Université Catholique de Louvain

Ecole Polytechnique de Louvain

Academic Year 2013-2014

Co-optimization of Gas Forward Contracts and Unit Commitment

In Partial Fulfillment
of the Requirements for the Degree
Master in Mathematical Engineering

Author:
Adrien Baland

Supervisor:
Pr. Anthony Papavasiliou[a]

Examiners:
Pr. Pierre-Antoine Absil[b]
Dr. Dimitri Tomanos[c]

May 2014

[a]Department of Mathematical Engineering, CORE & GDF-Suez Chair, UCL
[b]Department of Mathematical Engineering, ICTEAM Institute & EPL, UCL
[c]CEEME, GDF-Suez

Acknowledgements

Above all, I first would like to express my deepest gratitude to my supervisor, Prof. Anthony Papavasiliou, for always finding the time to help me whenever I needed it and guiding me through this thesis.

I would like to thank Dr. Dimitri Tomanos for providing me with all the necessary data, giving me feedback, and pretty much acting as a co-supervisor during the past year.

I am also grateful to the people at GDF-Suez Louvain-La-Neuve for their help in the defense preparation and their feedback.

Besides, thank you to Prof. Pierre-Antoine Absil for agreeing to be part of the defense jury.

Furthermore I would like to thank Antoine and Michael, without whom I probably would not be where I am now.

Finally, my gratitude goes to my family and friends, especially my mother, for giving me the motivation and supporting through all my years at this university.

Contents

Appendix

Introduction

Whether it be manufacturing, aerospace, finance, energy, . . . , industries have always used models to assess what actions need to be done in order to achieve their goals. For private companies such as GDF-Suez, this goal usually is to maximize profit, which can only be done effectively if the said models embed a good representation of what reality is.

Despite the always increasing computational power of computers, having too large an amount of details in these models to get an ideal representation generally results in solving times that are way too high to be useful in practice using state-of-the-art solvers, and usually forces these companies to make a choice between setting aside the complicating details or finding alternative ways to solve the problem. The former, although easier to do, can often result in lack of precision, ergo possible money losses due to changes in decision caused by these now-removed details. The latter is trickier to do since there is no predefined guaranteed-to-work way to increase efficiency and usually consists in finding a model-specific approach that will help make the solving task easier for the solver.

This increase in model complexity is commonly encountered when a large time solving horizon combined with a small granularity are involved, e.g. a yearly horizon with an hourly granularity for the decision process. Such a scheme may for example be found in one of GDF-Suez's optimization models used to determine a one-year activity schedule for their power plants, for which a computational limit of one hour is desired. Although GDF-Suez managed to design a way that returned a solution before this deadline, consisting in the aggregation of time periods to get a larger granularity, leading to improvements in terms of efficiency, this solution is not always satisfactory to them. The purpose of this thesis is to get an acceptable solution fast enough.

First, we expand the different features of GDF-Suez's model that we replicate in ours, features leading to obtain a problem that cannot be solved within the hour. Then, we present the construction of the model we ended up trying to solve. After the model is built, we explain our solving algorithm which is divided in three blocks : an initialization phase where we use simplified versions of the model to get some initial variables and parameter values, a bounding scheme where we break down the full model using Lagrangian relaxation, an optimization technique, to create smaller subproblems easier to solve, and a feasible solution recovery method where we use the bounding scheme and the initialization phases to try coming up with feasible solutions to the problem. Finally, because the structure of energy markets may be different from one location to another, we try to stay as general as possible when building both the model and the algorithm, and conclude by running some robustness tests in an attempt to validate the effectiveness of our algorithm.

Chapter 1

Model scope

Companies in the energy industry do not have the luxury of representing the whole reality of things, which would make their models contain millions and millions (billions and billions?) of parameters just to get small improvements in the details, barely significant, while letting complexity explode.

Developers therefore have to select the worthy enough *features* the model will take on board. Some of these are of course “must-haves” (such as maximum output capacity) but others may sometimes be less obvious to grasp, which is why in the following sections, we present and show graphs of the features that we have decided to retain in order to get an optimization problem that can help companies like GDF-Suez find yearly activity schedules for their generators with an objective of profit maximization.

Keeping in mind that our model is dynamic (that is, it takes into account more than one time period), we present a first specific feature related to the transitions from one time period to the next (see Section 1.1). Then, we elaborate on the different configurations, called *modes*, among which generators have to select when they want to produce power, each with its own technical characteristics such as minimal/maximum output, startup costs,... (see Section 1.2).

Once this is done, we move on to the piecewise linear fuel consumption description retained for the underlying cost of producing power (see Section 1.3), and introduce the concept of temperature-dependency for power plants that have to be started up (see Section 1.4). To conclude with the way generators work, we also present what is called *profiles*, a pre-fixation of production levels for a short time period whenever power plants have to be started up or shut down (Section 1.5).

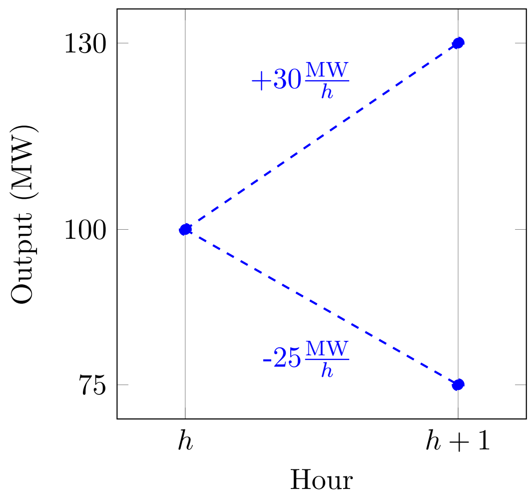
After that, we focus on the fuel used by the generators whose costs are given by forward contracts, and the network that is used for conveying it to the power plants (Section 1.6). Finally, we mention some regulations imposed by the System Operator on the whole power network through *ancillary services*, that consist in forcing generators to leave a security margin between what they are able to produce and what they actually produce, to intervene in case of unforeseen events that could cause a black-out on the power grid (see Section 1.7).

1.1 System dynamics and transitions

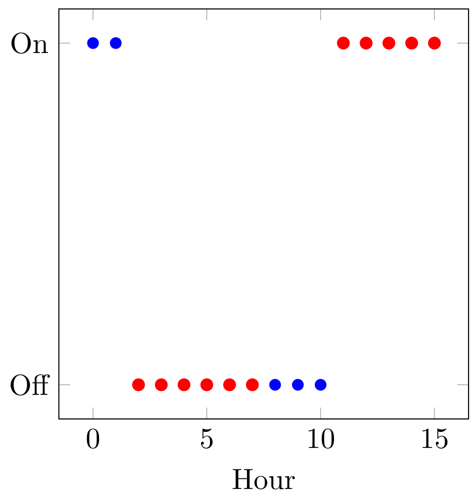
When looking at generator scheduling problems over a large time period, two characteristics have to be implemented in order to avoid safety issues and stability problems : ramp rates and minimum up/down times. The former bounds the increase or decrease in production that a generator is allowed to make between two consecutive time periods, as shown in Example 1, the latter prevents starting up and shutting down a power plant in alternance, by forcing a plant to stay active for a given amount of time periods whenever it has been started up or by forcing it to stay inactive whenever it has been shut down, as shown in Example 2.

Example 1 *A generator that produces* 100 MW *at hour* h *cannot produce less than* 75 MW *or more than* 130 MW *at hour* $h+1$ *if it has a 25*MW $/h$ *ramp down rate and a* 30 MW $/h$ *ramp up rate.*

Example 2 (Taken from [Sch10]) *Coal power plants have a technically mandated minimum down time of approximately 6 to 15 hours, meaning they cannot be started up for 6 to 15 hours once they have been shut down. Their minimum operating time also has the same length.*



(a) Ramp rates from Example 1, the two marks in hour $h+1$ represent the upper and lower limits for this hour, due to the production of 100MW in hour h.



(b) Min up/down times from Example 2, red marks correspond to forced states due to minimum up/down times and blue marks are free choices of (in)activity for the generator.

Figure 1.1: Illustrations for the two examples from Section 1.1

1.2 Multiple modes in a power plant

The first real feature we introduce in our model (the previous section was more of a must-have than a real feature) is what is called working modes, or *modes*. They represent the fact that for some generators, choices can be made regarding the part of the whole power plant which is on. For example, consider a plant composed of two gas turbines, 1 and 2, such that turbine 2 cannot be working if turbine 1 is not also active at the same time.

When we start up this plant, we have to choose between starting turbine 1 only, or starting both of them altogether. Besides, we cannot decide to switch from one working turbine at time t, to both working at time $t+1$, then back to 1 at time $t+2$ and alternate every hour. These kinds of decisions have to be made for a given amount of time before we are allowed to alternate. This illustration also points to another fact : the technical characteristics for this power plant depend on the working mode we decide to use, e.g. the minimum and maximum production capacity for the whole power plant will be different if we choose to use turbine 1 only or turbine 1 and 2 simultaneously.

Although this example of possible modes is the easiest to illustrate, the data we received from GDF-Suez contain another type of mode, an *economy mode*, which is used when a power plant becomes unprofitable for a small amount of time but has a large startup cost. Because of this high startup cost, shutting down the plant when it becomes unprofitable leads to a high price when the decision of bringing it back up is made. Economy modes are working modes with a very small production capacity, so that a power plant can be kept active while avoiding high costs when producing power is not cost-effective, all the while also preventing shutting down the plant and incurring a startup cost in the future. Figure 1.2 shows an example of such working modes, where we only display the production capacity range for these modes, because it is the easiest characteristic to visualize. In this Figure, Mode0 represents an Off mode, corresponding to an inactive generator, Mode1 an economy mode, and (Mode2,Mode3) can illustrate a “One or two turbines active” situation

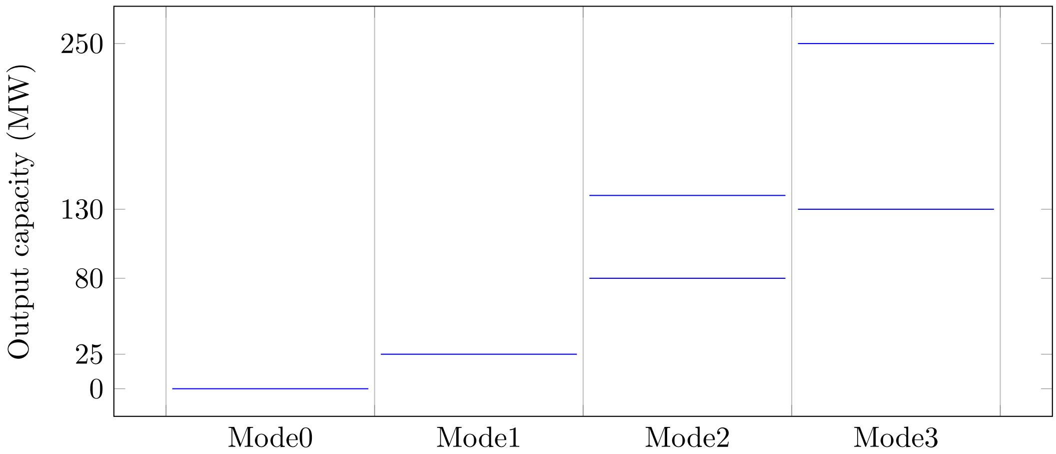


Figure 1.2: Example of minimum and maximum energy output for different modes

1.3 Nonlinear heat-rate curve

To produce electricity, one has to realize that the function linking fuel consumption and electricity output is not a linear one, even though it may be close to one, which is why many models approximate this function by such a curve. Ignoring once and for all the nonlinearities of the fuel-to-production curve as most models do is easy and in general greatly simplifies the computation tasks for a solver, but could cause great inaccuracy in the true cost evaluation of this production. On the contrary, describing the exact shape for this function simply renders the problem too massive and complex to be solvable in practice.

In an attempt to compromise between accuracy of the cost curve and solvability, we approximate it here by a piecewise linear function, where we try to keep the number of "segments" small. Using data shown in [Kle98], we construct Figure 1.3, which shows how the production curve may be in reality, how some models approximate it and how ours does. This figure also points out the fact that there may be multiple linear approximations for the curve, depending on the choice between matching approximation and reality at both ends (continuous black line) or at some other output levels (dashed black line). For example, if we assume the generator has a minimum output of 1 MW, a justification for the second linear approximation (dashed black line) may be that the higher inaccuracy in the fuel consumption before this production level will not have an impact in the evaluation of the fuel consumption that is made by the model because the generator cannot produce less than 1 MW of power. Besides, the increasing pattern for the heat-rate curve in this Figure is not a generality in practice and an decreasing pattern or a firstly decreasing curve then increasing one may occur for different types of power plant.

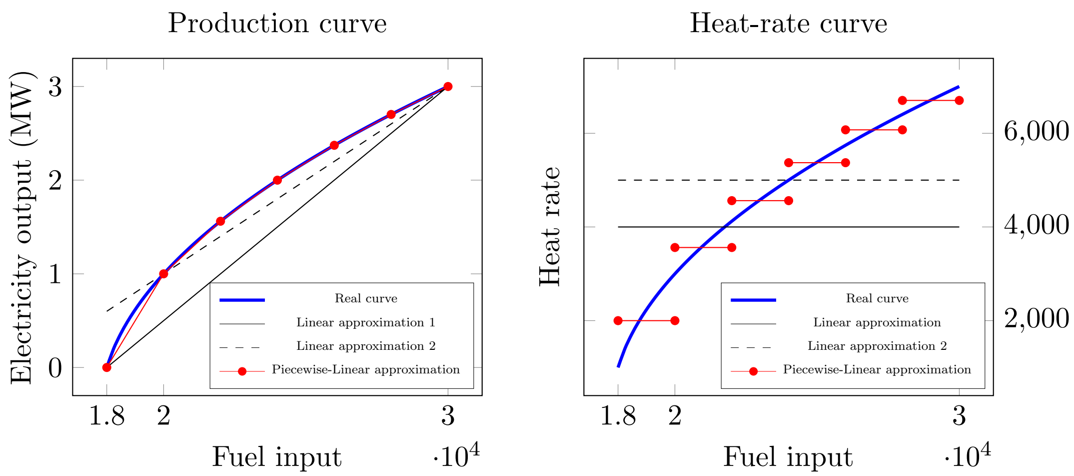


Figure 1.3: Example of a production curve, heat-rate curve and how they may be approximated by different models.

As we will see when building the mathematical model, each segment which is added for a better approximation of this curve introduces a new binary variable to our problem, hence increases the algorithmic complexity, which is why the number of segments used is chosen small, to control time increase while giving us what is considered a good enough approximation nonetheless.

1.4 Temperature-dependent startups

A power plant needs a certain amount of fuel to be fired up and able to produce power. If a power plant was shut down a few hours ago and the decision to bring it back up is made, the generator is still hot and is "more ready" to produce power than if it hasn't been active for 2 months, meaning it will consume less fuel to be started up. A function which is sometimes used for this startup fuel consumption is the following

$$K_a + K_b(1 - e^{-t}),$$

where K_a and K_b are two parameters that vary from a generator to another, the latter creating the temperature dependency.

As in the previous section, we attempt to reproduce this phenomenon without getting too detailed and having the whole description of the full temperature-dependent startup consumption curve. To do so, we divide the possible temperatures of the generator into a small number of temperature intervals, each one associated to a given startup fuel consumption.

Example 3 *A generator could have three temperature states :*

1. *hot : between shutdown time and 24 hours after, 500 giga-joules (GJ) fuel is necessary for starting up;*
2. *warm : between 25 hours after shutdown and 48 hours after, 550 GJ;*
3. *cold, from 49 hours after shutdown on, 580 GJ*

1.5 Startup/Shutdown profiles

Besides the previously mentioned fuel consumption, another aspect has to be taken into account when representing the startup of power plants. Assume we have a generator starting up at time t, we cannot always expect this generator to be fully able to produce any desired level of power (within capacity) immediately after. It will sometimes have to pass through some *checkpoints* and will only be able to freely produce power at time $t + \Delta t$. The following example shows one case where this kind of scenario could happen[1].

Example 4 *Suppose we have a power plant with minimum and maximum capacity of* $100\,\mathrm{MW}$ *and* $200\,\mathrm{MW}$ *respectively. Assume furthermore that this generator has ramp rates parameters of* $50\frac{\mathrm{MW}}{h}$*. If we start this generator at time* t*, the production between* t *and* $t+1$ *will have to be forced to respectively* 50 *and* 100 *because the ramping constraint bounds the difference between two consecutive hours of production from above by* 50*, and the minimum capacity for the power plants forces it to produce* $100\,\mathrm{MW}$ *or more as fast as possible. This means that the sequence of checkpoints that characterizes the startup profile should be* (50, 100).

The above example talked about startup profiles. The same rationale may be used for shutting down a generator. If we have a generator producing at full capacity, we cannot always shut it down the next hour without raising technical difficulties or taking risks with the plant's safety. This is why shutdown profiles are also introduced.

Figure 1.4 shows an example of a plant with both a startup and a shutdown profile. As it is displayed in this Figure, the consecutive production levels during a profile phase can both be below or above minimal capacity and the only purpose of Example 4 is to give a specific case of profile occurrence. We may also consider that we have profiles that violate ramping constraints.

1.6 Fuel delivery

As we mentioned earlier, our power plants need fuel to produce electricity. Three questions arise from this simple fact :

[1]This example illustrates a situation where the startup profile is in fact due to ramp rates. Such a situation is not a generality and this illustration was only created to give one example of what can cause startup profiles.

