus and Systems*, PAS-102(2):330–338. 31

Yare, Y. and Venayagamoorthy, G. (2010). Optimal maintenance scheduling of generators using multiple swarms-MDPSO framework. *Engineering Applications of Artificial Intelligence*, 23(6):895–910. 25, 31, 34, 35, 40

Yellen, J. and Al-Khamis, T. (1992). A decomposition approach to unit maintenance scheduling. *IEEE Transactions on Power Systems*, 7(2):726–733. 25, 29, 33, 40

Yssaad, B., Khiat, M., and Chaker, A. (2014). Reliability centered maintenance optimization for power distribution systems. *International Journal of Electrical Power & Energy Systems*, 55:108–115. 11

Zeng, B. and Zhao, L. (2013). Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461. 55

Zhan, J., Guo, C., Wu, Q., and Zhang, L. (2014). Generation Maintenance Scheduling Based on Multiple Objectives and Their Relationship Analysis. *Journal of Zhejiang University*, pages 1–13. 26, 27, 31, 35, 40

Zhang, J., Chowdhury, S., Messac, A., and Castillo, L. (2013). A Multivariate and Multimodal Wind Distribution model. *Renewable Energy*, 51:436–447. 123

Zhang, Y., Wang, J., and Wang, X. (2014). Review on probabilistic forecasting of wind power generation. *Renewable and Sustainable Energy Reviews*, 32(April):255–270. 124

Appendix A

Notations

To help the reader, we summarize below the main notations used in Part II.

For two integers a_1 and a_2, symbol $\{a_1, ..., a_2\}$ refers to $[a_1; a_2] \cap \mathbb{Z}$ where $\mathbb{Z}$ is the set of integers.

Time horizon

- $\mathcal{T}$: time horizon (totally ordered set)
- $\mathcal{D}$: set of days
- $\mathcal{T}_d$: set of time periods that belongs to day $d \in \mathcal{D}$
- t_d^{rest}: last time period $t \in \mathcal{T}$ before the rest time period following day $d \in \mathcal{D}$
- d_t: day associated with time period $t \in \mathcal{T}$

Locations

- $\mathcal{L}$: set of locations (wind farms and technician home depots)
- $\delta_{ll'} = \begin{cases} 1 & \text{if a technician is allowed to work at both locations } l \text{ and } l' \text{ during the same day,} \\ 0 & \text{otherwise.} \end{cases}$

Technicians

- $\mathcal{S}$: set of skills
- $\mathcal{R}$: set of technicians
- $\zeta_{rs} = \begin{cases} 1 & \text{if technician } r \in \mathcal{R} \text{ masters skill } s \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases}$
- $\rho_r^t = \begin{cases} 1 & \text{if technician } r \in \mathcal{R} \text{ is available during time period } t \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$
- l_r^t: location of technician $r \in \mathcal{R}$ when he or she is not available during time period $t \in \{t' \in \mathcal{T} \mid \rho_r^{t'} = 1\}$

Tasks

- $\mathcal{I}$: set of tasks to perform at the different locations
- $ov(\mathcal{I})$: family of sets of tasks that cannot overlap
- l_i: location where task $i \in \mathcal{I}$ has to be performed
- $\mathcal{I}_l$: set of tasks to perform at location $l \in \mathcal{L}$ (i.e. $\mathcal{I}_l = \{i \in \mathcal{I} \mid l_i = l\}$)
- $\mathcal{M}_i$: set of execution modes for task $i \in \mathcal{I}$
- m_i^0: execution mode related to the postponement of task $i \in \mathcal{I}$ ($m_i^0 \in \mathcal{M}_i$)
- q_{im}: number of technicians required during each time period to perform task $i \in \mathcal{I}$ in mode $m \in \mathcal{M}_i$ ($q_{im_i^0} = 0$)
- d_{im}: duration of task $i \in \mathcal{I}$ if performed in mode $m \in \mathcal{M}_i$ ($d_{im_i^0} = 0$)

— $\vartheta_i^t = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ can be executed during time period } t \in \mathcal{T} \text{ (time windows, preferences)},\\ 0 & \text{otherwise.} \end{cases}$

— $\ddot{\vartheta}_i^t = \begin{cases} 1 & \text{if task } i \in \mathcal{I} \text{ can be executed during time period } t \in \mathcal{T} \text{ (safety concerns)},\\ 0 & \text{otherwise.} \end{cases}$

— s_i: skill required to perform task $i \in \mathcal{I}$

— o_i: penalty if task $i \in \mathcal{I}$ is postponed

Turbines

— $\mathcal{W}$: set of turbines

— l_w: location of turbine w

— $\mathcal{W}_l$: set of turbines at location $l \in \mathcal{L}$ (i.e. $\mathcal{W}_l = \{w \in \mathcal{W} \mid l_w = l\}$)

— $b_{wi} = \begin{cases} 1 & \text{if the execution of task } i \in \mathcal{I} \text{ shuts down turbine } w \in \mathcal{W} \text{ when technicians}\\ & \text{are effectively working on } i,\\ 0 & \text{otherwise.} \end{cases}$

— $\widetilde{b}_{wi} = \begin{cases} 1 & \text{if the execution of task } i \in \mathcal{I} \text{ shuts down turbine } w \in \mathcal{W} \text{ during the rest time}\\ & \text{periods it overlaps,}\\ 0 & \text{otherwise.} \end{cases}$