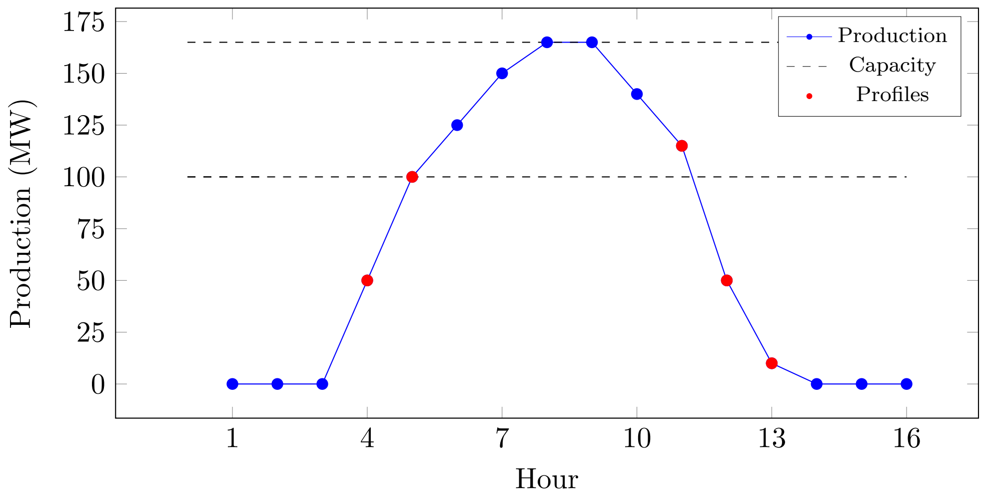


Figure 1.4: Example of a startup and shutdown profile

- Where does this fuel come from and how does it end up at the generator's gate?
- How much can we take?
- How much will we have to pay for it?

The answers to those three questions may be summed up in two words : forward contracts. Those contracts are characterized by 3 elements, each answering one of the above question :

- A delivery spot : the fuel supplier agrees to deliver its good at a certain location, or *spot*, located inside a pipeline network, meaning the fuel will then have to travel to the generators using this network. The network may easily be seen as a graph, where the different delivery spots are nodes and where contracts and generators respectively supply and take fuel inside/from this network, creating a flow. Note that in our model, to coincide with reality, we do not make the strong assumption that the network is "complete", that is we do not assume that fuel can be transported from every spot to every spot. We only consider a simple directed network, where fuel can sometimes go from a spot i to a spot j but not necessarily the other way around.

  An example of such a network is shown in Figure 1.5. In this Figure, generators 1, 2 and 3 can withdraw from all contracts even though generator 1 is on a different location than the other two, generator 4 can only withdraw from contract 4. Finally, for the sake of simplification, we suppose there is no delay between the moment fuel is injected inside the network and the moment it is taken off by generators.

- Off-take bounds : there may be a limit to how much fuel can be withdrawn, both from above and below. Note that a single contract can include multiple upper and lower bounds, depending on the timeframe considered. For example, a single contract can have hourly, daily, weekly, monthly and/or yearly upper and lower bounds;

- A price per unit : one major advantage with forward contracts is that they enable the fixation in advance of future prices. If a company has entered into a one-year forward

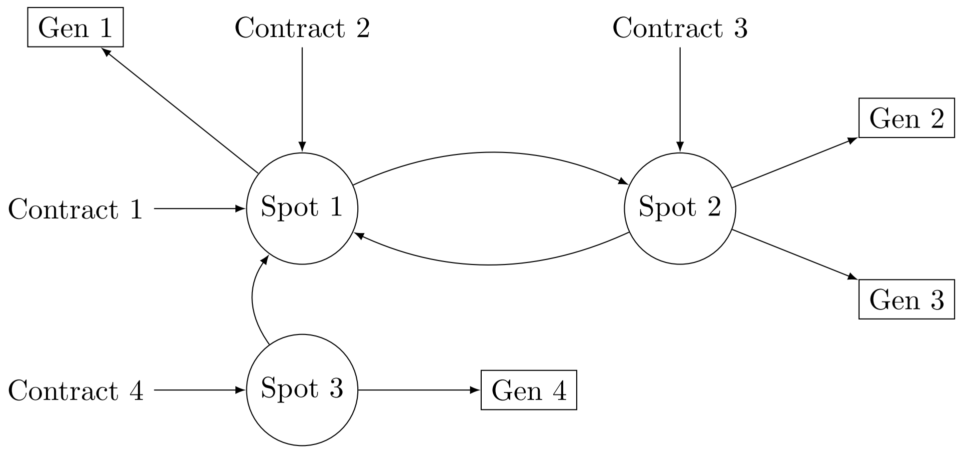


Figure 1.5: Example of a network where fuel can go both ways between Spot 1 and Spot 2 but only from Spot 3 to Spot 1.

contract with another party, it knows the price it will have to pay for withdrawing fuel at any time during the whole upcoming year.

Finally, note that the bounds sometimes force generators not to withdraw fuel from the most profitable contract (with the smallest cost) even if the upper bound on the off-take is not yet met because another contract with a higher price has to supply at least its minimal off-take value.

1.7 Ancillary services

To conclude with the description of the model scope, we move on to reserve contributions, or *ancillary services*. In the electricity market, the system operator requires generators to leave a production margin with both their minimum and maximum output so as to be able to respond to unforeseen events. There are three kinds of ancillary services possible :

1. Primary reserves consist of immediate small regulations (due to changes of frequency in the network) made in up to 30 seconds, or compensation for loss of production units;

2. Secondary reserves can be activated both upwards or downwards, meaning generators have to be prepared to produce more or less than what they were supposed to produce, to compensate for imbalances in the network within 15 minutes;

3. Tertiary reserves are reserves that may be supplied by generators that are inactive at a certain time but that are able to startup and increase their production fast enough to supply them. We distinguish two kinds of tertiary reserves : spinning reserves, supplied by active power plants, and non-spinning reserves, supplied by inactive ones. This type of reserves is able to kick in within an hour.

At this point, three things are worth mentioning :

- Because the dataset we received from GDF-Suez had zero-value requirements regarding primary reserves, our model will set these aside when built;

- Not every plant can contribute to all kinds of reserves. It in fact depends on its responsiveness but can also be constrained by reserve contribution limits;
- Ramp rates can influence the amount of reserves that generators can offer. This is due to the fact that the activation of reserves creates an increase (or a decrease) of production, causing ramp rates to bound this increase or decrease;

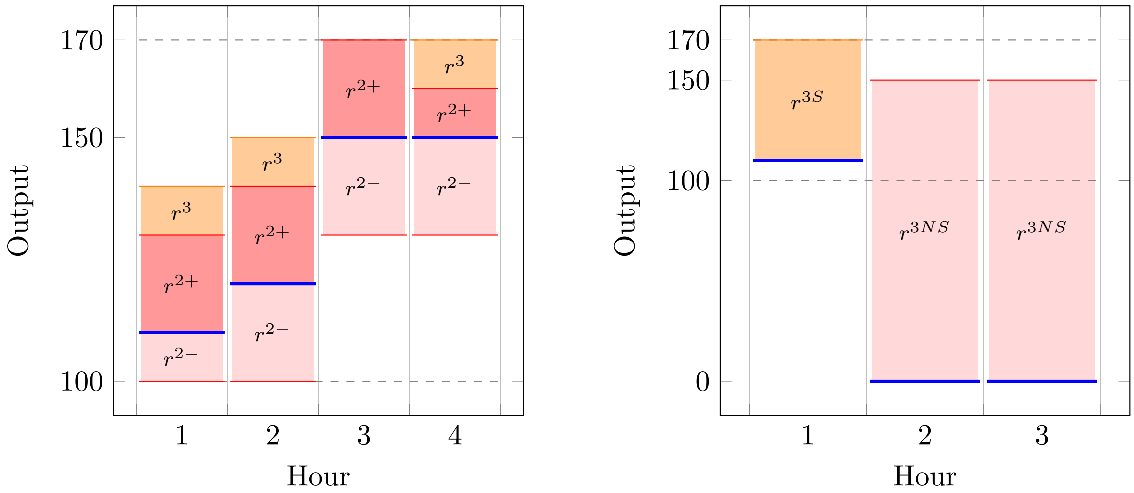
Examples 5, 6, Figures 1.6a and 1.6b show illustrations of generators and their contributions to ancillary services. More information can also be found in [Pap13a] and [Eli].

Example 5 *Let us consider a power plant with the following features :*

- *Maximum contribution to both upward and downward secondary reserves of* 20 MW*;*
- *Maximum contribution to tertiary reserves of* 10 MW*;*
- *Minimum and maximum capacity of* 100 *and* 170 MW*;*
- *Planned production for hours* 1, 2, 3 *and* 4 *of respectively* 110, 120, 150 *and* 150 MW*;*

Its contribution to downward secondary reserves is respectively, for hour 1, 2 *and* 3, *of* 10 MW *(more would violate the minimum capacity),* 20 MW *(more would violate both minimum capacity and maximum contribution to reserves) and* 20 MW *(more would violate maximum contribution to reserves).*
Its availability to contribution to upward secondary reserves is of 20 MW *for all four hours because there is always at least a* 20 MW *gap between maximum output and planned production. However, for hours* 3 *and* 4, *more than one situation can happen. Because this gap is less than* 30 MW *(sum of maximum contributions to upward secondary reserves and tertiary reserves), the generator has to make a choice on the reserve it will contribute to. For example, on hour* 3, *it could contribute* 20 MW *to upward secondary reserves (hence* 0 *to tertiary) and on hour* 4, 10 MW *to both. This example is shown in Figure 1.6a.*

Example 6 *Assume a generator unable to contribute to secondary reserves but with a* 150 MW *maximum contribution to tertiary reserves, with minimum and maximum capacity of* 100 *and* 170, *and with planned production for hour* 1, 2 *and* 3 *of respectively* 110, 0 *and* 0 *(the plant is shut down at time* 2*). Because it is capable to contribute at most* 150 MW *to tertiary reserves whether it is active or not, the effective contribution to tertiary reserves will respectively be of* 60 *(more would violate the maximum capacity),* 150 *and* 150 *for hours* 1, 2 *and* 3. *This example is shown in Figure 1.6b.*



(a) Example of capacities (grey dashed lines), production (blue line), downward secondary reserves contribution (light red area), upward secondary reserves contribution (dark red area) and tertiary reserves contribution (orange area) for a generator

(b) Example of contributions to tertiary reserves and distinction between spinning (orange area) and non-spinning (light red area)

Figure 1.6: Illustrations of reserve contributions from Examples 5 and 6

Chapter 2

Model definition

As stated in the introduction, private companies like GDF-Suez have models that aim at economically optimizing operations. However, this optimization cannot be done without first ensuring that some constraints are met, like satisfying demand, technical generator constraints, system requirements and so on. In this chapter, we translate the brief descriptions made in the previous chapter into their mathematical equivalent respecting the following scheme[2] :

- In section 2.1 we formulate the objective function as an energy company's profit, introducing the multiple revenue and expense sources the company faces in all stages of the production process;
- In section 2.2 we set the foundations of the whole model by defining all the binary variables needed to represent the dynamics of functioning, the startup and shutdown of plants in time, and by linking them all together;
- In section 2.3 we focus on the energy production process and the associated technical limitations;
- In section 2.4 we set aside production and focus on the generators' fuel supply network;
- Finally, in section 2.5 we focus on ancillary services requirements necessary for coping with unexpected events;

2.1 Objective function

To compute the company's profit, three kinds of revenue/expense sources are taken into account :

- Financial transactions : those cash flows occur when selling or buying electricity to networks to fulfill demand requirements, and when buying the fuel that will later be converted into electricity;
- Workforce expenses : whenever plants are on or when a plant is switching from one mode to another, workers have to be paid for maintenance or mode transition work;

[2]Although every parameter, set and variable will be introduced when necessary, a summary of notation is available in Appendix A.

- Ecological fines : the European Union Emission Trading Scheme highly regulates CO_2 emissions and companies have to pay whenever they produce energy using pollutant fuels;

Let T be the considered time horizon for our model[3], G the set of generators, C the set of forward contracts, M_g the modes for generator g, $M_g^0 \subset M_g$ the set of modes for generator g after removal of the Off mode, and A_g the possible transitions between those modes. Those three parts contribute to the definition of the following objective function over the whole horizon

$$
\begin{aligned}
\max \sum_{t\in T} \Bigg(& p_t^s \sum_{g\in G} s_{gt} - p_t^b b_t - \sum_{c\in C} p_{ct}^{\text{fuel}} o_{ct} \\
& - \sum_{g\in G} \Bigg(\sum_{m\in M_g^0} VOM_{gm} u_{gmt} + \sum_{a\in A_g} TC_{ga} v_{gat} \\
& \qquad + p_t^{CO_2} \sum_{m\in M_g} E_{gm} f_{gmt} \Bigg) \Bigg)
\end{aligned}
$$

The first line represents the financial transactions, where s_{gt}, b_t and o_{ct} are the variables that respectively correspond to the amount of power sold by generator g, the amount of power bought in the market and the off-take of fuel from contract c at time t, and $p_t^s, p_t^b, p_{ct}^{\text{fuel}}$ their associated prices.

The second line is the workforce expenses, u_{gmt} being a binary variable equal to 1 in case the plant is active and in mode m at time t, implying workers are to be paid a salary of VOM_{gm}, and v_{gat} a binary variable equal to 1 if mode transition a occurs at time t, associated to a cost TC_{ga}.

The third part covers the CO_2 emission costs, with E_{gm} being the fuel emission rate associated to generator g running in mode m, f_{gmt} the fuel consumption for this mode at time t and $p_t^{CO_2}$ the emission rights price.

2.2 Generator dynamics

Since our work aims at finding an optimal production dispatch, generator schedule and fuel consumption for a large time horizon, the dynamic aspect of our model must be dealt with great care to enable real-life application of the recommendations given by its optimal solution. In this section we use three main notations for the multiple new binary variables to be introduced : v for starting something up, z for shutting something down and u for having something active.

2.2.1 The basics

The first constraint associated to the dynamic aspect of a power plant is the transition from one time period $t-1$ to the next one t for a given generator g, which can be written

[3] In the rest of this thesis, we often use the set $T_j^k \subset T$, which corresponds to a sub-horizon of T, starting at the j−th hour and ending at time $(t_{end} - k)$, where t_{end} designates the last hour of T. Besides, if either subscript or superscript is missing in this notation, then the first (respectively last) time period considered is the first (respectively last) time period there is in T, meaning $T_j = \{t_j, t_{j+1}, \ldots, t_{end}\}$ and $T^k = \{t_1, t_2, \ldots, t_{end} - k\}$.

as

$$u_{gt} = u_{g,t-1} + v_{gt} - z_{gt} \qquad \forall g \in G, t \in T_2.$$

We also consider situations where the value of some of these variables is fixed in advance and there is no choice between having the generator active or not for a certain period. An example of such situations is generator maintenance, during which the generator has to be inactive. Denoting respectively $MR_g \subseteq T$ and $MS_g \subseteq T$ the time slots when a power plant g must run and when it must be off, we have

$$u_{gt} = 1, \qquad \forall g \in G, t \in MR_g \quad (2.1)$$

$$u_{gt} = 0, \qquad \forall g \in G, t \in MS_g \quad (2.2)$$

Note that we also have the same kind of constraints for modes. The new constraints in that case are just a copy of the above ones with an added index m to u_{gt}, MR_g, MS_g and an additional $\forall m \in M_g^0$ for the domain of definition.

Assuming the Off state for a generator is one of its modes, then at every instant t, each power plant g must be in exactly one of its modes, implying that

$$\sum_{m \in M_g} u_{gmt} = 1 \qquad \forall g \in G, t \in T$$

Finally, using this same assumption, whether a generator is On or not at time period t can be determined using

$$u_{gt} = \sum_{m \in M_g^0} u_{gmt} \qquad \forall g \in G, t \in T.$$

2.2.2 The transitions

Let A_g the set of all possible transitions from a specific mode to another for a given generator g, $a \in A_g$ a particular transition, and v_{gat} a binary variable that equals 1 if transition a occurs at time t. Let the two operators $F : A_g \to M_g$ and $T : A_g \to M_g$, such that a corresponds to the transition $(F(a), T(a))$. Then, at each time t, the mode transition is given by

$$u_{gmt} = u_{gm,t-1} + \sum_{\substack{a \in A_g \\ T(a)=m}} v_{gat} - \sum_{\substack{a \in A_g \\ F(a)=m}} v_{gat}, \qquad \forall g \in G, m \in M_g, t \in T_2$$

Also, as no more than one transition can occur per generator per instant, we have

$$\sum_{a \in A_g} v_{gat} \leq 1 \qquad \forall g \in G, t \in T$$

Finally, because the Off state is considered as a mode, whether or not a generator started up into a certain mode m can be found looking at transition (Off, m), which reads

$$v_{gmt} = v_{gat}, \qquad \forall g \in G, m \in M_g^0, a = (\text{Off}, m), t \in T$$

and the overall startup decision can be found summing, over all the possible transitions, each of those that begin at the Off state :

$$v_{gt} = \sum_{\substack{a \in A_g \\ F(a)=\text{Off}}} v_{gat} \qquad \forall g \in G, t \in T$$

2.2.3 Minimum Up/Down Times

As discussed in section 1.1, we cannot expect a generator to be able to be started up or shut down in alternance because there are some time limitations on how soon a generator is able to start up again once it is shut down and on how long it has to wait to be capable of being shut down once it has started up, called minimum up times and minimum down times. Denoting UT_g the minimum up time for generator g, that is the number of time periods must be kept working once it has been started up, and DT_g the minimum down time, we have the two following constraints

$$\sum_{y=t-UT_g+1}^{t} v_{gy} \leq u_{gt}, \qquad \forall g \in G, t \in T_{UT_g}$$

$$\sum_{y=t-DT_g+1}^{t} z_{gy} \leq 1 - u_{gt}, \qquad \forall g \in G, t \in T_{DT_g}$$

The same rationale applies to transitions between modes, in the sense that if a transition is made at time t to arrive in (respectively leave) state m, the generator has to be maintained in this new mode (respectively cannot return to the original mode) for at least ut_{gm} (respectively dt_{gm}). This reads

$$\sum_{y=t-ut_{gm}+1}^{t} \sum_{\substack{a \in A_g \\ T(a)=m}} v_{gay} \leq u_{gmt}, \qquad \forall g \in G, m \in M_g, t \in T_{ut_{gm}}$$

$$\sum_{y=t-dt_{gm}+1}^{t} \sum_{\substack{a \in A_g \\ F(a)=m}} v_{gay} \leq 1 - u_{gmt} \qquad \forall g \in G, m \in M_g, t \in T_{dt_{gm}}$$

2.2.4 Startup/Shutdown profiles

Before we introduce any new element, it is necessary to point out that startup profiles are assumed to be mode-dependent and temperature-dependent whereas shutting down profiles are only mode-dependent. This is explained by the fact that a generator will always be hot when it is shut down, hence there is only one possible temperature state.

Let G^{SUP} denote the set of generators that possess startup profiles, G^{SDP} those with shutdown profiles, M_g^{SUP} and M_g^{SDP} their respective modes with profiles (we here assume there could be generators for which certain modes have profiles and others do not). We introduce our to-be-needed binary variables u_{gmt}^{SUP}, u_{gmt}^{SDP} and u_{gmt}^{DISP} that are respectively equal to 1 if generator g is active, in working mode m and following a startup profile at time t, if it is following a shutdown profile, or if no profile is currently forcing its production level. When the generator is working under a certain mode, only one of those three states can be effective at the same time, hence we have

$$u_{gmt} = u_{gmt}^{SUP} + u_{gmt}^{DISP} + u_{gmt}^{SDP} \qquad \forall g \in G, m \in M_g^0, t \in T$$

We now need to mathematically define u_{gmt}^{SUP} and u_{gmt}^{SDP}. Suppose that for a startup profile associated to a generator's temperature state l, the profile's time length is τ_{gml}^{SUP} hour, then whether or not a generator g working in mode m is currently operating within a profile in temperature l is determined by looking at the previous τ_{gml}^{SUP} time periods and identifying whether the generator was started up in this mode during those hours. We can then remove the temperature-dependency by doing this for every temperature and summing over them, which gives us the following

$$u_{gmt}^{SUP} = \sum_{l \in \Theta_{gm}^{SUP}} \sum_{\substack{y=t-\tau_{gml}^{SUP}+1 \\ y \geq 1}}^{t} v_{gmly} \qquad \forall g \in G^{SUP}, m \in M_g^{SUP}, t \in T_{\tilde{\tau}_{gm}^{SUP}}$$

where Θ_{gm}^{SUP} designates the set of temperature-dependent startups that have a startup profile and $\tilde{\tau}_{gm}^{SUP} = \min_{l \in \Theta_{gm}^{SUP}} \tau_{gml}^{SUP}$.

The same rationale goes for shutdown profiles, except we do not have to consider temperature-dependency and that we now look in the future time periods for a shutdown instead of the previous ones for a startup. We therefore obtain

$$u_{gmt}^{SDP} = \sum_{y=t}^{t+\tau_{gm}^{SDP}-1} v_{gat} \qquad \forall g \in G^{SDP}, m \in M_g^{SDP}, a = (m, \text{Off}), t \in T^{\tau_{gm}^{SDP}+1}$$

with τ_{gm}^{SDP} the length of the shutdown profile for mode m of generator g[4].

2.2.5 Temperature-dependent startup

As mentioned in section 1.4, we consider that once shut down, power plants cool down with time and when started up later, the startup fuel consumption varies depending on how long the plant was shut down. First, as a startup always occurs in exactly one of the possible temperature states, we have

$$v_{gmt} = \sum_{l \in \Theta_{gm}} v_{gmlt}, \qquad g \in G, m \in M_g^0, t \in T$$

where Θ_{gm} is the set of possible temperature states and v_{gmlt} a binary variable equal to 1 if we startup generator g in mode m at instant t with temperature l.

Then, suppose that temperature l occurs between $\underline{\tau}_{gml}$ and $\overline{\tau}_{gml}$ after the shutdown. A startup at temperature l is possible if and only if the power plant has been shut down between $\underline{\tau}_{gml}$ and $\overline{\tau}_{gml}$ hours before, hence

$$v_{gmlt} \leq \sum_{y=t-\underline{\tau}_{gml}+1}^{t-\overline{\tau}_{gml}} z_{gy}, \qquad g \in G, m \in M_g^0, l \in \Theta_{gm}, t \in T_{\underline{\tau}_{gml}}.$$

2.3 Power production

Now that the general functioning of the plants has been established, let us focus on the production per say. One of the most important constraints of the whole model is the

[4]For every generator without startup and shutdown profiles or with only one of the two, this way of thinking is easily reproduced and the model takes that into account, which means u^{DISP}, u^{SDP} and u^{SUP} exist for all generators, not only for those with both profiles. A zero value is imposed to u^{SDP} and u^{SUP} for those generators, as well as the production p^{SUP} and p^{SDP} to be introduced in the next section. We do not rewrite those constraints for these similar cases in order to simplify the exposition.

market clearing constraint which expresses the balance between the amount of power that is produced, sold, bought and consumed inside the power grid. This is represented by

$$\sum_{g\in G} p_{gt} + b_t = D_t + \sum_{g\in G} s_{gt}, \qquad \forall t \in T \quad (2.3)$$

with p_{gt} the production of generator g at time t, b_t the amount of power bought at time t, D_t the demand and s_{gt} the amount of power sold by generator g. Moreover, the amount of power bought and/or sold both have to be nonnegative and respectively bounded by the demand and what is produced during that time period :

$$b_t \geq 0 \qquad \forall t \in T$$
$$b_t \leq D_t \qquad \forall t \in T$$

$$s_{gt} \geq 0 \qquad \forall g \in G, t \in T$$
$$s_{gt} \leq p_{gt} \qquad \forall g \in G, t \in T$$

The other constraints involving the production variables are of two types : capacity and ramping.

2.3.1 Segment and mode production

As mentioned earlier, a power plant can only be operated in exactly one mode at every instant t. We can therefore consider the production per mode instead of the production per generator using the following formula

$$p_{gt} = \sum_{m\in M_g^0} p_{gmt} \qquad \forall g \in G, t \in T$$

We can also cut this production per mode even further by only considering the production for each linear segment that composes the piecewise-linear curve. To do so, we introduce the production per slice variable p_{gmst}, which is linked to the production per mode by

$$p_{gmt} = \sum_{s\in S_{gm}} p_{gmst} \qquad \forall g \in G, m \in M_g^0, t \in T$$

Let us define u_{gmst} as a binary variable set to 1 if segment s is used at time t to produce power. Because this variable has to be active in order to enable the plant to produce power inside the segment and because every segment possesses its own maximum capacity P^+_{gms}, we have the minimum and maximum capacity per segment as follows

$$p_{gmst} \geq 0 \qquad \forall g \in G, m \in M_g^0, s \in S_{gm}, t \in T$$
$$p_{gmst} \leq P^+_{gms} u_{gmst} \qquad \forall g \in G, m \in M_g^0, s \in S_{gm}, t \in T$$

We also introduce two types of profile production variables, p^{SUP}_{gmt} and p^{SDP}_{gmt}, depending on whether the considered profile is one of startup or one of shutdown. Fixing the value for p^{SUP}_{gmt} (respectively p^{SDP}_{gmt}) to its corresponding production level within the profile uses the same rationale as in the definition of the u^{SUP}_{gmt} (respectively u^{SDP}_{gmt}) variables, except that when looking in the past for a startup occurrence (respectively in the future for a shutdown), we also include a production level factor that includes how far in the past (respectively future) we are looking. This whole reasoning is captured by

$$p_{gmt}^{SUP} = \sum_{l \in \Theta_{gm}^{SUP}} \sum_{\substack{y=t-\tau_{gml}^{SUP}+1 \\ y \geq 1}}^{t} v_{gmly} P_{gml,t-y+1}^{SUP} \qquad \forall g \in G^{SUP}, m \in M_g^{SUP}, t \in T_{\tilde{\tau}_{gm}^{SUP}}$$

$$p_{gmt}^{SDP} = \sum_{y=t+1}^{t+\tau_{gml}^{SDP}} v_{gat} P_{gm,y-t}^{SDP} \qquad \forall g \in G^{SDP}, m \in M_g^{SDP}, a = (m, \text{Off}), t \in T^{\tau_{gm}^{SDP}}$$

We now focus on minimum and maximum capacity for a given mode. The usual form for this kind of constraint is $p_{gmt} \leq P_{gm}^{+} u_{gmt}$ but the presence of profiles invalidates this form. Two things have to be captured by the constraints to be created : on the one hand the production has to be fixed to p_{gmt}^{SUP} or p_{gmt}^{SDP} in case the plant is currently following a profile, and on the other hand the classic minimum and maximum output constraints have to be the only limits if the generator is neither starting up nor shutting down. Those two aspects are respected if we write the following equations

$$p_{gmt} \geq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{-} u_{gmt}^{DISP} \qquad \forall g \in G, m \in M_g^0, t \in T \quad (2.4)$$

$$p_{gmt} \leq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{+} u_{gmt}^{DISP} \qquad \forall g \in G, m \in M_g^0, t \in T \quad (2.5)$$

because on one side if the power plant is working under a profile, p_{gmt}^{DISP} will be zero and only one of the two remaining terms will be nonzero, hence fixing p_{gmt}, and on the other side if no profile is active, the first two terms have to be zero, and we re-obtain the standard minimum/maximum capacity constraint.

2.3.2 From one segment to another

Since heat-rate curves are not necessarily increasing, some additional constraints have to be added to ensure those segments are activated in order, which would not necessarily be the case otherwise. For example, without these constraints, if the successive heat-rates were decreasing, a generator would have a tendency to first produce on the last slice, because its marginal fuel consumption is the smallest, would then produce on the one just before last, then on the one before, and so on until it reaches a slice for which the marginal fuel consumption becomes too high to make it profitable to produce power (this will also be illustrated in example 10). We would thus end up with a situation where the last slices are producing at full capacity and the first ones are inactive, which is unrealistic in practice.

The first constraint ensures that we can always start producing power on the first slice. This is done by activating this slice as soon as the mode is active, which gives

$$u_{gm,s_1,t} = u_{gmt} \qquad \forall g \in G, m \in M_g, t \in T$$

The other constraints only allow the generator to activate a certain slice if the previous segments have been activated and are producing at full capacity

$$u_{gm,s+1,t} \leq \frac{p_{gmst}}{P_{gms}^{+}} \qquad \forall g \in G, m \in M_g^0, s \in S_{gm} \backslash \{s_{gm}^{end}\}, t \in T$$

where s_{gm}^{end} designates the last slice for generator g in mode m.

2.3.3 Ramp up/down constraint

As briefly mentioned in Example 1 and 4, power plants produce as they desire within their range of production while having to ensure that some ramping constraints are respected, one for increasing production and one for decreasing it. In simplified problems, the mathematical form for these two equations is generally $-R_g^- \leq p_{gt} - p_{g,t-1} \leq R_g^+$ with R_g^+, R_g^- the values for those ramping parameters. With this model, we have a few additional terms to add in order to correct some issues that the introduction of profiles and multiple modes has created.

Let us first tackle the profile issue. Example 4 introduced a profile that respects ramping rate constraints. In fact, in that example, the startup profile was actually driven by ramp rates. In reality, the fact that profiles respect ramping rate constraints is not mandatory and we have to deactivate those constraints when the power plant is working under a profile. Let M a large number, if we add the term $+Mu_{gmt}^{SUP}$ to the ramp up constraint and $-Mu_{gmt}^{SDP}$ to the ramp down constraint, the upper and lower bounds for the difference between two successive productions will be pushed somewhere around $-M$ and $+M$ which, if M is a large enough number, will automatically render these constraints inactive when a profile is active. The only assumption made for this to be valid is that startup profiles have increasing production levels and shutdown profiles have decreasing ones. If we remove this assumption, we also have to add Mu_{gmt}^{SDP} to our ramp up constraint and $-Mu_{gmt}^{SUP}$ to the ramp down constraint.

We now take on the mode switching issue. The gap between production ranges for two distinct modes can sometimes be relatively high, and thus can create infeasibility when the generator changes mode, which is shown in Example 7.

Example 7 *Suppose that the ramp rate parameters R^+ and R^- are mode-dependent and not just generator-dependent and let us ignore profiles for now. If we try a constraint of the form*

$$-R_{gm}^- \leq p_{gmt} - p_{gm,t-1} \leq R_{gm}^+,$$

a problem arises when we leave or enter a mode. If we enter mode m at time t, we have $p_{gm,t-1} = 0$, because it was inactive at time $t-1$, and the constraint for the mode we enter becomes

$$-R_{gm}^- \leq p_{gmt} \leq R_{gm}^+$$

The left inequality is always verified because the left-hand-side is always nonpositive and the right-hand-side nonnegative, which means we are left with $p_{gmt} \leq R_{gm}^+$. This means that if the ramp up parameter is not at least P_{gm}^-, we are never able to enter mode m.

To be as general as possible and enable this kind of unauthorized transition, we have to add a term that deactivates ramp rates whenever we enter or leave a mode in the same way we did for profiles.

To conclude this section, the constraints can be written as

$$p_{gmt} - p_{gm,t-1} \leq R_{gm}^+ + M \sum_{\substack{a \in A_g \\ T(a)=m}} v_{gat} + Mu_{gmt}^{SUP} \qquad \forall g \in G, m \in M_g^0, t \in T_1$$

$$p_{gm,t-1} - p_{gmt} \leq R_{gm}^- + M \sum_{\substack{a \in A_g \\ F(a)=m}} v_{gat} + Mu_{gmt}^{SDP} \qquad \forall g \in G, m \in M_g^0, t \in T_1$$

2.4 Fuel origin and transformation into power

2.4.1 From the contracts to the generators

We mentioned earlier in section 1.6 that the fuel consumed by generators in order to produce power comes from multiple contracts and is delivered to specific delivery points. To model the fuel network and off-takes, a few additional notations are in order. Let L be the links between fuel contracts and their corresponding delivery spot, PL the set of pipelines that link delivery spots together, FD the set of those spots and $TF(c)$ the set of timeframes over which minimum and/or maximum off-take constraints exist for a contract c. We also introduce the following four operators :

- $F^*(l) : L \to C$ that returns the contract of link l, and $T^*(l) : L \to FD$ that returns its delivery spot;

- $F^{**}(h) : PL \to FD$ and $T^{**}(h) : PL \to FD$ that respectively designate the origin and destination delivery spot from pipeline h.

Finally, we introduce the following four variables : o_{ct}, the amount of fuel withdrawn from contract c at time t, q_{lt} the quantity that is flowing through line $l \in L$ (meaning from contract $F^*(l)$ to the corresponding delivery spot), $\tilde{q}_{ht}$ the quantity flowing through pipeline h (meaning from spot $F^*(h)$ to spot $T^*(h)$) and f_{gt} the fuel consumption of generator g at hour t.

The first two equations to be introduced impose balance in the network. The first one starts from the point of view of a contract and distributes the amount of fuel withdrawn from this contract into the appropriate lines

$$o_{ct} = \sum_{\substack{l \in L \\ F^*(l)=c}} q_{lt} \qquad \forall c \in C, t \in T$$

Note that this constraint does not impose that a contract be linked to one and only one delivery spot, because there could be contracts that are able to deliver their fuel at multiple locations. The second type of constraints takes a delivery spot point of view and ensures that what arrives at this spot either immediately goes to a generator located there or goes to another spot using pipelines. This gives us

$$\sum_{\substack{g \in G \\ g \in \mathrm{Pool}(d)}} f_{gt} + \sum_{\substack{h \in PL \\ F^{**}(l)=d}} \tilde{q}_{ht} = \sum_{\substack{l \in L \\ T^*(l)=d}} q_{lt} + \sum_{\substack{h \in PL \\ T^{**}(l)=d}} \tilde{q}_{ht} \qquad \forall d \in FD, t \in T \quad (2.6)$$

with $\mathrm{Pool}(\cdot)$ a spot-dependent subset of G that only contains generators located at this spot. Note that the model does not impose any capacity limit on the flow going through the various pipelines. This could easily be done by bounding variable $\tilde{q}_{ht}$ from above by the maximum capacity of pipeline h.