— g_w^t: profit if turbine $w \in \mathcal{W}$ can produce electricity during time period $t \in \mathcal{T}$

— $\widetilde{g}_w^d$: profit if turbine $w \in \mathcal{W}$ can produce electricity during the rest time period following day $d \in D$

Plans

— $\mathcal{P}$: set of plans

— $\mathcal{P}_i$: set of plans involving task $i \in \mathcal{I}$

— i_p: task involved in plan $p \in \mathcal{P}$

— $a_p^t = \begin{cases} 1 & \text{if task } i_p \text{ is executed during time period } t \in \mathcal{T}\\ 0 & \text{otherwise.} \end{cases}$

— S_p: starting time period of plan $p \in \mathcal{P}$ ($S_p = \min_{t\in\mathcal{T}} a_p^t t$)

— C_p: ending time period of plan $p \in \mathcal{P}$ ($C_p = \max_{t\in\mathcal{T}} a_p^t t$)

— $\mathcal{D}_p$: set of days overlapped by plan p

— q_p: number of required technicians if plan $p \in \mathcal{P}$ is selected

— $s_p = s_{i_p}$

— $l_p = l_{i_p}$

— $\mathcal{R}_p = \mathcal{R}_{i_p}$

— $b_{wp} = b_{wi_p}$

— $\widetilde{b}_{wp} = \widetilde{b}_{wi_p}$

— $o_p = o_{i_p}$

Patterns

— $\mathcal{H}$: set of patterns

— $\mathcal{P}_h$: set of plans associated with pattern $h \in \mathcal{H}$

— h_p: pattern associated with plan $p \in \mathcal{P}$

— $a_h^t = \begin{cases} 1 & \text{if pattern } h \text{ covers time period } t \in \mathcal{T}\\ 0 & \text{otherwise.} \end{cases}$

— S_h: starting time period of pattern $h \in \mathcal{H}$ ($S_h = \min_{t\in\mathcal{T}} a_h^t t$)

— C_h: completion time period of plan $p \in \mathcal{P}$ ($C_h = \max_{t \in \mathcal{T}} a_h^t t$)

— s_h: skill associated with pattern $h \in \mathcal{H}$

— l_h: location associated with pattern $h \in \mathcal{H}$

— $\mathcal{H}_l$: set of patterns associated with location $l \in \mathcal{L}$

— $\mathcal{R}_h$: set of technicians that can be assigned to pattern h

Jobs

— $\mathcal{J}^{\mathcal{H}}(\bar{x})$: set of jobs associated with the patterns involved in the selection of plans $\bar{x}$

— $\mathcal{J}^{\mathcal{R}}$: set of jobs associated with technician unavailability time periods

— $\mathcal{J}(\bar{x})$: set of jobs associated with the selection of plans $\bar{x}$
$\mathcal{J}(\bar{x}) = \mathcal{J}^{\mathcal{H}}(\bar{x}) \cup \mathcal{J}^{\mathcal{R}}$

— h_j: pattern associated with job j

— q_j: number of technicians required by job j

— S_j: starting time period of job j

— C_j: completion time period of job j

— $\mathcal{S}_j$: set of skills such that a technician should master at least one of these skills in order to perform job j

— l_j: location associated with job j

— $\mathcal{R}_j$: set of technicians that can be assigned to job j

Appendix B

Instance generation

The purpose of this section is to describe how we randomly generate the instances used in our computational experiments.

An instance of the problem is primarily characterized by:

- a finite time horizon (a finite number of time periods)
- a number of time periods per day (yielding the number of days)
- a set of locations (wind farms + home depot)
- a set of wind turbines distributed over the wind farms
- a set of maintenance tasks to perform at the different locations and that impact the availability of the turbines
- a set of technicians to perform the tasks
- wind speed for each time period and location
- postponing penalties

The generator is based on the following parameters:

- $n_{\mathcal{T}}$, $n_{\mathcal{D}}$, $n_{\mathcal{I}}$, $n_{\mathcal{S}}$ (length of time horizon, number of days, number of wind farms, number of tasks, and number of skills)
- $Dn^{\mathcal{L}}$: probability distribution of the number of locations
- Dl^{xy}: probability distribution of the coordinates associated with each location
- $Dn^{\mathcal{L}}_{\mathcal{W}}$: probability distribution of the number of turbines per location
- $Dn^{\mathcal{W}}_{\mathcal{I}}$: probability distribution of the number of tasks per turbine
- Δl^{min}: minimum distance between two locations
- Δr^{max}: maximum distance between two locations such that they can be visited by the a technician during the same day
- K: set of all types of preventive tasks that we consider
- $p(k)$: probability of generating a task of type $k \in K$
- $Di_{impact}(k)$: probability distribution of the impact of each type of preventive task on the wind turbines
- $Di_{dur}(k)$: probability distribution of the duration of each type of preventive task
- $Di_{req}(k)$: probability distribution of the number of technicians that can perform each type of preventive task during any time period
- $Dr_{\#skills}$: probability distribution of the number of skills mastered by a technician
- $Dr_{\mathbb{P}(unv)}$: probability that a technician has some unavailability time periods during the time horizon

— $Dr_{\#unv}$: probability distribution of the number of time periods during which a technician is unavailable

— $Dr_{d^{unv}}$: probability distribution of the duration of the unavailability of a technician (in man-hours)

— Dw_{power}: probability distribution of the nominal power (in kW) of each turbine

— $\widehat{\phi}$: average wind speed on each wind farm

— Υ_{max}^{safety}: maximum wind speed allowed to perform a task

— Δl^{max}: maximum distances for the spatial correlation of the wind speed

— δ: number of values used in the moving average for the time-wise dependency between the wind speeds

— α: correlation factor between wind speed

We generate an instance following multiple steps. First of all, the length of time horizon, the number of days, the number of tasks, and the number of skills are input values. This yields directly the set $\mathcal{T}$ of time periods and the set $\mathcal{D}$ of days.

We then start the generation of an instance by building the set $\mathcal{L}$ of locations whose cardinality is set by sampling the $Dn^{\mathcal{L}}$ distribution. According to the distance Δl^{min}, we then generate the coordinates of each location by sampling the Dl^{xy} distribution. Based on these coordinates and on the distance Δr^{max}, we compute the parameters $(\delta_{ll'})_{(l,l')\in\mathcal{L}^2}$ that enables us to define the daily location-based incompatibility constraints.

Afterwards, we built the set $\mathcal{W}$ of wind turbines. To this end, according to the target number of tasks, we start by generating a number of wind turbines per locations by sampling the $Dn_{\mathcal{W}}^{\mathcal{L}}$ distribution. For each location where there is at least one wind turbine (i.e., this location is a wind farm), we then generate a nominal power by sampling the Dw_{power} distribution and we set the nominal power P_w equal to this latter value for each wind turbine $w \in \mathcal{W}$ of the wind farm.

After that, we call procedure **genTasks**() to create the set $\mathcal{I}$ of tasks. Notice that for each task $i \in \mathcal{I}$ we build the set $\mathcal{M}_i$ of execution modes such that it meets the two following requirements:

— $\forall m, m' \in \mathcal{M}_i$, $q_{im} \neq q_{im'}$,

— $\forall m, m' \in \mathcal{M}_i$, $q_{im} < q_{im'} \rightarrow d_{im} > d_{im'}$.

Arbitrarily, we build $ov(\mathcal{I})$ considering that overlapping tasks are forbidden on the same turbine. Notice that, according to some experts in the field, it is reasonably realistic to only consider these subsets. After the generation of the tasks, we generate the set $\mathcal{R}$ of technicians using procedure **genTechnicians**().

The last part of the generator concerns the parameters related to the objective function. For the sake of convenience, we introduce the set $\mathcal{T}^+$ of all time periods formed by the union of set $\mathcal{T}$ and the set of rest time periods that occur between each day. More specifically, we include a rest time period after every $\frac{|\mathcal{T}|}{|\mathcal{D}|}$ consecutive time periods of $\mathcal{T}$.

As it concerns the wind speed at hub height, the main purpose is to use realistic values. First, we generate wind speed $\bar{\phi}_l^t$ for every location $l \in \mathcal{L}$ and every time period $t \in \mathcal{T}^+$ using a Rayleigh distribution with a scale parameter equal to $\widehat{\phi}\sqrt{\frac{2}{\pi}}$ (so that the expected wind speed is $\widehat{\phi}$). Since space correlation can be significant, we compute a corrected wind speed $\bar{\bar{\phi}}_l^t$ for every location l and every time period t as follows:

$$\bar{\bar{\phi}}_l^t = \frac{\sum\limits_{\substack{l'\in\mathcal{L} \\ s.t.\ \Delta_{ll'}<\Delta l^{max}}} (\Delta l^{max} - \Delta_{ll'})\, \bar{\phi}_{l'}^t}{\sum\limits_{\substack{l'\in\mathcal{L} \\ s.t.\ \Delta_{ll'}<\Delta l^{max}}} (\Delta l^{max} - \Delta_{ll'})}.$$

Procedure genTasks

$\mathcal{I} \leftarrow \emptyset$
for $i \in \{1, ..., n_{\mathcal{I}}\}$ **do**
    ;/* Creation of a new task i */
    Associate randomly a wind turbine to task i by sampling the $Dn_{\mathcal{I}}^{\mathcal{W}}$ distribution
    Define the type $k \in K$ of the task according to the probabilities $p(k)$
    Define the impact of the task on the wind turbines by sampling the $Di_{impact}(k)$ distribution
    Draw randomly the skill s_i required by task i from the set $\mathcal{S}$
    Set the minimal (q_i^{MIN}) and the maximal (q_i^{MAX}) numbers of technicians that can perform task i at any given time period by sampling the $Di_{req}(k)$ $Dn_{\mathcal{I}}^{\mathcal{W}}$
    Generate a task duration d_i by sampling the $Di_{dur}(k)$ distribution
    $n_{\mathcal{M}_i} \leftarrow q_i^{MAX} - q_i^{MIN} + 1$
    $\mathcal{M}_i \leftarrow \emptyset$
    ;/* d_i^{prev}: duration of the last executing mode created for task i */
    for $m \in \{1, ..., n_{\mathcal{M}_i}\}$ **do**
        Create executing mode m for which task i requires $q_i^{\mathcal{M}}$ technicians and lasts $d_i^{\mathcal{M}}$ time periods with:
            $q_i^{\mathcal{M}} \leftarrow q^{MAX} - m + 1$
        ;/* We assume that the duration of a working day is 8 hours. */
            $d_i^{\mathcal{M}} = \max(d_i^{prev} + 1, \lfloor \frac{d_i |\mathcal{T}|}{8 |\mathcal{D}| q_i^{\mathcal{M}}} + 0.5 \rfloor$
        Add the created executing mode to $\mathcal{M}_i$
        $d_i^{prev} \leftarrow d_i^{\mathcal{M}}$
    end
    Add the created task i to $\mathcal{I}$
end