To finish up with the network part, we impose the maximum and minimum limitations on off-takes $\overline{O}$ and $\underline{O}$ for the different timeframes, which are written as

$$\underline{O}_{c\tau} \leq \sum_{y \in \tau} o_{cy} \leq \overline{O}_{c\tau}, \qquad \forall c \in C, \forall \tau \in TF(c).$$

2.4.2 From fuel to power

Now that a way for fuel to go from contracts to the generators has been created, let us focus on its transformation into power. In order to do that, we consider 4 kinds of fuel consumption at each time slot :

1. Startup consumption : as discussed in section 2.2.5, we try to take into account different startup consumptions depending on the temperature state l of the generator. Using the binary v_{gmlt} introduced in the same section and letting SUC_{gml} be the fuel consumption for a startup done in temperature state l, we have

$$\sum_{l\in\Theta_{gm}} v_{gmlt}SUC_{gml}$$

   the amount of fuel consumed when starting up the plant. Note that only one of the terms in this sum can be nonzero at a time since the generator's temperature is unique at a given time. Note also that this consumption is completely separated from the possible presence of profiles. The fuel consumed by the production imposed in a profile will be included in the production consumption paragraph to come;

2. No-load consumption : no-load consumption is a predetermined quantity of fuel that is used each time a power plant is running but not producing power. This can thus be interpreted as a fixed cost, where the cost is expressed in fuel units, to be paid as a "right to produce power". Using the binary variable u_{gmt} defined earlier and letting NLC_{gm} be the no-load consumption for plant g in mode m, we have

$$NLC_{gm}u_{gmt}$$

3. Transition consumption : the transition consumption TC_{ga} associated to generator g and transition $a = (m_1, m_2)$ is the fuel that is used whenever a generator leaves mode m_1 to enter mode m_2. We include this consumption in the destination mode equation (meaning we consider it is a consumption linked to entering a new mode instead of a consumption linked to leaving it), which means we have the following consumption for mode m

$$\sum_{\substack{a\in A_g\\ T(a)=m_2}} TC_{ga}v_{gat},$$

4. Production consumption : each slice of our piecewise linear fuel consumption in section 1.3 has its own specific heat-rate value. To obtain the fuel consumption corresponding to power production alone, we sum the fuel consumption for each slice, equal to the amount of power production in the slice times its associated heat rate, for a total value of

$$\sum_{s\in S_{gm}} HR_{gms}p_{gmst}.$$

Summing all those terms, we get the total fuel per mode consumption

$$f_{gmt} = \sum_{l\in\Theta_{gm}} v_{gmlt}SUC_{gml} + NLC_{gm}u_{gmt} + \sum_{\substack{a\in A_g\\ T(a)=m}} TC_{ga}v_{gat} + \sum_{s\in S_{gm}} HR_{gms}p_{gmst} \qquad \forall g\in G, m\in M_g^0, t\in T.$$

Besides, since a generator g can only be in one mode at any given time t, the total fuel per generator can be found by summing over the different modes

$$f_{gt} = \sum_{m \in M_g^0} f_{gmt} \qquad \forall g \in G, t \in T$$

2.5 Ancillary Services

Last but not least, we tackle the subject of ancillary services, which adds reserve requirements to the system. As we mentioned in section 1.7, we only consider secondary and tertiary reserve requirements, and set aside primary reserves. In addition, we do not bear in mind that upward secondary reserves can be used as tertiary spinning reserves because they can be activated more rapidly. Note that a large proportion of this section is a simplification of and based on [SBB10].

2.5.1 Secondary reserves

Before thinking about the new constraints per say, it is necessary to point out that ancillary services will influence the model's production part by influencing minimum and maximum energy output. More sophisticated models could also be built that also make ancillary services interact with ramping constraints, but the general philosophy is that **if** ancillary services are to be used in critical grid conditions, ramping limits may be violated at the expense of mechanical stress on generators.

Secondary reserves are used to ensure that plants can reduce or increase their production by a certain amount without facing technical limitations, hence we divide the constraint into two categories, one for production increase (with r^{2+}_{gmt} the contribution to upward regulation) and one for production decrease (with r^{2-}_{gmt}). First, we impose that any plant contribution to reserves be nonnegative

$$r^{2+}_{gmt} \geq 0 \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$r^{2-}_{gmt} \geq 0 \qquad \forall g \in G, m \in M_g^0, t \in T$$

Then, we also force the sum of effective production and contribution to reserves to stay within the production range of the power plant. This is done reusing equations (2.4) and (2.5), with one additional term for each of them, the upward reserve contribution for maximum capacity, and the downward for minimum capacity. This gives us

$$p_{gmt} + r^{2+}_{gmt} \leq p^{SUP}_{gmt} + p^{SDP}_{gmt} + P^{+}_{gm} u^{DISP}_{gmt} \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$p_{gmt} - r^{2-}_{gmt} \geq p^{SUP}_{gmt} + p^{SDP}_{gmt} + P^{-}_{gm} u^{DISP}_{gmt} \qquad \forall g \in G, m \in M_g^0, t \in T$$

Aside from the reserve requirements per say, we introduce one last bound on reserve contribution, which is a simple upper bound inherent to the generator of the form

$$r^{2+}_{gmt} \leq MCR^{2+}_{gmt} u^{DISP}_{gmt} \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$r^{2-}_{gmt} \leq MCR^{2-}_{gmt} u^{DISP}_{gmt} \qquad \forall g \in G, m \in M_g^0, t \in T$$

where $MCR^{2\pm}_{gmt}$ is the maximum contribution to secondary reserves by generator g working in mode m[5].

[5] In the last four equations, we use binary variable u^{DISP}_{gmt} instead of u_{gmt} because a generator cannot contribute to reserve if it is working under a profile.

To finish up with secondary reserves, the contributions of all generators have to cover the requirements, which means

$$\sum_{g\in G}\sum_{m\in M_g^0} r_{gmt}^{2+} \geq RR_t^{2+} \qquad \forall t \in T \quad (2.7)$$

$$\sum_{g\in G}\sum_{m\in M_g^0} r_{gmt}^{2-} \geq RR_t^{2-} \qquad \forall t \in T \quad (2.8)$$

with $RR_t^{2\pm}$ the requirements for upward/downward secondary reserves for time period t.

2.5.2 Tertiary reserves

For tertiary reserves, we also distinguish two categories : contribution to spinning reserves (from plants that are active) and non-spinning (plants that are off). The rationale for the former is very close to that of upward secondary reserves, and the first two constraints read

$$r_{gmt}^{3S} \geq 0 \qquad \forall g \in G, m \in M_g, t \in T$$

$$p_{gmt} + r_{gmt}^{2+} + r_{gmt}^{3S} \leq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{+} u_{gmt}^{DISP} \qquad \forall g \in G, m \in M_g^0, t \in T$$

where r_{gmt}^{3S} designates the contribution to tertiary spinning reserves. For non-spinning reserves, the representation is more complex. Because plants have to be off to contribute to tertiary non-spinning reserves, we introduce a new binary variable u_{gmt}^{3NS} that is equal to 1 if mode m from generator g contributes to tertiary non-spinning reserves. We have

$$u_{gmt}^{3NS} \leq 1 - u_{gt} \qquad \forall g \in G, m \in M_g^0, t \in T$$

Note that this constraint prevents contribution to both spinning and non-spinning reserves because if it contributes to non-spinning, then u^{3NS} is 1, which means u and r^{3S} are 0. Besides, at most 1 mode can contribute to non-spinning reserves because a plant cannot be started up in two distinct modes at the same time to contribute to reserves, which is captured by

$$\sum_{m\in M_g} u_{gmt}^{3NS} \leq 1 \qquad \forall g \in G, t \in T$$

In addition, generators forced to an Off state by the must-stop equations (2.2) cannot be switched on to supply reserves. This implies that

$$u_{gmt}^{3NS} = 0 \qquad \forall g \in G, m \in M_g^0, t \in MS_g$$

and its equivalent for the set MS_{gm}. The nonnegative constraint remains present for non-spinning reserves

$$r_{gmt}^{3NS} \geq 0 \qquad \forall g \in G, m \in M_g^0, t \in T$$

and the upper bound on this type of reserves for the considered mode now becomes only the maximum capacity because there is no conflict with production, as opposed to the case of spinning reserves

$$r_{gmt}^{3NS} \leq P_{gm}^{+} u_{gmt}^{3NS} \qquad \forall g \in G, m \in M_g^0, t \in T$$

We also have an upper bound on reserve contributions, which is assumed the same for both spinning and non-spinning reserves. Because of this and because a plant cannot contribute to spinning and non-spinning reserves at the same time, we have

$$r^{3S}_{gmt} + r^{3NS}_{gmt} \leq MCR^{3}_{gmt} \qquad \forall g \in G, m \in M^{0}_{g}, t \in T, \quad (2.9)$$

which could be replaced by two sets of constraints, bounding each of the two contributions individually[6].

Finally, the last set of equations indicates the requirements tertiary reserves, which is written as

$$\sum_{g \in G} \sum_{m \in M^{0}_{g}} (r^{3S}_{gmt} + r^{3NS}_{gmt}) \geq RR^{3}_{t} \qquad \forall t \in T \quad (2.10)$$

In conclusion to this chapter, note that the full version of the model can be found in Appendix B.

2.6 Resulting model

Now that the model has been described, we briefly address in this section the size of our problem instance using data we received from GDF-Suez. For confidentiality reasons, we here present the size of our problem only and do not give the actual values of the parameters:

- 7344 hours;
- 18 generators, 2 of them with 2 possible working modes (an economy mode and a regular mode in both cases) and the others with only one. Both economy modes have one production slice only, with fixed production. One of the generators has to run during the whole horizon with a fixed production. All the other generators (15) have 5 or 6 production slices.
- 10 of the 18 generators have only one possible temperature state, the other 8 have 7.
- 7 generators have startup profiles and shutdown profiles, all of them with a one-hour duration.
- Zero demand in the power network, implying that every MW produced is sold to the network at price p^{s}_{t}, and no power is bought.
- The fuel network contains 6 contracts, 3 delivery spots and only one of those contracts has minimum/maximum off-take limits, on an hourly, daily and yearly timeframe.

Using AMPL, after pre-solve has eliminated some constraints and variables, the reduced problem has a total of 4 287 922 variables, 2 461 417 of them binary, and 5 557 822 constraints.

[6] This replacement would be completely equivalent because at most one of the two reserve contributions can be nonzero.

Chapter 3

Model solving

We now move on to the next objective of this thesis, which is to obtain a solution deemed good enough in an acceptable amount of time. For energy models, the computational operational time limit is generally set to an hour. To this effect, we tried to run CPLEX on the model with data mentioned in section 2.6, with an at least 1% optimality guarantee required to terminate[7]. No solution was returned after an hour of CPLEX running. After seven hours, the computer crashed due to the lack of available RAM[8,9] with still no feasible solution returned but with a sequence of upper bounds on our objective function.

In the following sections, we attempt to find a way to achieve two objectives : find a feasible solution fast enough, which we would like to be as close as possible to the optimum, and find an upper bound on our primal problem (which is a maximization), hopefully as small as possible so as to get some optimality guarantee on this found feasible solution. Our method is based on Lagrangian relaxation and the chapter explaining the approach is developed using the following structure :

- Section 3.1 is devoted to the theory behind Lagrangian relaxation and expands 3 ways to iteratively get upper bounds on our problem;
- Section 3.2 shows how we apply this theory to our optimization problem;
- Section 3.3 focuses on how to obtain feasible solutions;
- Sections 3.4 and 3.5 presents one of the drawbacks we may get when switching from theory to practice and state our algorithm using all the previous sections;

3.1 Lagrangian relaxation, duality and bounds

3.1.1 Lagrangian relaxation

To get upper bounds on the problem through Lagrangian relaxation, we relax some of the constraints and add a penalty when violated and a reward when satisfied, in order to get a new problem that is easier to solve[10]. Suppose we have the following maximization

[7] Computer specifications : Macbook Pro : 2.9Ghz Intel Core i7 Processor, 8Go 1600 MHz DDR3 RAM

[8] CPLEX was the only program running when the test was launched, so no other software was taking a share of the memory

[9] A solution would have been to fix a limit to the RAM available for CPLEX to use and that any additional memory use be done on the hard drive memory but this would make the solver progress much slower than if it used only RAM

[10] A lot of this section is taken from [Sä04], [Lem00] and [Boy12].

problem

$$\begin{array}{ll} \max & f(x) \\ & \left.\begin{array}{l} x \in X \\ h_i(x) = 0, i = 1, \ldots, m \quad (\lambda_i) \\ g_j(x) \leq 0, j = 1, \ldots, n \quad (\mu_j) \end{array}\right\} x \in X^* \end{array} \tag{3.1}$$

where X is a set and λ_i, μ_j denote the dual multipliers of the h_i and g_j constraints. We form the Lagrangian function $L : X \times \mathbb{R}^m \times \mathbb{R}^n_+ \to \mathbb{R}$

$$(x, \lambda, \mu) \mapsto L(x, \lambda, \mu) = f(x) - \sum_{i=1}^{m} \lambda_i h_i(x) - \sum_{j=1}^{n} \mu_j g_j(x) \tag{3.2}$$

and the dual function θ that depends only on the dual multipliers $\theta : \mathbb{R}^m \times \mathbb{R}^n_+ \to \mathbb{R}$ as

$$(\lambda, \mu) \mapsto \theta(\lambda, \mu) = \max_{x \in X} L(x, \lambda, \mu). \tag{3.3}$$

The dual problem consists in minimizing this dual function over its arguments, or

$$\min_{\lambda,\mu} \theta(\lambda, \mu), \qquad \lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^n_+ \tag{3.4}$$

We observe the weak duality

$$\theta(\lambda, \mu) \geq f(x) \qquad \forall x \in X, (\lambda, \mu) \in \mathbb{R}^m \times \mathbb{R}^n_+$$

and we also have the following theorem (see [Lem00] Theorem 10 for the rationale behind the proof)

Theorem 8 *Whatever the data X, f, h_i, g_j in (3.1) can be, the function θ is always convex and lower semi-continuous.*

For mixed integer linear programming problems (like the unit commitment problem), the property of strong duality (meaning $\exists (x, \lambda, \mu) \in X^* \times \mathbb{R}^m \times \mathbb{R}^n_+$ such that $f(x) = \theta(\lambda, \mu)$) does not hold in general.

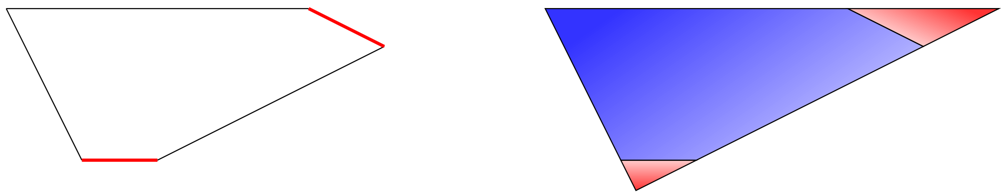
Besides, if λ, μ are such that (3.3) has an optimal solution $x_{\lambda,\mu}$ (not necessarily unique), then

$$\begin{pmatrix} \sigma_\lambda \\ \sigma_\mu \end{pmatrix} = \begin{pmatrix} -h(x_{\lambda,\mu}) \\ -g(x_{\lambda,\mu}) \end{pmatrix} \tag{3.5}$$

is a subgradient of θ at (λ, μ), i.e.

$$\theta(\tilde{\lambda}, \tilde{\mu}) \geq \theta(\lambda, \mu) + \sigma_\lambda^T (\tilde{\lambda} - \lambda) + \sigma_\mu^T (\tilde{\mu} - \mu) \qquad \forall (\tilde{\lambda}, \tilde{\mu}) \in \mathbb{R}^m \times \mathbb{R}^n_+ \tag{3.6}$$

Now that a subgradient of our problem has been formally derived, we can start thinking about solving (3.4) using three possible iterative methods : subgradient methods, cutting-plane methods or a mix between the two, bundle methods.



(a) Original feasible set   (b) New feasible set after relaxation

Figure 3.1: Feasible set before and after Lagrangian Relaxation. In the left figure, the red lines are the constraints that will be relaxed. In the right figure, a blue color means a reward in the objective function (the darker the more rewarding) and a red color corresponds to a penalty.

3.1.2 Subgradient methods

The subgradient method is a generalization of the gradient method. At each iteration, we extract a dual problem subgradient, in our case the one given in equation (3.5), then choose a step-size α_k and update the dual multipliers in the subgradient opposite direction.

$$\lambda_i^{(k+1)} \leftarrow \lambda_i^{(k)} - \alpha_k \sigma_{\lambda,i}^{(k)} \qquad i = 1, \ldots, m \quad (3.7)$$

$$\mu_j^{(k+1)} \leftarrow \mu_j^{(k)} - \alpha_k \sigma_{\mu,j}^{(k)} \qquad j = 1, \ldots, n \quad (3.8)$$

A problem with this simple update rule is that we have no guarantee over whether or not the μ_j's will be nonnegative, which is mandatory for our problem. One simple fix for this issue is to then project these multipliers on a feasible set, which is known as the projected subgradient method. An example of such a projection is the projection of the $\mu_j^{(k+1)}$ on the nonnegative orthant in the following way

$$\mu_j^{(k+1)} \leftarrow \left(\mu_j^{(k+1)}\right)^+, \qquad (3.9)$$

where $(\cdot)^+$ denotes the positive part operator, that is $(\cdot)^+ = \max(\cdot, 0)$. That being said, the opposite to the subgradient direction has no guarantee whatsoever to be a descent direction, meaning we have to keep track of our progress. The generic algorithm without a specific stopping criterion writes

Input: A sequence of steps α_k;
Initial dual multipliers $(\lambda^{(0)}, \mu^{(0)})$;
Maximum number of iterations $k_{\max}$;
for $k = 0$ **to** $k_{\max}$ **do**
  Solve *Relaxed Problem*
    Extract subgradient $(\sigma_\lambda^{(k)}, \sigma_\mu^{(k)})$;
    Extract optimal objective value f_k^*;
  Update dual multipliers via equations (3.7), (3.8) and (3.9);
end
return $\min_k f_k^*$;

Algorithm 1: A general projected subgradient algorithm

Step-size choosing

In the above, we said a sequence of step-sizes α_k is required, but never provided any ways to obtain such a sequence. If no information is known about the optimal value to the dual problem, classic updating rules include constant step-size ($\alpha_k = \alpha$), constant step length ($\alpha_k = \gamma/||(\sigma_\lambda^{(k)}, \sigma_\mu^{(k)})||_2$) or some decreasing step-size rules, each of them having its own convergence results.

If we possess some information on the optimal objective function as in our case, which we will show in the following sections, more sophisticated step-sizes can be computed. Assume we know the exact value of the dual problem θ^*, its corresponding optimal argument being $x^* = (\lambda^*, \mu^*)$, and that we currently have a point $x^{(k)} = (\lambda^{(k)}, \mu^{(k)})$, with associated subgradient $g^{(k)}$. The following inequality holds

$$\begin{aligned} ||x^{(k+1)} - x^*||_2^2 &\leq ||x^{(k)} - \alpha_k g^{(k)} - x^*||_2^2 \\ &= ||x^{(k)} - x^*||_2^2 - 2\alpha_k (g^{(k)})^T (x^{(k)} - x^*) + \alpha_k^2 ||g^{(k)}||_2^2 \end{aligned}$$

where the inequality is due to the projection into the nonnegative orthant of the dual multipliers associated to inequality constraints, and the equality is a simple expansion of the $||\cdot||_2$ norm. Then, because $g^{(k)}$ is a subgradient, this next inequality also holds

$$\theta^{(k)} - \theta^* \geq (g^{(k)})^T (x^{(k)} - x^*).$$

Hence, when combining those two inequalities, we obtain

$$||x^{(k+1)} - x^*||_2^2 \leq ||x^{(k)} - x^*||_2^2 - 2\alpha_k(\theta^{(k)} - \theta^*) + \alpha_k^2 ||g^{(k)}||_2^2$$

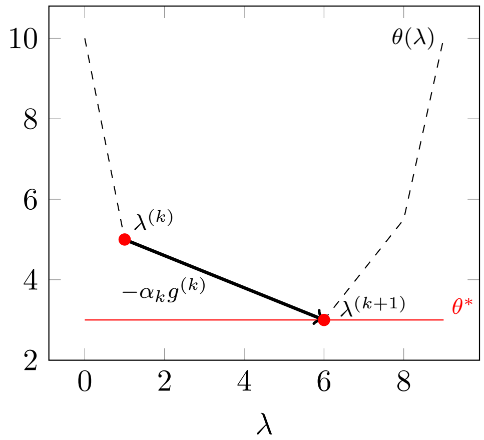
Because we want our next iterate to be as close as possible to optimum, we minimize the right-hand size by differentiating with respect to α_k, which gives us an optimal value for this step of

$$\alpha_k = \frac{\theta^{(k)} - \theta^*}{||g^{(k)}||_2^2}$$

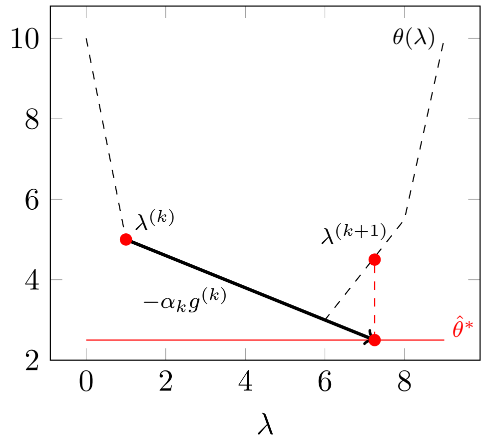
This method of update is known as the Polyak rule and the geometrical interpretation for this rule is the assumption that the theoretical sub-evaluation realized by the subgradient is actually a correct evaluation, meaning the optimal step can be found going down the subgradient hyperplane until we hit θ^*. The problem with this rule is that the optimal objective function is seldom known and the subgradient hyperplane does not necessarily coincide with the descent direction (4 different possibilities of applying the Polyak rule are displayed on Figure 3.2). However, it still can be adapted (more information can be found in [Ned08]) to fit our needs to a certain extent, by using a modified Polyak's step size rule, where we use the best approximation currently at our disposal $\hat{\theta}^*$ of the optimal value θ^* instead of θ^* itself, which we do not know, and where we also multiply this theoretical step by a damping factor $0 < \gamma \leq 1$.

$$\alpha_k = \gamma \frac{\theta^{(k)} - \hat{\theta}^*}{||g^{(k)}||_2^2} \qquad 0 < \gamma \leq 1$$

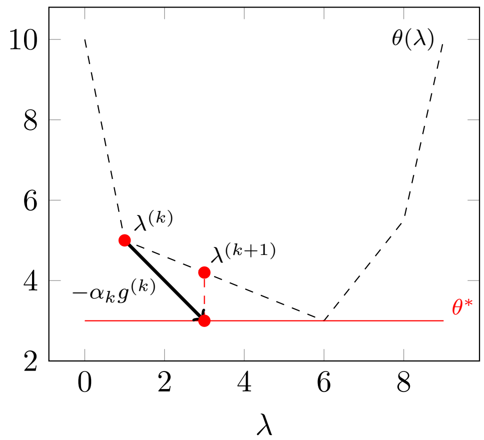
The results section of this thesis will compare how the subgradient algorithm behaves for different values of γ in our model.



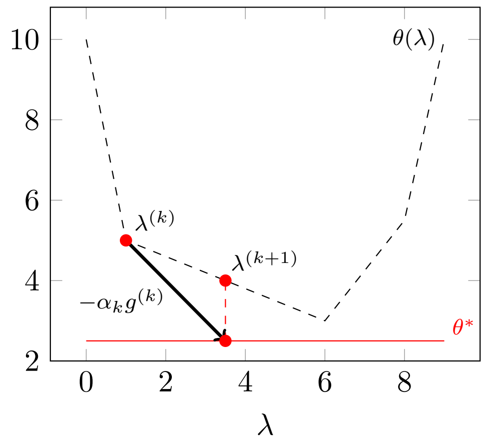
(a) Ideal situation (subgradient at $\lambda^{(k)}$ correctly evaluates $\theta(\lambda)$, and knowledge of optimal value θ^*) $\Rightarrow \lambda^{(k+1)}$ lands on λ^*



(b) Subgradient correctly evaluates $\theta(\lambda)$ but only approximation $\hat{\theta}^*$ of $\theta^* \Rightarrow \lambda^{(k+1)}$ goes past λ^* (if sub-approximation)



(c) Subgradient only provides a sub-evaluation of $\theta(\lambda)$ but perfect knowledge of optimal value $\theta^* \rightarrow \lambda^{(k+1)}$ does not land far enough



(d) Our usual situation : subgradient gives a sub-evaluation of $\theta(\lambda)$ and only a sub-approximation of θ^* is available

Figure 3.2: Application of the Polyak step rule to 4 typical cases of dual problems when only one dual multiplier is involved.

3.1.3 Cutting-plane method

The idea behind the cutting-plane method entirely relies on equation (3.6). Suppose we have K points $(\lambda^{(k)}, \mu^{(k)})$ $(k = 1, \dots, K)$, their dual function value $\theta(\lambda^{(k)}, \mu^{(k)})$ and a corresponding subgradient $(\sigma_\lambda^{(k)}, \sigma_\mu^{(k)})$ for each of these points. Using this information, we try to minimize function θ using all sub-estimations at our disposal. This is equivalent to the problem

$$\begin{aligned} \min_{r,\lambda,\mu} \quad & r \\ s.t. \quad & r \geq \theta(\lambda^{(k)}, \mu^{(k)}) + \left(\sigma_\lambda^{(k)}\right)^T \left(\lambda - \lambda^{(k)}\right) + \left(\sigma_\mu^{(k)}\right)^T \left(\mu - \mu^{(k)}\right) \qquad k = 1, \dots, K \\ & \mu \geq 0 \end{aligned} \tag{3.10}$$

With the solution (λ, μ) as our next iterate, we find a new dual function value and a new subgradient, and re-solve the above problem with $K := K + 1$. The sequence r_{K+1} is nondecreasing, because every new plane adds a lower bound to the problem, and is bounded from above by $\theta(\lambda^{(k)}, \mu^{(k)})$ for every $k \in \{1, \dots, K\}$ and we have

$$\liminf_{K\to\infty} \theta(\lambda^{(K)}, \mu^{(K)}) = \limsup_{K\to\infty} r_{K+1} = \min_{\lambda,\mu} \theta(\lambda, \mu)$$

In this expression, even though the sequence of r_{K+1} is nondecreasing, the sequence of values for $\theta(\lambda^{(k)}, \mu^{(k)})$ has no such pattern and is in general oscillating. The main problem with this method in fact occurs at the beginning of the algorithm, when K is small, because only a few points and subgradients are available, compared to a great number of variables (the dual multipliers of the relaxed constraints), which often renders the problem unbounded. To counter this unboundedness, two options can be explored:

- Find a feasible point to the original problem : weak duality states that this feasible primal point is a lower bound on r at every iteration and could therefore be used. But this option does not fix another issue that arises when too large a number of dual multipliers is involved. In this case, the efficiency of this method is diminished because when too few cutting-planes are available, the iterates tend to oscillate among the boundaries of the feasible space, which implies a large number of oscillations when the dimension of the feasible space is large;

- Bundle methods : these methods create an optimization problem for new dual multipliers that has a lower bound even when only one cutting plane is available and that prevents large movements in only one direction but instead encourages multiple small variations for all multipliers;

3.1.4 Bundle Methods

Bundle methods differentiate themselves from cutting-plane methods in the sense that they include a stability center point and penalize any deviation from this point.

Suppose we start from a point $(\overline{\lambda}, \overline{\mu}) \in \mathbb{R}^m \times \mathbb{R}^n_+$, which we deem to be a relatively good estimator of the optimal values for the dual optimization problem (3.4). We therefore also assume that $\theta(\overline{\lambda}, \overline{\mu})$ is relatively close to its optimal value. This implies that when solving problem (3.10), we would like to stay close to $(\overline{\lambda}, \overline{\mu})$. To do so, we add a positive term that penalizes any deviation from this point. Reusing problem (3.10), a new possible objective function to minimize could be

$$r + (||\lambda - \overline{\lambda}||_2^2 + ||\mu - \overline{\mu}||_2^2).$$

Using this new objective function instead of r prevents unboundedness for the problem. Indeed, imagine that only one cutting-plane is at our disposal, then the new dual multipliers would try to minimize r by going down along this plane, but doing so would increase the second term, until we arrive at an equilibrium between $r-$minimization and stability-center remoteness. Even though this improves the simple cutting-plane method, there is still one issue to be solved, which is scaling. Although we penalize any deviation from the stability center, we do not take into account how well we estimate this point compared to the optimal value, which is why we also introduce a scaling factor t. The objective now becomes

$$r + \frac{1}{2t}||\lambda - \overline{\lambda}||_2^2 + \frac{1}{2t}||\mu - \overline{\mu}||_2^2. \tag{3.11}$$

When we believe that our estimator $(\overline{\lambda}, \overline{\mu})$ is close to the optimal value (λ^*, μ^*), we use a small t value, which makes the quadratic term predominant and forces a new (λ, μ) close to the stability center. Different update rules can be used for this scaling factor, such as

$$t_{K+1} = t_K \frac{\theta(\overline{\lambda}, \overline{\mu}) - r^{(K+1)}}{2(\theta(\lambda^{(K+1)}, \mu^{(K+1)}) - r^{(K+1)})}, \tag{3.12}$$

or a constant t can be used, both being heuristics presented in [Sä04]. The bundle method problem at iteration K thus reads

$$\begin{aligned} \min_{r,\lambda,\mu} \quad & r + \frac{1}{2t_K}||\lambda - \overline{\lambda}||_2^2 + \frac{1}{2t_K}||\mu - \overline{\mu}||_2^2 \\ s.t. \quad & r \geq \theta(\lambda^{(k)}, \mu^{(k)}) + \left(\sigma_\lambda^{(k)}\right)^T \left(\lambda - \lambda^{(k)}\right) + \left(\sigma_\mu^{(k)}\right)^T \left(\mu - \mu^{(k)}\right) \qquad k = 1, \ldots, K \\ & \mu \geq 0 \end{aligned} \tag{3.13}$$

Both choices for t are compared in the results section of this thesis, along with the subgradient method.

3.2 Model decomposition

Now that the theory has been covered, we will apply it to our model. To do so, we start with choosing the constraints we relax.

3.2.1 Relaxed constraints

The main reason why the computer crashed when running CPLEX on the full model is that the branch and bound tree became too large to be handled, ergo we would like to allow CPLEX to work with smaller problems, so that there are fewer decisions to handle simultaneously. In our algorithm, we decompose the problem into generator subproblems, which means that we have to relax all constraints that link generators together, which occurs when we sum over all generators or over a subset of them.

These constraints are of three kinds : power balance in the grid (equation (2.3)), balance in the fuel network (equation (2.6)) and reserve requirements (equations (2.7), (2.8) and (2.10)). In the rest of this chapter, we will use the notation λ_t^{PB} (Power Balance) for the dual multipliers to equation (2.3), λ_{dt}^{FE} (Fuel Equilibrium) for equation (2.6) and $\lambda_t^{R2+}, \lambda_t^{R2-}$ and λ_t^{R3} for the ancillary services requirements constraints.

Input: An initial value t_0;
Initial dual multipliers $(\lambda^{(0)}, \mu^{(0)})$;
Maximum number of iterations $k_{\max}$;
Set stability center $(\overline{\lambda}, \overline{\mu})$ to $(\lambda^{(0)}, \mu^{(0)})$;
for $k = 0$ **to** $k_{\max}$ **do**
    Solve *Relaxed problem (3.3)*
        Extract subgradient $(\sigma_\lambda^{(k)}, \sigma_\mu^{(k)})$;
        Extract optimal objective value $\theta(\lambda^{(k)}, \mu^{(k)})$;
    Solve *Bundle problem (3.13)*
        Extract new dual multipliers $(\lambda^{(k+1)}, \mu^{(k)})$;
    if $\theta(\lambda^{(k)}, \mu^{(k)}) < \theta(\overline{\lambda}, \overline{\mu})$ **then**
        Update stability center $(\overline{\lambda}, \overline{\mu}) := (\lambda^{(k)}, \mu^{(k)})$;
    end
    If we chose parameter t_k to be dynamic, update it
end
return $\min_k \theta(\lambda^{(k)}, \mu^{(k)})$;

Algorithm 2: Generic bundle method algorithm with maximum iteration number

3.2.2 Subproblems

After the five relaxations have been made, our relaxed problem can be decomposed into multiple subproblems because the interdependency among generators has been removed. In fact, we obtain more than $\#(G)$ subproblems (where $\#$ denotes the cardinality operator). To be more specific, we obtain three types of subproblems : the generator subproblems, a network subproblem and a "trivial" subproblem. Because their constraints are a redistribution of the full model constraints, we do not expand their mathematical form here and only limit ourselves to write the objective function as far as mathematical formulas are concerned, although their complete description can be found in Appendix C.

Trivial subproblem

This subproblem only has the amounts of power bought b_t as decision variables and products between constant parameters and dual multipliers, which are kept constant during each algorithm iteration. Its objective reads

$$\max \sum_{t \in T} \left(-p_t^b b_t - \lambda_t^{PB}(b_t - D_t) - \lambda_t^{R2+} RR_t^{2+} - \lambda_t^{R2-} RR_t^{2-} - \lambda_t^{R3} RR_t^3 \right)$$

We call this subproblem trivial because we have a linear problem in b_t and the only constraints are bounds on these variables, so the solution can be immediately found depending on the sign of the factor before b_t, that is

- $b_t = 0$ if $-\lambda_t^{PB} < p_t^b$;
- $b_t = D_t$ if $-\lambda_t^{PB} > p_t^b$;
- $b_t \in [0; D_t]$ if $-\lambda_t^{PB} = p_t^b$

Network subproblem

This problem only looks at the fuel network, with off-takes and flows as variables (but not variables f_{gt} and f_{gmt}). The constraints it includes are the redistribution of the off-takes into flow through the links and minimum and upper bound constraints. The objective reads

$$\max \sum_{t \in T} \left[-\sum_{c \in C} p_{ct}^{\text{fuel}} o_{ct} - \sum_{d \in FD} \lambda_{dt}^{FE} \left(-\sum_{\substack{l \in L \\ T^*(l)=d}} q_{lt} + \sum_{\substack{h \in PL \\ F^{**}(l)=d}} \tilde{q}_{ht} - \sum_{\substack{h \in PL \\ T^{**}(l)=d}} \tilde{q}_{ht} \right) \right]$$

Generator subproblems

The previous two subproblems were merely "collateral damages", in the sense that our goal was to break down our full problem into generator subproblems, which secondarily led to their appearance. The generator subproblems are those which matter the most in our algorithm because together they include every binary variable from the original problem and aside from minimum off-takes, maximum off-takes and the relaxed constraints, they hold every main constraint from our original problem. The objective for generator g's subproblem has the following expression

$$\begin{aligned} \max \sum_{t \in T} \Bigg[p_t^s s_{gt} - \sum_{m \in M_g^0} VOM_{gm} u_{gmt} - \sum_{a \in A_g} TC_{ga} v_{gat} - p_t^{CO_2} \sum_{m \in M_g^0} E_{gm} f_{gmt} \\ - \lambda_t^{PB} (p_{gt} - s_{gt}) + \lambda_{\text{Spot}(g),t}^{FE} f_{gt} \\ - \sum_{g \in G} \sum_{m \in M_g^0} (\lambda_t^{R2+} r_{gmt}^{2+} + \lambda_t^{R2-} r_{gmt}^{2-} + \lambda_t^{R3} (r_{gmt}^{3S} + r_{gmt}^{3NS})) \Bigg] \end{aligned}$$

Computing the upper bounds

Now that all the subproblems to be used in our algorithm have been defined, we can address how we compute the upper bound to our full model. This task simply consists in solving each of the subproblems separately, and summing their objective function optimal value. Finally, when computing our upper bounds, we also keep the plant commitment u_{gt} along with the subgradient of our relaxed problem in memory, which are later used in the feasible solution recovery method and the dual multiplier updating process, the former being presented in section 3.3.2.

3.2.3 Initial dual multipliers

Up to now, we have seen which dual multipliers were the most interesting ones for our approach, how to update them and use them once updated. To finish up with the upper bound computation part of our algorithm, we go to the very beginning of our algorithm and tackle the choice of initial values for these initial dual multipliers.