Wind speeds were generated independently from a time period to another one. However, this time-wise independence assumption is unlikely to be verified in practice. To smooth out the speed-values, we use a δ-weighted moving average that yields wind speed ϕ_l^t according to the following formula:

$$\phi_l^t = \frac{\bar{\bar{\phi}}_l^t + \sum_{t'=\max(0,t-\delta)}^{\max(0,t-1)} \alpha^{t-t'} \phi_l^{t'}}{1 + \sum_{t'=\max(0,t-\delta)}^{\max(0,t-1)} \alpha^{t-t'}}.$$

The resulting values are rounded to the nearest tenth. From our perspective, they compare well to realistic data.

Afterwards, for each task $i \in \mathcal{I}$ and every time period $t \in \mathcal{T}$, we compute the binary parameter $\ddot{\vartheta}_i^t$ equal to 1 if and only if $\phi_l^t < \Upsilon_{max}^{safety}$ (i.e. the task i can be scheduled during time period t according to safety concerns). Arbitrarily, we set each parameter ϑ_i^t equal to 1 for every task i and every time period t. We point out here that this choice makes the instances more complicated to solve as there is a wide flexibility to schedule the maintenance operations. This also matches field observations.

The last step consists in computing the revenue value g_w^t for every wind turbine $w \in \mathcal{W}$ during each time period $t \in \mathcal{T}^+$. We compute the revenue from the nominal power P_w of the wind turbine and from the wind speed ϕ_l^t. We also use an estimation $hours(t)$ of the number of hours during every time period t. More specifically, we compute the revenue g_w^t generated by each turbine $w \in \mathcal{W}$ that is available during time period $t \in \mathcal{T}$ as follows:

$$g_w^t = 0.08 \cdot P_w \cdot hours(t) \cdot CF(\phi_{l_w}^t).$$

Procedure genTechnicians

Let d^{unv} be the average number of time periods during which a technician is not available according to $Dr_{\#unv}$ and $Dr_{d^{unv}}$.
$\mathcal{R} \leftarrow \emptyset$
for $s \in \{1, ..., n_{\mathcal{S}}\}$ **do**
  ;/* compute the average total request of the tasks RS_s^{avg} */

$$RS_s^{avg} = \sum_{\substack{i \in \mathcal{I} \\ s_i = s}} \frac{1}{|\mathcal{M}_i|} \sum_{m \in \mathcal{M}_i} q_{im}$$

  ;/* n_s minimum number of technicians mastering skill s */

$$n_s \leftarrow \epsilon \cdot \frac{RS_s^{avg}}{d^{unv}}$$

  for $r \in \{1, ..., n_s\}$ **do**
    Create a technician mastering skills s and generate his or her unavailability time periods by sampling the $Dr_{\#unv}$ and $Dr_{d^{unv}}$ distributions
    Add this technician to $\mathcal{R}$
  end
end
for $r \in |\mathcal{R}|$ **do**
  Sample the $Dr_{\#skills}$ distribution to generate the number of skills mastered by technician r
  According to the previous value, generate additional skills for technician r
end

where

- 0.08: is an approximation to the selling price in euros of 1 kWh of wind energy (this selling price is guaranteed for the next 10 years in France).
- $hours(t)$: estimation of the number of hours during time period $t \in \mathcal{T}^+$
- P_w: nominal power of wind turbine $w \in \mathcal{W}$
- $\phi_{l_w}^t$: wind speed during time period t at the location of turbine $w \in \mathcal{W}$
- $CF(\phi)$ [1]: the ratio of the net electricity generated according to a wind speed equal to ϕ to the electricity that could have been generated at full-power operation (this ratio is given by a piece-wise linear function estimated from real data)

Finally, we compute a single postponing penalty set equal for each task. This penalty is equal to the maximum loss of revenue that can be generated by the scheduling of a task of $\mathcal{I}$ plus one. With this definition, we almost always (if not always) ensure that postponing a task is non-profitable. With this penalty we therefore almost ensure to schedule the maximum number of tasks during the time horizon according to the availability of the technicians. This is quite in line with the practice in the field.

Table B.1 presents the detail parameter setting used in the generation process.

1. see Figure 3.3

Table B.1 – Detail parameter setting used by the instance generator

Parameters	**Values**							
$(n_{\mathcal{T}}, n_{\mathcal{D}})$	$(10, 5)$		$(20, 5)$		$(20, 10)$		$(40, 10)$	
$n_{\mathcal{I}}$	20	40	20	40	40	80	40	80
$n_{\mathcal{S}}$	$1 \ or \ 3$							
$Dn^{\mathcal{L}}$	$\mathcal{U}(\{5, ..., 10\})$	$\mathcal{U}(\{8, ..., 12\})$	$\mathcal{U}(\{5, ..., 10\})$	$\mathcal{U}(\{8, ..., 12\})$	$\mathcal{U}(\{8, ..., 12\})$	$\mathcal{U}(\{12, ..., 20\})$	$\mathcal{U}(\{8, ..., 12\})$	$\mathcal{U}(\{12, ..., 20\})$
Dl^{xy}	$\mathcal{U}(\{0, ..., 80\}) \times \mathcal{U}(\{0, ..., 80\})$							
$Dn^{\mathcal{L}}_{\mathcal{W}}$	$\mathcal{U}(\{1, 2, ..., 12\})$							
$Dn^{\mathcal{W}}_{\mathcal{I}}$	$\mathcal{U}(\{0, 1, 2\})$							
K	main maintenance tasks				inspection, gearbox oil change, other operations...			
$p(k)$	0.5				0.5			
$Di_{impact}(k)$	$\mathcal{U}(\{ST, BL\})$				$\mathcal{U}(\{NST, ST, BL\})$			
$Di_{dur}(k)$	$\mathcal{U}(\{20, 40, 60, 80\})$				$\mathcal{U}(\{8, 16\})$			
$Di_{req}(k)$	$2, 3, 4$				$2, 3$			
$Dr_{\#skills}$	$\mathcal{U}(\{1, ..., \lvert S \rvert\})$							
$Dr_{\mathbb{P}(unv)}$	0.25							
$Dr_{\#unv}$	1							
$Dr_{d^{unv}}$	$\mathcal{U}(\{1, ..., 2\})$							
ϵ	$1 \ or \ 1.2$							
Dw_{power}	$\mathcal{U}(\{2000, 2500, 3000\}$							
Δl^{min}	5							
Δr^{max}	25							
$\widehat{\phi}$	6							
Υ^{safety}_{max}	12							
Δl^{max}	15							
α	0.9							
δ	1		2		1		2	
$hours(t)$	$hours(t) = \begin{cases} 14 & \text{if } t \text{ is a rest time period} \\ 5^* & \text{if } t \text{ is not a rest time period and } \frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} = 2 \\ 2.5^* & \text{if } t \text{ is not a rest time period and } \frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} = 4 \end{cases}$							
$CF(\phi)$	piecewise linear approximation with the following 9 points (see below)							

ϕ	0	3.5	5.5	7	12.5	14	24.9	25	30
$CF(\phi)$	0	0	0.1	0.23	0.91	1	1	0	0

* We consider that the turbine is stopped slightly longer than the duration of a task (that has an impact on its execution) because the turbine has often to be stopped a bit earlier than the starting time of this task .

NS No turbines are shut down during the execution of the task.

ST The task impacts only the turbine on which it is executed. This turbine is stopped during the execution of the task (not during the rest time periods the task overlaps).

BL The task impacts only the turbine on which it is executed. This turbine is stopped during the execution of the task and during the rest time periods the task overlaps.

Remark: The model can also consider tasks that shut down multiple turbines in a wind farm (e.g., the maintenance of the wind farm substation), but this is extremely rare in practice.

Appendix C

Detailed computational results

C.1 CP formulation

Table C.1 – Computational results on testbed G1 for the direct resolution of the CP formulation (BS)