For small optimization problems, such a choice may not really matter because they can be solved relatively fast by solvers and quickly reach a satisfactory value, as opposed to this case where the choice is critical. Because our model contains plenty of details to be as accurate as possible and our time horizon is of 10 months (7344 time periods in the data), the time taken to solve a single relaxation is of approximately 8 minutes. Starting from a point which is completely off from optimality costs us a lot because even if it takes

Input: Problem instance data;
Dual multipliers (λ, μ);
Function `Compute_Upper_Bound`(λ, μ)
  foreach $g \in G$ **do**
    Solve *Generator g subproblem*
      Extract plant commitment u_{gt};
      Extract optimal objective function f_g^{GEN};
  end
  Solve *Network subproblem*
    Extract optimal objective function f^{NET};
  Solve *Trivial subproblem*
    Extract optimal objective function f^{TRIV};
  Compute profit upper bound $f = \sum_{g \in G} f_g^{GEN} + f^{NET} + f^{TRIV}$;
  Compute subgradients $(\sigma_\lambda, \sigma_\mu)$;
  return $(u_{gt}, f, \sigma_\lambda, \sigma_\mu)$;

Algorithm 3: Profit upper bound computation using Lagrangian relaxation

only 20 iterations to reach an acceptable value, those 20 iterations are equivalent to two hours (if we neglect the second part of our algorithm which is feasible solution recovery), which is already twice the time limit we would like not to go beyond.

Our choice of dual multipliers is actually a relatively simple one but is, at least intuitively, not such a bad one and consists of taking the dual multipliers to the full problem linear relaxation. Thus, we relax the integrality of all the binary variables and replace the integrality type by lower and upper bounds of 0 and 1. This choice of initialization can also be found in [CBFL$^+$09] and [Pap13b].

Input: Problem instance data;
Function `Initial_Dual_Multipliers()`
  Solve *Full problem linear relaxation*
    Extract dual multipliers $(\lambda^{LIN}, \mu^{LIN})$ of complicating constraints;
  return $(\lambda^{LIN}, \mu^{LIN})$;

Algorithm 4: Getting initial dual multipliers

3.3 Feasible solution

The previous sections have shown how our algorithm can get a sequence of upper bounds on our original maximization problem, but not how to get a feasible solution. This section focuses on this subject. In what follows, we first show how to find an initial feasible solution by doing some simplifications, and then how to recover other feasible solutions during the course of our sequence of upper bounds computations, keeping in mind that controlling and preventing any time explosion is also a priority.

3.3.1 Initial solution and simplifications

Whether it is for computing upper bounds to our problem with subgradient methods or using bundle methods, a lower bound is always useful information, and the better the bound, the more practical it is to evaluate a new step α_k for subgradient methods or the r variable for bundle methods. In this first section, we aim at simplifying our problem to make it solvable “fast enough”. In the upcoming paragraphs, we will explain step by step how we arrived at our simplified model.

Binaries

We already mentioned a few times that the main reason why our computer crashed when trying to solve the full model was because the binary tree had become too large to be contained in the laptop random access memory. A linear problem does not have this kind of issue (unless the model given to the solver is just too big, but this is another issue). Sending fewer binary variables to CPLEX should therefore make the optimization more likely to terminate. To get an idea on how to do so, the following fact in our model is of great use :

Fact 9 *When choosing the values of the binaries, choosing the value u_{gt} is predominant over the choice of u_{gmt}, which is itself predominant over u_{gmst}, in the sense that we first have to decide whether or not a generator will be on, then what mode it will work in, and finally on what slices it will produce. Besides, we have the following relation*

$$u_{gt} \geq u_{gmt} \geq u_{gmst}$$

and also

$$\#(u_{gt}) \leq \#(u_{gmt}) \leq \#(u_{gmst})$$

From this simple fact, we can get the intuition that the binaries u_{gmst} are too many compared to how much they matter for our model, hence as far as a simplified model is concerned, those binaries should not be included.

Heat-rate curve

Filling the slices with production in the correct order can only be guaranteed if we use the variables u_{gmst}, so removing those binaries would allow solutions where the slice order is not respected, which is illustrated in Example 10.

Example 10 *Suppose we have a piecewise-linear heat-rate curve as in Figure 3.3. In general, removing the binaries would make the solver first fill slice S_3, then S_2, S_4, S_5 and finally S_1. Suppose that for a given hour, information on the fuel cost and power selling price make it worthwhile producing at a heat-rate between the heat-rate values of S_2 and S_4. Removing the binaries would make the optimal solution more likely to have full production on slices S_2 and S_3, and nothing on the others, but when going back to the original problem, we have on obligation to have full production on S_1 before we are allowed to move to S_2 and S_3. The problem with this is that having full production in S_1, S_2 and S_3 is not cost-effective anymore.*

This shows that there is a conflict to be fixed between removing the binaries and decreasing heat-rate curves. To this aim, when we simplify our problem we first remove the slice binaries and look at the pattern in the heat-rate curve. Starting at slice 1, we look at the next slice and, if its heat-rate is lower than that of the first slice, we merge those two slices together, renumber the whole set of slices and move on to the next slice. Note that if we put all slices together, we fall back to the linear approximation. We could therefore

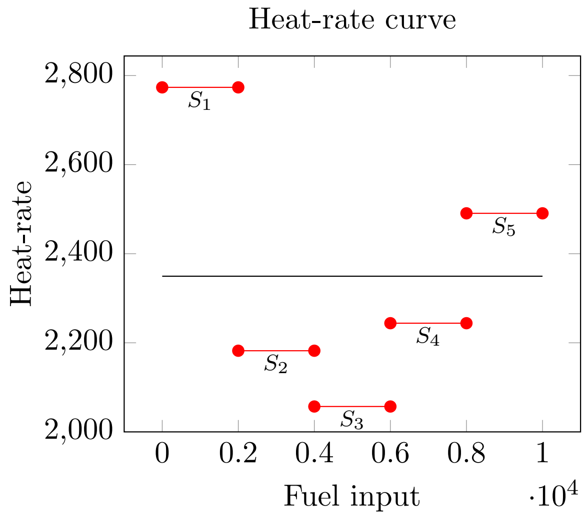


Figure 3.3: Heat-rate for the piecewise-linear fuel consumption curve (red lines) and its linear approximation (black line)

argue that using the linear approximation would be easier than keeping multiple slices (in a smaller number) for our simplified problem but using 2 slices can still sometimes make the difference between starting up a generator and keeping it down. Figure 3.4 shows how our simplification works and what "re-slicing" we obtain with the initial heat-rate curve from Example 10, for which we do not use binaries.

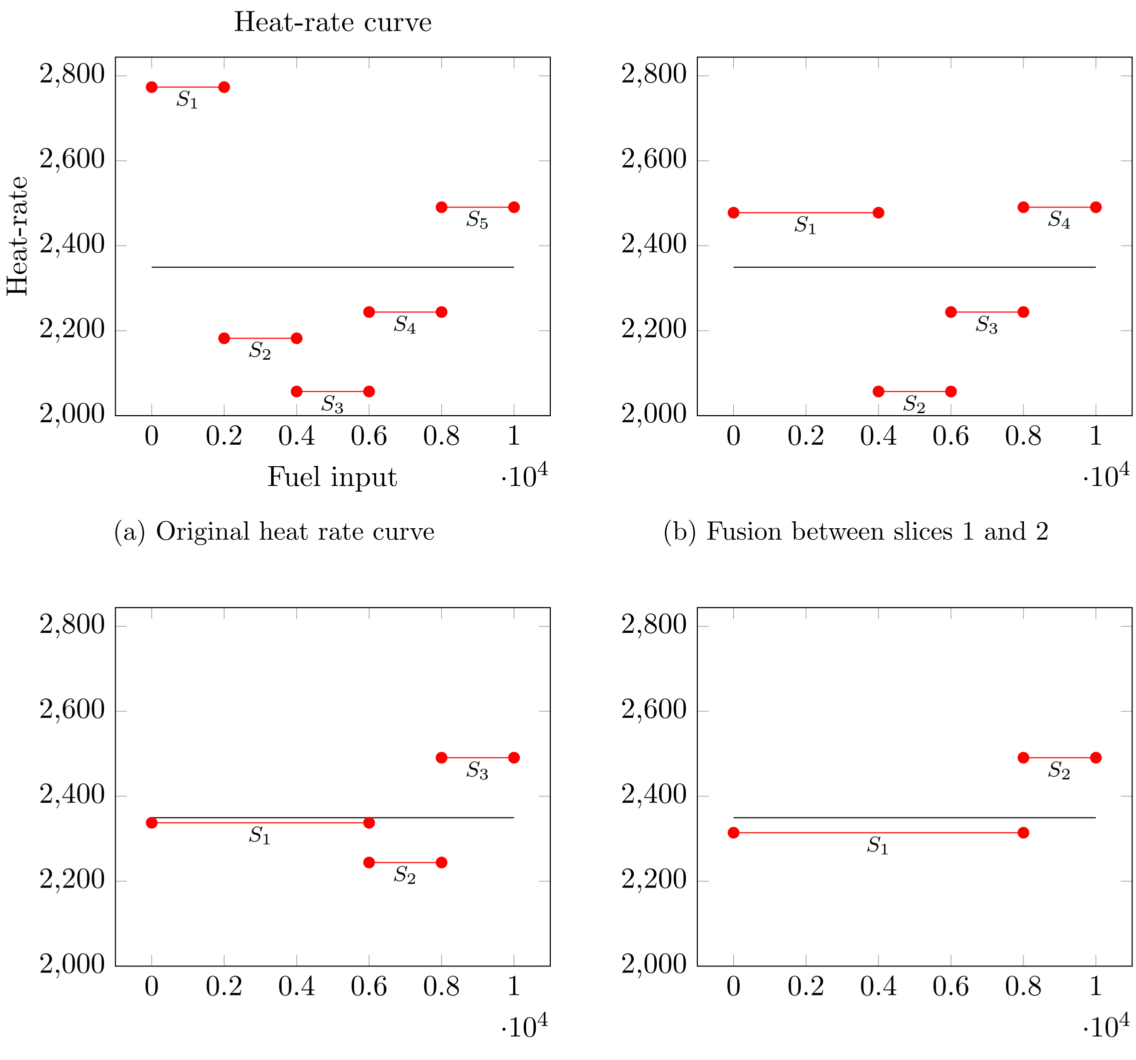
Predominant feasibility factor

After removing all the u_{gmst} binaries and reslicing our heat-rate curve, we tried to run the simplified model and saw that even though the algorithm now managed to find a solution to this new problem, the time taken to do so was still too large, which meant we had to keep searching for a new simplification for our model and possibly reduce the amount of binaries. To do so, we first had to analyze what the significant constraints were in our model, that is constraints that really caused generators to do something or prevent them to act as they would like to. These constraints are of two types : minimum/maximum off-takes on contracts, and reserve requirements. This implies that our next simplification has to maintain as best as possible the interdependencies between generator decisions and those constraints. If we look at all possibles binary variables u, we have 4 potential candidates for removal[11] : u_{gt}, u_{gmt}, u_{gmlt} and the profiles binaries $u_{gmt}^{SDP}/u_{gmt}^{SUP}/u_{gmt}^{DISP}$. The first two are clearly not to be removed since they represent the foundations of our model, which means we are left with either removing temperature-dependency (u_{gmlt}) or removing profiles when building the simplified model.

Profiles or temperature dependency

When a generator is working under a profile, it cannot contribute to any kind of reserves. This means that choosing to remove profiles when simplifying could make a generator contribute to ancillary services when it is in fact incapable of doing so. Besides, profiles

[11] v and z are mainly used to simplify notations and having all values for every u immediately fixes values for v and z



(a) Original heat rate curve

(b) Fusion between slices 1 and 2

(c) Fusion between slices 1-2 and 3

(d) Final heat-rate curve

Figure 3.4: Example of re-slicing on initial heat rate curve shown in Figure 3.4a. If the heat rate at which points it becomes profitable to produce power was located between the linear approximation heat rate and the first slice from the resulting resliced piecewise-linear curve, the choice of turning on or shutting down the generator would be different for these two approximations.

completely fix production and fuel consumption when they are active, and the removal of profiles therefore creates an artificial consumption flexibility that is not applicable in reality. Removing profiles would thus lead us to a situation where both feasibility factors mentioned above are “damaged”.

Temperature dependency influences two things : startup consumption and which profile the generator will follow when starting up, if the power plant has profiles. To simplify temperature dependency, we cannot just remove the binaries but instead have to choose a *reference temperature state*, that is we restrict the set of possible temperature states a generator can be in to only one state. Because as we just saw, profiles preclude contribution to reserves, and different temperature states may have different profile lengths, the reference state chosen will be the one with the longest profile. This choice is motivated by the fact that when going back to the original problem, the effective zero-contribution length will never be longer than what our simplified problem returned and thus can be seen as a cautious choice that cannot create infeasibilities for the provision of ancillary services. However, from a fuel off-take point of view, the fixed fuel consumption could create infeasibilities regarding the minimum/maximum off-take requirements, because we could end up with a higher fuel consumption than permitted when going back to the original problem. That being said, doing simplifications often results in loss of precision/applicability, and between both options available (profiles or temperatures), the choice of temperature simplification seems to be the less damaging one because it only conflicts with minimum and maximum off-take, as opposed to the removal of profiles that conflicts with both feasibility factors mentioned in the previous paragraph. Finally, because the longest profiles will often be associated to the coldest temperature state the generator can startup in, this state is the one we choose for our simplified problem, thus also resulting in the highest startup fuel consumption we have.

In the end, we have the problem simplification method shown in Algorithm 5.

Input: Problem instance data;
foreach $g \in G, m \in M_g$ **do**
    `/* Reslicing */`
    foreach $s \in S_{gm}$ *such that s is not last slice* **do**
        if *Next slice has smaller heat-rate* **then**
            Merge slices s and $s+1$ together;
            Go back to 3;
        end
    end
    `/* Establishing reference temperature state */`
    if (g, m) *has Startup Profiles* **then**
        Get coldest temperature state θ^*_{gm};
        foreach $l \in \theta_{gm} \backslash \{\theta^*_{gm}\}$ **do**
            Set profile for temperature state l to profile for θ^*_{gm};
            Set SUC_{gm} for temperature state l to SUC_{gm} for θ^*_{gm};
        end
    end
end

Algorithm 5: Model simplification algorithm

Back to the original problem

After re-running tests on our newly simplified model, we obtained a feasible solution for this model in a reasonable amount of time (10 minutes). The issue here is that even though this solution is highly likely to be feasible for the original problem and maximizes profit, the said profit has been maximized for simplified data. We could just reuse this solution and recalculate the profit for our original problem, which would be a valid choice, of course, but something even better can be done.

Imagine we do not take the whole obtained solution but only parts of it, fix the values, and re-optimize the full problem with the corresponding part of the original problem fixed to this partial solution. This would reduce the total elapsed time for solving the full model and partly re-introduce the original information of the model, before simplifications were made. We are thus here left with the choice of the variables we want to fix.

We reuse Fact 9 and choose to retain the binaries u_{gt}. This choice is motivated by multiple factors : when re-optimizing, the idea is to get as close as we can get to the original problem, but without re-creating the full complexity of the model, mainly originating from the conjoint decision over generators to fulfill ancillary services requirements and off-take bounds, all the while using binaries to achieve such goals. If we fix u_{gt}, we already force the decision to run a plant or not, and the choices left to the solver are which mode to use if it has to be on, and how much to produce. Intuition would suggest that these new decisions are easier to make than choosing whether to start a plant or not. Experiments in our model indeed showed that these decisions were much faster to make (if we fix u_{gt}, the time taken was around 2 minutes for the re-optimization).

To sum this section up, we have finally found a feasible solution that has an optimized profit to a certain extent, because the plant commitment was decided based on an acceptable but not perfect representation of our original model. In a way, this is already better than a mere recourse to CPLEX, which crashed before returning a solution.

Input: Problem instance data;
Function `Initial_Feasible()`
  Simplify data (no decreasing slices, no temperature-dependency);
  Solve *Full problem with simplified data*
    Extract plant commitment u_{gt}^{SIMP};
  Solve *Full problem with original data and commitment fixed to* u_{gt}^{SIMP}
    Extract feasible objective f^{FEAS};
  return $(u_{gt}^{SIMP}, f^{FEAS})$;

Algorithm 6: Getting initial feasible solution

3.3.2 Feasibility recovery

We saw in the previous section how to get a feasible solution. Even though this is not a bad thing, this solution could be a good one but not a great one (our goal is to have an at most 1% duality gap for the solution to the optimization problem). Even if we have found the best bound for our relaxed problem, that is the smallest upper bound, our feasible solution may still not be good enough and will actually never be if we do not find ways to retrieve more feasible solutions, one of them hopefully better than the initial one. In this section, we will see how we manage this goal using the other part of the algorithm, that is the computation of upper bounds. This section is an adaptation of a method presented in [TB00].

Consider (λ^*, μ^*) the optimal values for (λ, μ) in our dual problem, meaning those that make the dual problem reach its smallest value. If strong duality holds, then we know that a solution given by the dual problem using dual multipliers (λ^*, μ^*) is feasible and optimal for the primal problem. The issue is that we have no guarantee that strong duality holds. Even if it held, waiting for our algorithm to terminate (when the dual multipliers have finally converged) and then take the optimal solution from the dual could take hours and is therefore not an option. This is no reason to despair and we can still try to use what we have as best we can. Weak duality implies that the optimal solution found with our dual algorithm will not necessarily be primal feasible, but here, as in the previous section, we can make readjustments to get something feasible.

We have seen in the previous section that when computing the initial feasible solution, we first solve a simplified problem, take its optimal plant commitment, plug it into the full problem, and re-optimize with this commitment as fixed. We can use a similar yet slightly different approach to recover feasible solutions from our dual-optimal commitment. With the commitment, the re-optimization on the full problem was done efficiently and hence can be reused as is, which means we now have to find what new commitment to feed to the full original model.