Family	15s		30s		60s		180s		300s	
	Gap	%S	Gap	%S	Gap	%S	Gap	%S	Gap	%S
10_2_1_20_A	9.8%	93%	9.8%	93%	9.8%	93%	9.7%	93%	9.7%	93%
10_2_1_20_B	3.5%	99%	3.5%	99%	3.5%	99%	3.5%	99%	3.5%	99%
10_2_1_40_A	8.3%	95%	8.3%	95%	8.3%	95%	8.3%	95%	8.3%	95%
10_2_1_40_B	1.7%	99%	1.7%	99%	1.7%	99%	1.7%	99%	1.7%	99%
10_2_3_20_A	11.8%	90%	11.7%	90%	11.7%	90%	11.7%	90%	11.7%	90%
10_2_3_20_B	1.1%	100%	1.1%	100%	1.1%	100%	1.1%	100%	1.1%	100%
10_2_3_40_A	15.1%	93%	15.1%	93%	15.1%	93%	15.1%	93%	15.1%	93%
10_2_3_40_B	2.9%	99%	2.9%	99%	2.9%	99%	2.9%	99%	2.9%	99%
20_2_1_40_A	11.5%	94%	11.2%	94%	11.2%	94%	11.2%	94%	11.2%	94%
20_2_1_40_B	0.4%	100%	0.4%	100%	0.4%	100%	0.4%	100%	0.4%	100%
20_2_1_80_A	11.2%	94%	11.2%	94%	11.2%	94%	11.2%	94%	11.2%	94%
20_2_1_80_B	0.7%	100%	0.7%	100%	0.7%	100%	0.7%	100%	0.7%	100%
20_2_3_40_A	6.1%	94%	5.6%	95%	5.6%	95%	5.6%	95%	5.6%	95%
20_2_3_40_B	0.7%	100%	0.7%	100%	0.7%	100%	0.7%	100%	0.7%	100%
20_2_3_80_A	3.0%	99%	3.0%	99%	3.0%	99%	3.0%	99%	3.0%	99%
20_2_3_80_B	0.5%	100%	0.5%	100%	0.5%	100%	0.5%	100%	0.5%	100%
20_4_1_20_A	18.8%	87%	18.3%	87%	18.2%	87%	18.2%	87%	18.2%	87%
20_4_1_20_B	19.1%	90%	19.1%	90%	19.1%	90%	17.6%	91%	17.6%	91%
20_4_1_40_A	23.6%	88%	21.9%	89%	21.9%	89%	21.9%	89%	21.9%	89%
20_4_1_40_B	6.2%	96%	6.2%	96%	6.2%	96%	6.2%	96%	6.2%	96%
20_4_3_20_A	19.0%	89%	18.9%	89%	18.9%	89%	18.8%	89%	18.8%	89%
20_4_3_20_B	5.5%	96%	5.4%	96%	5.4%	96%	5.4%	96%	5.4%	96%
20_4_3_40_A	14.4%	91%	13.1%	91%	13.1%	91%	13.1%	91%	13.1%	91%
20_4_3_40_B	7.1%	96%	7.1%	96%	7.1%	96%	7.1%	96%	7.1%	96%
40_4_1_40_A	21.6%	85%	19.3%	86%	18.5%	87%	18.5%	87%	18.5%	87%
40_4_1_40_B	1.2%	100%	1.2%	100%	1.2%	100%	1.2%	100%	1.2%	100%
40_4_1_80_A	14.1%	91%	14.1%	91%	14.1%	91%	14.1%	91%	14.1%	91%
40_4_1_80_B	3.5%	98%	3.5%	98%	3.0%	99%	3.0%	99%	3.0%	99%
40_4_3_40_A	13.8%	91%	12.9%	91%	12.9%	91%	12.9%	91%	12.9%	91%
40_4_3_40_B	3.1%	98%	3.1%	98%	3.1%	98%	3.1%	98%	3.1%	98%
40_4_3_80_A	18.8%	89%	18.8%	89%	18.8%	89%	18.8%	89%	18.8%	89%
40_4_3_80_B	2.7%	98%	2.7%	98%	2.7%	98%	2.7%	98%	2.7%	98%

Family	15s		30s		60s		180s		300s	
	Gap	%S	Gap	%S	Gap	%S	Gap	%S	Gap	%S
$\lvert\mathcal{S}\rvert = 1$	9.7%	94%	9.4%	94%	9.3%	94%	9.2%	94%	9.2%	94%
$\lvert\mathcal{S}\rvert = 3$	7.9%	95%	7.7%	95%	7.7%	95%	7.7%	95%	7.7%	95%
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} = 2$	5.5%	97%	5.5%	97%	5.5%	97%	5.5%	97%	5.5%	97%
$\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{D}\rvert} = 4$	12.0%	93%	11.6%	93%	11.5%	93%	11.4%	93%	11.4%	93%
Type = A	13.8%	91%	13.3%	92%	13.3%	92%	13.3%	92%	13.3%	92%
Type = B	3.8%	98%	3.7%	98%	3.7%	98%	3.6%	98%	3.6%	98%
All	8.8%	95%	8.5%	95%	8.5%	95%	8.4%	95%	8.4%	95%

C.2 CPLNS

Table C.2 – Computational results on testbed G1 for the CPLNS with a time limit of 300 seconds (average over 10 runs)

Family	Mean	Best	Worst
10_2_1_20_A	0.75%	0.70%	0.87%
10_2_1_20_B	0.03%	0.00%	0.03%
10_2_1_40_A	0.42%	0.11%	0.95%
10_2_1_40_B	0.02%	0.01%	0.03%
10_2_3_20_A	0.60%	0.37%	0.79%
10_2_3_20_B	0.01%	0.01%	0.03%
10_2_3_40_A	1.05%	0.22%	1.78%
10_2_3_40_B	0.05%	0.01%	0.08%
20_2_1_40_A	0.86%	0.20%	1.68%
20_2_1_40_B	0.04%	0.02%	0.09%
20_2_1_80_A	2.24%	1.49%	2.92%
20_2_1_80_B	0.06%	0.04%	0.08%
20_2_3_40_A	0.58%	0.47%	0.70%
20_2_3_40_B	0.02%	0.01%	0.04%
20_2_3_80_A	0.34%	0.26%	0.43%
20_2_3_80_B	0.07%	0.05%	0.09%
20_4_1_20_A	1.03%	0.96%	1.28%
20_4_1_20_B	0.09%	0.01%	0.21%
20_4_1_40_A	5.30%	4.52%	5.81%
20_4_1_40_B	0.24%	0.09%	0.87%
20_4_3_20_A	2.02%	1.03%	3.52%
20_4_3_20_B	0.23%	0.04%	0.42%
20_4_3_40_A	3.86%	2.78%	4.35%
20_4_3_40_B	0.46%	0.17%	0.71%
40_4_1_40_A	4.37%	4.26%	4.50%
40_4_1_40_B	0.13%	0.07%	0.20%
40_4_1_80_A	5.38%	3.99%	6.67%
40_4_1_80_B	0.23%	0.16%	0.30%
40_4_3_40_A	2.79%	1.71%	3.95%
40_4_3_40_B	0.12%	0.06%	0.19%
40_4_3_80_A	5.62%	4.86%	6.76%
40_4_3_80_B	0.17%	0.12%	0.22%

C.3 Branch-and-check

Table C.3 – Description of the average number of cuts generated in the B&C approach for testbed G1.