Because we start from initial dual multipliers that solve the linear relaxation and use generally small steps in our subgradient algorithm, our dual multipliers should often be a relatively good approximation to their optimal values. The upper bound commitments we retrieve at each iteration should therefore also be good approximations to the optimal commitments in the full problem, in the sense that although they may not lead to a feasible solution, they carry a representation of how penalizing it is to violate constraints to a certain extent.

Suppose that at iteration k, the optimal commitment to the dual problem is $U_g^{(k)} = \{u_{gt}^{(k)}\}_{t \in T}$. Even though this commitment is based on a lot of information from the original problem, this information is not perfect, and fixing this commitment when re-optimizing might not be a good idea because there could be no feasible solution for this specific planning. However, and this is where our initial feasible solution comes in handy, we computed in the previous section a commitment which **is feasible** for our primal problem. An idea could be to try a mix between those two solutions and see how it works, but we have to determine how to mix them. We have two options here :

- If both u_{gt} from feasible and dual schedules have the same value for a given generator and a given time t, we force this value, else we let the binary choice; we apply this for every generator and time period;

- If for a generator, the whole planning $U_g^{(k)}$ is the same as the whole feasible planning for this generator U_g^{FEAS}, we force this planning, else we make a choice between the two schedules;

Although the first option allows much more flexibility, the second allows more control on the solving time of the model, because for every generator there is at most 2 choices for the commitment. Since time is a critical factor for us, we choose the second approach, and introduce two new binary variables, both generator-dependent, x_g^{FEAS} and x_g^{UB} , and add the two following constraints

$$x_g^{FEAS} + x_g^{UB} = 1 \qquad \forall g \in G$$

$$u_{gt} = u_{gt}^{(k)} x_g^{UB} + u_{gt}^{FEAS} x_g^{FEAS} \qquad \forall g \in G, t \in T$$

We now have to choose between adding these constraints on the full model or on the simplified problem. Although these constraints have made the decision process easier for the solver, the full model is still too large to be solvable fast enough with those two additional elements, which apparently means it can only be solved with a fixed plant commitment and that a choice between plant commitments is not restrictive enough. That being said, the simplified model now has one useless simplification, which is temperature-dependency, because a fully determined schedule fixes any choice between temperature states, meaning we can re-introduce this detail into the model when we force the solver to make a choice between two schedules.

To sum up, all of what is written above aims at explaining the decisions retained for feasible solution recovery, these decisions being :

- To first retrieve the full planning for every generator from the initial feasible solution presented in the previous section;
- To extract the full planning for every generator from the relaxed generator subproblem;
- To solve a simplified model where successive slices with decreasing heat-rate have been merged and where a choice has to be made between both schedules;
- To take the plant commitment u_{gt} from this simplified model, feed it into the full problem and re-optimize;

This guarantees that a feasible solution can be found (because in the worst-case-scenario, we retrieve our feasible initial solution) and if after the above steps, the primal objective function has increased, we update our feasible solution, so that we always keep track of the best feasible solution found so far.

Input: Problem instance data;
Feasible plant planning u_{gt}^{FEAS};
Optimum f^{FEAS} to the full problem when commitment fixed at u_{gt}^{FEAS};
New plant commitment u_{gt}^{UB}, not guaranteed to lead to a feasible solution;

Function `Recover_Feasible(`$f^{FEAS}, u_{gt}^{FEAS}, u_{gt}^{UB}$`)`
  Simplify data (no decreasing slices);
  Solve *Full problem with simplified data and commitment fixed, for every generator, to either* u_{gt}^{FEAS} *or* u_{gt}^{UB}
    Extract plant commitment u_{gt}^{FEAS2};
  Solve *Full problem with original data and commitment fixed to* u_{gt}^{FEAS2}
    Extract feasible objective f^{FEAS2};
  if $f^{FEAS} > f^{FEAS2}$ **then**
    return $(u_{gt}^{FEAS}, f^{FEAS})$;
  else
    return $(u_{gt}^{FEAS2}, f^{FEAS2})$;
  end

Algorithm 7: Recovering feasible solution

3.4 Duality gap : from theory to practice

3.4.1 Generator subproblems

During the course of our algorithm tests, we sometimes encountered an issue with CPLEX and some generator subproblems. Although these subproblems have much fewer binary variables than the initial full model, they still contain a lot of these variables, and finding both a feasible solution and a 0% optimality guarantee can be cumbersome for the solver. For example, in one of the subproblems CPLEX managed to find a solution with 0.12% optimality gap in less than 2 minutes, but then got stuck at this guaranteed threshold for 3 hours and could have crashed the computer if we had not terminated the algorithm.

Let ω be the optimality guarantee held by the solver through the run, which decreases with time. Even though the rationale behind our algorithm stays valid and the complexity of our model is eased, the issue mentioned in the previous paragraph forces us to make some adaptations in order to avoid an iteration taking 5 hours for completion or crashing the computer again. Assume we set a time limit $t^{\max}$ for each generator subproblem and an optimality guarantee threshold Ω, meaning the solver will stop either when the time elapsed goes beyond $t^{\max}$ or when the ω is below Ω and then switch to the next subproblem or to the next part of the algorithm. Those limitations mean that when we wrote that Algorithm 3 returned the optimal objective value f_g^{GEN}, we were in fact too optimistic because we actually only end up with an interval $[\underline{f_g^{GEN}}; \overline{f_g^{GEN}} = \underline{f_g^{GEN}}(1+\omega)]$, containing f_g^{GEN}.

Because our dual optimal solution is used to get a guarantee on the optimality of our primal feasible solution, the original problem optimal value **cannot be higher than** the dual optimal value $f^{DUAL}(f^{GEN}) = \sum_{g \in G} f_g^{GEN} + f^{NET} + f^{TRIV}$. We thus cannot use $\underline{f_g^{GEN}}$ to get this guarantee because it is an under-estimation of f_g^{GEN} and the resulting upper bound obtained via the solver could be smaller than the real upper-bound that should have been returned in theory. Our optimality guarantee would therefore be too optimistic and could even be smaller than our feasible solution, which is why we have to use the value $\underline{f_g^{GEN}}(1+\omega)$, an over-approximation, making the upper-bound of our solver higher than the real one, hence a pessimistic but valid guarantee instead of an optimistic but possibly wrong one.

Example 11 *Consider an extreme yet very illustrative example with only one generator, where the time limit set for the solver is so small that it barely has time to do any calculations and ends up with a* $\underline{f_g^{GEN}}$ *of* $-\infty$ *and a* $\overline{f_g^{GEN}}$ *of* $+\infty$ *for every generator subproblem. Using* $\underline{f_g^{GEN}}$ *to get an upper bound will tell us that our optimal objective to our initial maximization problem is bounded from above by* $-\infty$*, meaning the problem is infeasible. However, this bound is just due to a small time limit* $t^{\max}$ *and has nothing to do with primal infeasibility. Conversely, using* $\overline{f_g^{GEN}}$ *gives us an upper bound of* $+\infty$*, which states that because the time limit was too small, the solver did not have enough time to get an upper bound, and does not make any claim on primal infeasibility.*

The above example also suggests that the choice of parameter $t^{\max}$ can be critical, because setting it too low destroys the whole method and setting it too high could make a single iteration take longer than the full algorithm's time limit. Finally, the reviewed version of Algorithm 3 is shown in Algorithm 8.

Input: Problem instance data;
Dual multipliers (λ, μ);
Time limit $t^{\max}$ and desired optimality Ω for the generator subproblems;

Function `Compute_Upper_Bound_Reviewed`(λ, μ)
  foreach $g \in G$ **do**
    Solve *Generator g subproblem with settings* $(t^{\max}, \Omega)$
      Extract plant commitment u_{gt};
      Extract found optimal objective function $\underline{f_g^{GEN}}$ and duality gap ω_g;
  end
  Solve *Network subproblem*
    Extract optimal objective function f^{NET};
  Solve *Trivial subproblem*
    Extract optimal objective function f^{TRIV};
  Compute profit upper bound $f = \sum_{g \in G} \underline{f_g^{GEN}}(1 + \omega_g) + f^{NET} + f^{TRIV}$;
  Extract subgradients $(\sigma_\lambda, \sigma_\mu)$;
  return $(u_{gt}, f, \sigma_\lambda, \sigma_\mu)$;

Algorithm 8: Reviewed version of the profit upper bound computation using Lagrangian relaxation due to the optimality gap of the branch and bound algorithm (differences with Algorithm 3 are highlighted in red).

3.4.2 Bundle algorithm

A similar issue arises when we use the bundle algorithm master problem (3.13). The idea is to use the sub-approximation given by the subgradients and to move along the hyperplane they define. The problem is that we do not know whether the cutting plane at our disposal, i.e. the one found using CPLEX, actually corresponds to the theoretical cutting plane that would have been found using the exact optimal value to the relaxed problem with fixed parameters $(\lambda^{(k)}, \mu^{(k)})$.

In our case, although we set aside the hyperplane slope inaccuracy (because the subgradient found in practice might not be the one associated to the real optimal value), we use the fact that these cutting-planes are supposed to give a sub-evaluation of $\theta(\lambda, \mu)$. To coincide with this idea, we need to use the smallest under-approximation we have at our disposal to avoid any loss of information. This is why the cutting plane constraints used in the master problem will use the profit upper bound $\sum_{g \in G} \underline{f_g^{GEN}} + f^{NET} + f^{TRIV}$ instead of the formula shown in Algorithm 8. This choice is motivated by the fact that the latter needed to consider the highest upper bound possible to **guarantee** optimality whereas the cutting planes in the master problem use under-approximations of the profit, hence the smallest upper bound possible.

3.5 Our algorithm

Now that every aspect of our model has been overviewed and discussed in detail, we can finally present our main algorithm, which is shown in Algorithm 9 and Figure 3.5.

Input: Problem instance data;
One of the methods to update dual multipliers (subgradient, bundle, cutting plane);
Maximum number of iterations $k_{\max}$;

`/* Initial dual multipliers */`
$(\lambda^{(0)}, \mu^{(0)})$ = `Initial_Dual_Multipliers()`;
`/* Initial feasible solution */`
$(u_{gt}^{FEAS}, f^{FEAS})$ = `Initial_Feasible()`;
`/* Starting Iterative bounds computation */`
for $k \leftarrow 0$ **to** $k_{\max}$ **do**
  `/* Upper bound */`
  $(u_{gt}^{(k)}, f^{(k)}, \sigma_{\lambda}^{(k)}, \sigma_{\mu}^{(k)})$ = `Compute_Upper_Bound_Reviewed(`$\lambda^{(k)}, \mu^{(k)}$`)`;
  `/* Lower bound */`
  $(f^{FEAS}, u_{gt}^{FEAS})$ = `Recover_Feasible(`$f^{FEAS}, u_{gt}^{FEAS}, u_{gt}^{(k)}$`)`;
  $(\lambda^{(k+1)}, \mu^{(k+1)}) \leftarrow$ Update dual multipliers $(\lambda^{(k)}, \mu^{(k)})$;
  Compute duality gap : $GAP^{(k)} = ((\min_k f^{(k)}) - f^{FEAS})/f^{FEAS}$;
end
return $(u_{gt}^{FEAS}, f^{FEAS}, GAP^{(k_{\max})})$

Algorithm 9: Algorithm with maximum number of iterations as stopping criterion

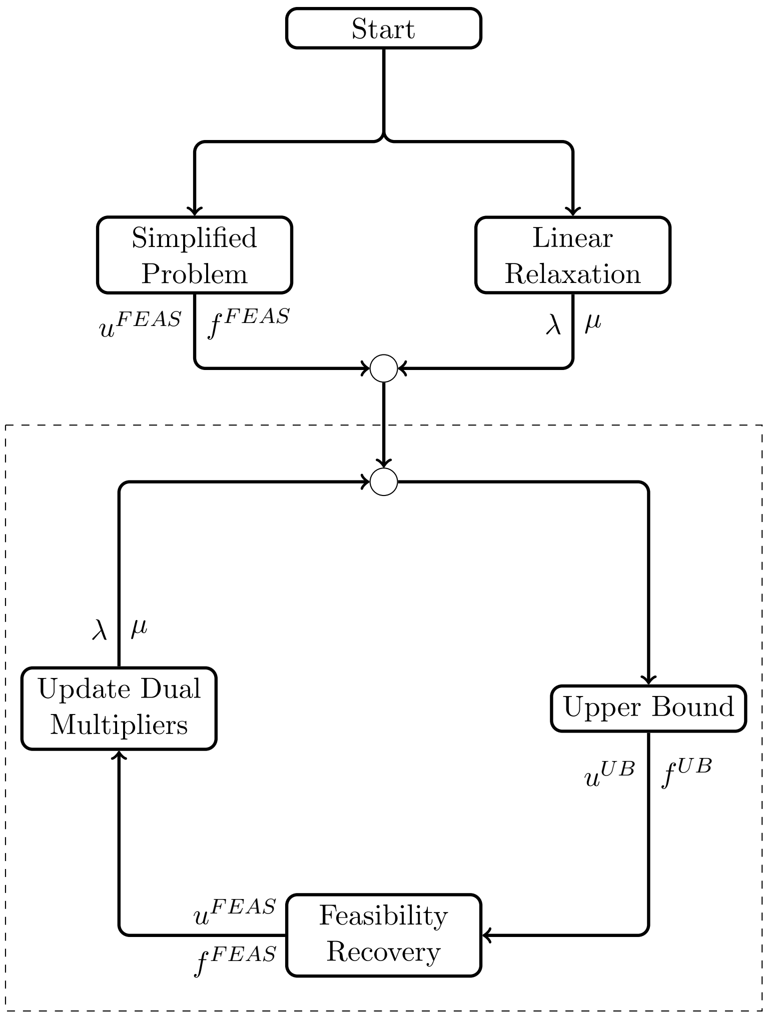


Figure 3.5: Flow-chart of our algorithm. The "Feasibility Recovery" and "Update Dual Multipliers" blocks can be performed simultaneously depending on the choice of update rule for the dual multipliers update. For example, a Polyak step-size rule should be improved if a better approximation of the true optimal objective function is available, improvement that can only be obtained via the feasibility recovery step of our algorithm. The same goes for a bundle algorithm where a lower bound (our current best feasible objective function) can be set on r to give additional information to the model.

Chapter 4

Results

Now that the "What do we want to do?" and the "How do we want to do it?" questions have been answered, it is time for us to get to the two final questions of this thesis, "Does it work?" and, if so, "How well does it work?". In this chapter, we show that although our algorithm seems to be effective, hence a positive answer to the former, the answer to the latter is a bit trickier depending on the method used.

In what comes next, we compare the results returned by our algorithm depending on the choice of initial dual multipliers (section 4.1), on the choice of iterative method for the dual multipliers update (section 4.2) and on different settings for the optimality gap threshold and the time limit of generator subproblems (section 4.3). For the choice of iterative method, five tests have been launched for the subgradient algorithm and two for the bundle algorithm. Those five subgradient tests correspond to four Polyak rules (with a damping factor of 1, 0.5, 0.1 or 0.01) and a constant step-size rule (10^{-6}) and the two bundle tests are either a static scaling factor t (10^{-7}) or a dynamic parameter, using the update formula (3.12), with an initial t of 10^{-7}.

These tests are launched on the first dataset that we received from GDF-Suez[12]. Other problem instances are used in the next chapter to test the robustness of the algorithm.

4.1 Initialization of dual multipliers

In this section, we compare three choices of initial dual multipliers :

- an initial value of zero;
- the value found in the initial feasible solution search, when we fed a simplified commitment to the full problem and re-optimized with this forced commitment;
- the value we presented in the previous chapter, which is the dual multiplier associated to the to-be-relaxed constraints in the linear relaxation of the full problem.

The different results for the first iteration can be found in Table 4.1[13]. These results confirm our intuition that the smaller the upper bounds found, the better our feasibility recovery method should work, because points close to optimality for the dual problem should be close to optimality for the primal despite the absence of strong duality.

[12] Unless stated otherwise, the solver CPLEX was used with the following settings : duality gap tolerance of 0.5% for the generator subproblems, time limit for the generator subproblems of 5 minutes.

[13] Because the third choice implies solving the linear relaxation of the full problem first, this choice takes an additional 12 minutes compared to the two others.

Choice	Zero	Initial Feasible	Linear Relaxation
Gap Before Recovery (%)	172.96	2.86	1.07
Gap After Recovery (%)	172.95	2.47	0.50
LB increase (%)	0.004	0.38	0.57
Time Elapsed (min)	26	26	38

Table 4.1: Results during and at the end of the first iteration, for different choices of initial dual multipliers. Gap before recovery refers to the duality gap between our initial feasible solution and our initial upper bound. Gap after recovery refers to the duality gap between the first recovered solution (mix between initial feasible and initial upper bound commitments) and our initial upper bound. Lower bound increase refers to the increase in our primal objective function obtained after the feasibility recovery step of our algorithm was applied.

4.2 Iterative method

When starting our algorithm on a test case, we do not have a large amount of iterations at our disposal because of the problem instance size. In fact, the time limit of an hour was reached around the end of the third iteration (the end of the third iteration occurred between 3590s and 3670s). In the following section, we first look at how the algorithm behaves when the time limit of an hour is up, which is what is of interest for practical purposes, and then how it behaves after a larger number of iterations.

As we discussed in the introduction of this chapter, we use five possible step-sizes for subgradient methods, with a max iteration number of 14, and two scaling factors for bundle methods, with a max iteration number of 99. The difference in the chosen iteration number for both methods is mainly linked to the fact that bundle methods need time to *calibrate*, in the sense that they first need a certain amount of iterations to get enough cutting planes to have an acceptable representation of $\theta(\lambda, \mu)$, whereas the subgradient method has no memory of past progress as far as step-size or choice of direction is concerned[14].