Family	All	CB	Other cuts		
			BF	MCbM	MWC
10_2_1_20_A	21	0	0	2	19
10_2_1_20_B	10	0	0	1	9
10_2_1_40_A	36	0	0	0.8	35
10_2_1_40_B	12	0	0	0.6	12
10_2_3_20_A	153	0	94	9	50
10_2_3_20_B	28	0	7	6	15
10_2_3_40_A	66	0	16	9	41
10_2_3_40_B	17	0	0.2	2	15
20_2_1_40_A	67	0	0	1	66
20_2_1_40_B	28	0	0	1	27
20_2_1_80_A	105	0	0	3	101
20_2_1_80_B	18	0	0	1	17
20_2_3_40_A	144	0	17	7	120
20_2_3_40_B	40	0	5	2	32
20_2_3_80_A	118	0	6	5	107
20_2_3_80_B	31	0	0.6	0.8	30
20_4_1_20_A	94	0	0	5	90
20_4_1_20_B	26	0	0	3	23
20_4_1_40_A	202	0	5	12	185
20_4_1_40_B	57	0	0	3	54
20_4_3_20_A	314	0	34	16	263
20_4_3_20_B	84	0	21	13	50
20_4_3_40_A	392	0	38	26	329
20_4_3_40_B	96	0	20	10	66
40_4_1_40_A	229	0	0	3	226
40_4_1_40_B	91	0	0	2	88
40_4_1_80_A	480	0	0	5	476
40_4_1_80_B	151	0	0	4	146
40_4_3_40_A	672	0	65	19	588
40_4_3_40_B	144	0	9	6	129
40_4_3_80_A	1,106	0	102	33	971
40_4_3_80_B	184	0	5	5	174

Table C.4 – Detailed computational results on testbed G1 of the B&C approach coupled with the CPLNS.

Family	B&C			CPLNS	
	Gap	#Opt	Time	Gap[1]	Gap[2]
10_2_1_20_A	-	5/5	9	-	2.6%
10_2_1_20_B	-	5/5	3	-	0.48%
10_2_1_40_A	-	5/5	8	-	3.1%
10_2_1_40_B	-	5/5	3	-	0.52%
10_2_3_20_A	-	5/5	101	-	1.1%
10_2_3_20_B	-	5/5	3	-	0.09%
10_2_3_40_A	-	5/5	13	-	2.5%
10_2_3_40_B	-	5/5	3	-	0.44%
20_2_1_40_A	-	5/5	115	-	0.82%
20_2_1_40_B	-	5/5	5	-	0.22%
20_2_1_80_A	0.02%	4/5	288	5.3%	1.1%
20_2_1_80_B	-	5/5	7	-	0.16%
20_2_3_40_A	0.16%	4/5	35	2.2%	0.28%
20_2_3_40_B	-	5/5	4	-	0.07%
20_2_3_80_A	-	5/5	57	-	0.72%
20_2_3_80_B	-	5/5	6	-	0.15%
20_4_1_20_A	2.1%	3/5	1,130	2.1%	0.29%
20_4_1_20_B	-	5/5	4	-	1.2%
20_4_1_40_A	1.3%	2/5	2,188	4.4%	5.1%
20_4_1_40_B	-	5/5	8	-	2.7%
20_4_3_20_A	2.0%	4/5	340	2.0%	4.3%
20_4_3_20_B	-	5/5	11	-	1.0%
20_4_3_40_A	0.64%	1/5	9,416	3.0%	2.4%
20_4_3_40_B	-	5/5	13	-	1.8%
40_4_1_40_A	2.0%	0/5	-	2.9%	-
40_4_1_40_B	-	5/5	30	-	0.19%
40_4_1_80_A	1.4%	0/5	-	4.5%	-
40_4_1_80_B	-	5/5	108	-	0.22%
40_4_3_40_A	0.51%	0/5	-	1.1%	-
40_4_3_40_B	-	5/5	36	-	0.33%
40_4_3_80_A	1.8%	0/5	-	4.3%	-
40_4_3_80_B	-	5/5	70	-	0.20%

[1] Takes into account the instances for which the time limit is reached in the B&C approach.
[2] Takes into account the instances solved to optimality by the B&C approach.

List of Tables

List of Figures

Acronyms

ACO ant colony optimization. 36–39

ALNS adaptive large neighborhood search. 44, 45, 81

B&C branch-and-check. 14, 43, 47, 59, 95, 110, 112, 114, 115, 119, 120, 131, 142, 146, 171–175

CP constraint programming. 13, 14, 35–39, 41–43, 45, 46, 55, 79, 81–83, 86–88, 90, 91, 93, 110, 145, 146, 169, 173–175

CPLNS constraint programming-based large neighborhood search. 14, 59, 79, 81, 88–91, 114, 115, 139, 143, 145, 170, 172–175

DISCO distribution company. 22, 23, 37

FOR forced outage rate. 29, 37

GA genetic algorithm. 30, 33–40

GENCO generation company. 22, 23, 26–28, 30, 36, 37

GMS generation maintenance scheduling. 24, 26–40

ILP integer linear programming. 14, 32, 41, 42, 50, 59, 69, 71, 72, 75–79, 84, 95, 96, 98, 109–112, 114, 115, 119, 120, 127, 132, 173

ISO independent system operator. 22, 26–28, 37

LBBD logic-based Benders decomposition. 43

LNS large neighborhood search. 13, 44, 45, 55, 81

LOLP loss of load probability. 29, 30, 35

LP linear programming. 13, 33, 35, 38, 39, 41–43, 45, 46, 49, 50, 55, 110, 111, 129, 132, 143

MCbM maximum cardinality b-matching. 101, 103, 104, 106–112, 114, 143, 171

MILP mixed-integer linear programming. 33, 37–39, 41–43, 45, 50, 51, 55, 131, 132

MWC maximum-weight clique. 103, 105–112, 114, 143, 171

OR Operations Research. 11–13, 21, 24, 32, 37, 51, 91, 145

PSO particle swarm optimization. 34, 35, 37, 38, 40

RCM reliability-centered maintenance. 11, 23, 32

RETAILCO retail energy service company. 22

SA simulated annealing. 34, 35, 37–40

TMS transmission maintenance scheduling. 28, 32, 33, 36–40

TRANSCO transportation company. 22, 23, 26, 28, 37

TS tabu search. 34, 36–38

Glossary

AF The ratio of the duration that a generating unit is available to provide electricity to the grid, for the time considered, to the duration of the same time period. 10, 61, 119

CF The ratio of the net electricity generated, for the time considered, to the electricity that could have been generated at continuous full-power operation during the same time period. 10, 12, 61, 65, 91, 93, 126, 127, 130, 141, 146

UNIVERSITE BRETAGNE LOIRE

université angers

Thèse de Doctorat

Aurélien Froger

Planification de la maintenance d'équipements de production d'électricité : une attention particulière portée sur un problème de l'industrie éolienne terrestre

Maintenance scheduling in the electricity industry: a particular focus on a problem rising in the onshore wind industry

Résumé

L'optimisation de la planification de la maintenance des équipements de production d'électricité est une question importante pour éviter des temps d'arrêt inutiles et des coûts opérationnels excessifs. Dans cette thèse, nous présentons une classification multidimensionnelle des études de Recherche Opérationnelle portant sur ce sujet. Le secteur des énergies renouvelables étant en pleine expansion, nous présentons et discutons ensuite d'un problème de maintenance de parcs éoliens terrestres. Le problème est traité sur un horizon à court terme et l'objectif est de construire un planning de maintenance qui maximise le revenu lié à production d'électricité des éoliennes tout en prenant en compte des prévisions de vent et en gérant l'affectation de techniciens. Nous présentons plusieurs modélisations du problème basées sur la programmation linéaire. Nous décrivons aussi une recherche à grands voisinages basée sur la programmation par contraintes. Cette méthode heuristique donne des résultats probants. Nous résolvons ensuite le problème avec une approche exacte basée sur une décomposition du problème. Dans cette méthode, nous construisons successivement des plannings de maintenance optimisés et rejetons, à l'aide de coupes spécifiques, ceux pour lesquels la disponibilité des techniciens est insuffisante. Les résultats suggèrent que cette méthode est la mieux adaptée pour ce problème. Enfin, pour prendre en compte l'incertitude inhérente à la prévision de vitesses de vent, nous proposons une approche robuste dans laquelle nous prenons des décisions garantissant la réalisabilité du planning de maintenance et le meilleur revenu pour les pires scénarios de vent.

Mots clés

recherche opérationnelle, planification, maintenance dans le secteur de l'électricité, éolien terrestre, programmation linéaire en nombre entiers, programmation par contraintes, branch-and-check, optimisation robuste.

Abstract

Efficiently scheduling maintenance operations of generating units is key to prevent unnecessary downtime and excessive operational costs. In this work, we first present a multidimensional classification of the body of work dealing with the optimization of the maintenance scheduling in the Operations Research literature. Motivated by the recent emergence of the renewable energy sector as an environmental priority to produce low-carbon power electricity, we introduce and discuss a challenging maintenance scheduling problem rising in the onshore wind industry. Addressing the problem on a short-term horizon, the objective is to find a maintenance plan that maximizes the revenue generated by the electricity production of the turbines while taking into account wind predictions, multiple task execution modes, and technician-to-task assignment constraints. We start by presenting several integer linear programming formulations of the problem. We then describe a constraint programming-based large neighborhood search which proves to be an efficient heuristic solution method. We then design an exact branch-and-check approach based on a decomposition of the problem. In this method, we successively build maintenance plans while discarding – using problem-specific cuts – those that cannot be performed by the technicians. The results suggest that this method is the best suited to the problem. To tackle the inherent uncertainty on the wind speed, we also propose a robust approach in which we aim to take risk-averse decisions regarding the revenue associated with the maintenance plan and its feasibility.

Key Words

operations research, scheduling, maintenance in the electricity industry, onshore wind farms, integer linear programming, constraint programming, branch-and-check, robust optimization.