The results after an hour are shown in Table 4.2. In this Table, the lower bound increase and upper bound decrease respectively correspond to the change in our feasible (primal problem) objective that increases iteration after iteration using feasibility recovery, and to the decrease in the dual objective function. In addition, the results when we allow for a larger number of iterations are shown in Table 4.3. A few comments can be made about these results :

- For subgradient methods, larger Polyak rules, that is Polyak rules with a higher value of γ, seem to give better results. Our intuition for this is that a larger Polyak factor allows for more *traveling* in the upper bound computations, hence the commitments retrieved from these computations allow for more diversity than those retrieved with much smaller steps. However, because we still use a Polyak rule, this commitment does not violate feasibility to a large extent in a single iteration, meaning the feasibility recovery now has two possible schedules, both assumed good enough, and relatively different from one another, instead of two barely distinguishable plannings, which is what would be obtained with really small step sizes;

[14]The very first iteration of the algorithm for every setting presented in this section is exactly the same because this section is devoted to changes in the update of the dual multipliers, which has no effect in the first iteration since the first update is used in iteration 2.

- For bundle methods, we first noticed that, as expected, a dynamic update of the scaling parameter is better than no update at all. The use of bundle methods with a dynamic update actually gives us the best results in terms of upper bounds, although the best gap is still above 0.35%. The only drawback of this method, and in fact the most crucial one, is that in the beginning of the algorithm we have a few iterations that lead to no improvements at all, for *calibration* of the cutting-plane approximation of $\theta(\lambda, \mu)$ and the scaling parameter t, but the solving time of a single iteration takes too long in our model to permit calibration.

- Whether we look at Table 4.2 or 4.3, and whether we consider bundle or subgradients methods, the last column in every case seems to suggest that the upper bound computation could be improved. Of course, this suggestion is only valid if the remaining duality gap of 0.36% is not entirely due to the absence of strong duality, meaning that our upper bound is actually optimal but having only weak duality prevents us to do better than that with our method. The sequence of upper bounds that CPLEX found when solving the initial full problem (when it ended up crashing the computer) was such that the smallest upper bound found by CPLEX was 0.05% higher than our best feasible solution found (found with Polyak rule using a γ factor of 0.5), which means that we have two possible scenarios here :

  - This better upper bound found by CPLEX has been computed using branch and bound and cannot be recovered by Lagrangian relaxation due to the duality gap;
  - This upper bound could be found using our algorithm, but using different parameters, dual multiplier updates, subgradients or else;

  As we mentioned in the end of the previous chapter, one of the problem that may occur in our algorithm is the optimality gap of branch and bound when we solve the generator subproblems. Because, as of now, the results were presented using a tolerance duality gap of 0.5%, we could try a stronger tolerance and see if this changes anything in the algorithm behavior.

- Given the 0.05% gap between CPLEX upper bounds and our feasible solution, we conclude that our feasibility recovery method is behaving quite well;

4.3 Generator subproblems, optimality gap and time limit

When we request a lower optimality gap for each generator subproblem, we cannot avoid an increase in the time taken for the full algorithm to terminate. In this section, we will look at how our best result obtained in the previous section (a subgradient method, using Polyak rule, with a γ factor of 0.5) varies using three settings for CPLEX, for both the hour time limit and the 14 iteration limit, as we did in the previous section. These settings are the following :

- Regular settings, which we already used to build Tables 4.2 and 4.3 : optimality gap of 0.5% and time limit of 5 minutes;

- Stronger settings : optimality gap of 0.1% and time limit of 5 minutes;

- Stronger settings coupled with shorter time limit : optimality gap of 0.1% and time limit of 3 minutes;

Choice	End Gap (%)	LB incr. (%)	UB decr. (%)
Subgradient			
Polyak (1)	0.438	0.629	$1.487 \cdot 10^{-3}$
Polyak (0.5)	0.434	0.634	$8.51 \cdot 10^{-4}$
Polyak (0.1)	0.468	0.600	$1.185 \cdot 10^{-3}$
Polyak (0.01)	0.485	0.583	$3.01 \cdot 10^{-4}$
Constant (10^{-6})	0.473	0.596	0
Bundle			
Static (10^{7})	0.470	0.599	0
Dynamic	0.447	0.622	0

Table 4.2: Results when the time limit of an hour has been reached. This time limit always corresponds to a total of three algorithm iterations because the dual multipliers update takes less than a second and the time taken to solve the different blocks of our algorithm did not vary much depending on the chosen step-size update settings.

Choice	End Gap (%)	LB incr. (%)	UB decr. (%)
Subgradient (14)			
Polyak (1)	0.386	0.681	$1.487 \cdot 10^{-3}$
Polyak (0.5)	0.382	0.686	$8.51 \cdot 10^{-4}$
Polyak (0.1)	0.400	0.667	$2.187 \cdot 10^{-3}$
Polyak (0.01)	0.434	0.634	$1.117 \cdot 10^{-3}$
Constant (10^{-6})	0.453	0.615	$1.476 \cdot 10^{-3}$
Bundle (99)			
Static (10^{7})	0.434	0.599	$2.907 \cdot 10^{-4}$
Dynamic	0.375	0.681	$1.33 \cdot 10^{-2}$

Table 4.3: Results when we do not care about time limit and allow for a higher number of iterations

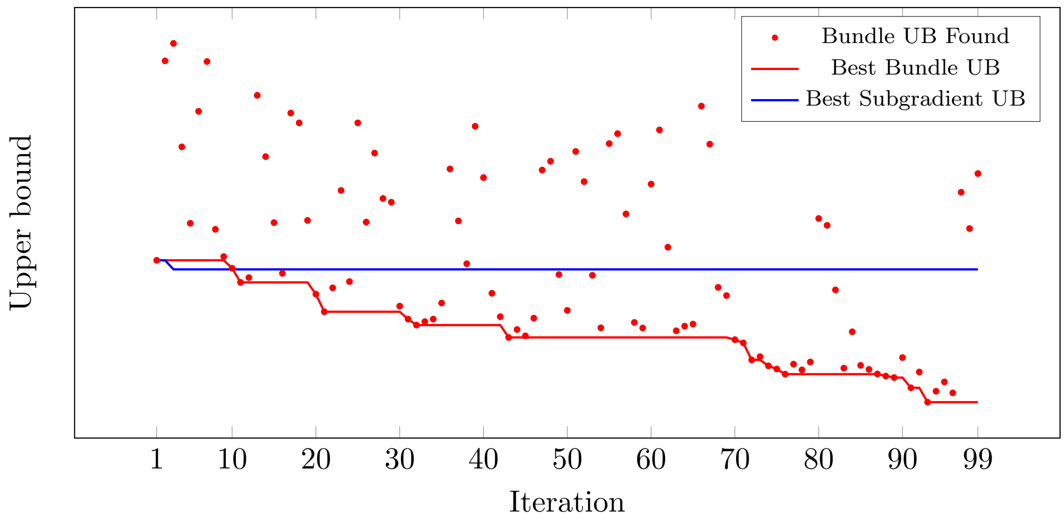


Figure 4.1: Sequence of upper bounds found using bundle methods with dynamic scaling parameter t against evolution of the best upper found using subgradients method (damping factor $\gamma = 0.5$). Smaller γ values were also tried (down to 0.05) but led to the same pattern (only one improvements). Much smaller values (< 0.01) might return better results but were not tried.

The results with these settings are displayed in Table 4.4[15].

Settings	(0.5%, 5min)	(0.1%, 5min)		(0.1%, 3min)
Time limit (around 1 hour)				
Time Elapsed	1h03s	47min	66min	59min
Iterations made	3	1	2	2
End Gap (%)	0.434	0.540	0.482	0.471
Iteration limit (14 iterations)				
Time Elapsed	3h21min	5h09min		4h13min
End Gap (%)	0.382	0.391		0.39

Table 4.4: Results of our algorithm when different optimality gap thresholds or time limits are attempted

Stronger tolerance does not necessarily imply better results in terms of duality gap. Actually, although we already saw that our feasibility recovery method would suggest that a better upper bound commitment, that is the commitment found when solving the dual problem, is more likely to lead us towards a better feasible point, these various settings are enough to show that this is not necessarily the case. Indeed, the results at the end of the first iteration for all three choices, displayed in Table 4.5, show that we can sometimes have a commitment A associated to a better upper bound for our primal problem than

[15]Because a time limit of an hour is not a strict mandatory deadline (an hour and five or six minutes may be accepted, we just do not want to go completely beyond that time goal), we actually display 2 results for the stronger settings, the first found in strictly less than an hour but with an additional 13 minutes available, the other one with 6 minutes beyond the hour, which does not completely overlap our limit.

another commitment B, mix it up with the same feasible commitment, and obtain a worse new feasible commitment in the end for our primal. For example, in this Table, the best recovered solution was found using the second best upper bound commitment, and the worst recovered solution was found using the best upper bound commitment.

Settings	(0.5%, 5min)	(0.1%, 5min)	(0.1%, 3min)
Upper Bound	$4.88900{\cdot}10^8$ (3)	$4.88877{\cdot}10^8$ (1)	$4.88880{\cdot}10^8$ (2)
Lower Bound	$4.864{\cdot}10^8$ (2)	$4.862{\cdot}10^8$ (3)	$4.865{\cdot}10^8$ (1)

Table 4.5: End of first iteration results for different choices of settings regarding optimality gap in branch and bound and time limit of subproblems. The number in parenthesis refers to the rank of a setting for the considered criterion (upper bound or lower bound)

Chapter 5

Algorithm robustness

During all the previous chapters, we have only used a single problem instance, representing the Dutch thermal power plants for a given set of prices $p_t^s, p_t^b, p_{ct}^{\text{fuel}}$, and have shown that our algorithm gives satisfactory results (less than 0.5% duality gap in less than an hour), though not ideal results (according to GDF-Suez, getting a 0.1% optimality guarantee is ideal, and a duality gap inferior to 1% acceptable). The problem is that although we did not really use any choice of parameter to calibrate our model (the only one was the choice of simplifications to reduce the time complexity of the problem), we only tried our algorithm on a single test case. The question now is to examine whether we were not just lucky because of this particular dataset. To this aim, we will test two types of robustness for our algorithm[16], the second being much more demanding than the first one :

- Robustness against new prices : other power selling prices on the network p_t^s and other fuel contract prices p_{ct}^{fuel} are used. In practice, GDF-Suez usually solves this kind of model for a large number of scenarios (because prices are not known in advance, they generate price scenarios and solve the model for each price scenario) with the same power plants and draw conclusions using the different outputs.
- Robustness against new structure : energy companies are usually working in more than a single location, which implies that they are active on multiple power grids, with different sets of power plants and different regulations. In this second robustness test, we will launch the algorithm on power plants located in Singapore. A first short analysis will first be made as we did in section 2.6, followed by the algorithm launched on the test case.

5.1 New prices

After using our algorithm on a few test cases, we realized that globally (this is not a rule that applies to every single price scenario we tried), the smaller the profit margin that can be made by generators when they produce power, the harder the problem becomes to solve. This means that when selling prices are low and contract prices are high, the algorithm will sometimes take a significantly higher amount of time to terminate. Given our run time constraint of one hour, we added a time limit of 20 minutes to two parts of our algorithm : the linear relaxation to get initial dual multipliers and the search for an initial feasible solution. Besides, because our initial feasible solution is mainly used as a feasible basis for our algorithm, we only impose a 10% optimality gap for the algorithm to

[16]With a Polyak step size rule, $\gamma = 0.5$ and the regular settings presented in the previous chapter.

terminate (in addition to the time limit of 20 minutes). Because of the time limit imposed on this initial feasible problem we lose the guarantee to have an initial feasible solution when the iterative part of the algorithm starts, which can clearly pose a problem in the feasibility recovery part of our algorithm, because not having a feasible solution at the end becomes a possibility.

For these robustness tests, we selected 10 scenarios out of the 500 that GDF-Suez gave us, where the selection was made trying to keep a relatively good representation of the price scenario range. The results are shown in Table 5.1[17]. These results can be sorted out into three main categories :

1. Everything went as in our initial problem instance, meaning both a solution and an inferior to 1% duality gap were found within the hour. This behavior can be seen in scenarios 1, 2, 3 and 7 of Table 5.1;

2. The initial feasible solution had an optimality gap (for the simplified problem) of approximately 5%, then the upper bound commitment found and used in the feasibility recovery method either did not help at all (scenario 10) or improved the solution by too small a factor to get a sufficient optimality guarantee (scenario 9);

3. The initial simplified problem was too hard to solve (scenarios 4, 5, 6 and 8), so no feasible solution was found within the 20 minutes available. For this type of behavior, two outcomes were possible :

   (a) The upper bound commitment was feasible at some iteration and the feasibility recovery method used it to get a feasible solution (scenarios 5 and 6), which could lead to good results (5) or not good enough, but not that bad either (6);

   (b) The upper bound commitment was not feasible at any iteration of the algorithm, and the resulting optimality gap exploded because we still had no feasible solution (scenarios 4 and 8);

   For scenario 4, although no admissible solution was found during the hour time limit requested, hence an infinite duality gap, a solution with a 1.26% gap was actually found after 75 minutes and a 0.44% gap after 88 minutes. This may be due to the fact that although the time limit of 20 minutes was reached for the linear relaxation, CPLEX was close to termination hence the values of the dual multipliers returned by CPLEX for this relaxation were close to optimal and thus good enough to retrieve a good feasible commitment in the upper bound computation.

   This pattern did not occur in scenario 8 : the linear relaxation objective value was far from optimal when the time limit was reached, thus the dual multipliers were not good estimators of the optimal ones; this led to an inaccurate representation of the problem constraints, and no feasible solution was found in the iterative part when we let it run for five iterations.

Improvement paths

As we stated before, and in line with the past observations of GDF-Suez, the smaller the profit margin a generator can achieve from producing power, the more the solver will

[17]Note that in this Table, the Initial Feasible row refers to getting an initial feasible schedule and does not include the re-optimization which was not really influenced by the prices and always took around a minute and a half to solve.

Scenario	1	2	3	4	5
Linear relaxation	11min 51s	12min 30s	13min 30s	20min	19min 46s
Initial feasible	8min 10s	9min 28s	9min 32s	20min	20min
Time first iteration	32min15s	33min17s	35min06s	50min39s	53min42s
Gap first iteration	0.269%	0.261%	0.219%	∞	0.864%
Total iterations	3	3	3	1	1
End gap	0.232%	0.258%	0.201%	∞	0.864%
Later feasible				Yes	

Scenario	6	7	8	9	10
Linear Relaxation	16min 27s	13min 18s	20min	12min 49s	13min 50s
Initial Feasible	20min	10min 35s	20min	9min 54s	7min 41s
Time first iteration	55min30s	35min46s	1h02min	30min09s	29min30s
Gap first iteration	2.074%	0.451%	∞	3.780%	3.366%
Total iterations	1	3	1	3	1
End Gap	2.074%	0.360%	∞	2.971%	3.366%
Later feasible			No		

Table 5.1: Results of our algorithm on different price scenarios. For scenario 4 and 8, although CPLEX managed to return a lower bound on the simplified problem, it did not return a feasible solution. Our philosophy here is that because we do not have a feasible solution, we set aside the lower bound returned by the solver and instead act as if it had only returned a $-\infty$ lower bound, hence an infinite optimality gap. Besides, this lower bound was found on the simplified problem and not on the original problem.

struggle to find solutions to the problem. In the rest of this section, we will make an attempt to adapt the algorithm in order to find a feasible solution faster to avoid the issue encountered in scenarios 4 and 8, that is the absence of an initial feasible solution when the iterative part of the algorithm started, and see if this adaptation leads to some improvement. This attempt will thus only modify the part of the algorithm that searches for an initial feasible solution and consists in a shift of every power selling prices p_t^s by a value Δ, in order to get higher selling prices, hence an easier problem according to the hypothesis that smaller prices create harder problems;

$$p_t^s \quad \rightarrow \quad p_t^s + \Delta \qquad\qquad \forall t \in T$$

Since we are only looking at **paths** for an improvement, we will only restrict ourselves to how these modifications change the results for scenario 8, and the impact on results of the other scenarios will not be explored. After we found a Δ that was high enough[18], we launched our algorithm and actually encountered another issue with the time limit of the linear relaxation problem. As opposed to scenario 4 where the linear relaxation solution was not optimal but was close to optimal, the solution found after 20 minutes for the linear relaxation in this case was just not capable to lead to satisfactory results. For the sake of comparison, we increased the time limit for the linear relaxation. The results of this comparison are found in Table 5.2.

Time limit (min)	20	25	30
Linear Relaxation Optimum Reached	No	No	Yes
Gap First Iteration	680.6%	1.631%	1.631%
End Gap	680.3%	1.630%	1.631%

Table 5.2: Variation of the duality gap for scenario 8 when selling prices are boosted up in the initial feasible problem, for different linear relaxation time limits. The gaps shown correspond to the gaps for the normal prices, not for the boosted up prices, which were only used to compute a feasible commitment faster.

5.2 New structure

We now get to the final part of this thesis, which is the last and most demanding robustness test. Although our model and algorithm were built trying to stay as general as possible, the different locations where energy companies are working at have different structures. For example, in the Dutch case we used in the previous pages, generators had no obligation to serve internal demand, hence the production was always entirely sold to the market at price p_t^s. Clearly, this means that we could adapt our algorithm and avoid relaxing the power balance constraints because every generator will produce what it sells and the generator coupling disappears in this set of constraints.

As we mentioned, our aim was to design a general algorithm. We therefore tested it on another case of power plants in Singapore. The main differences with the initial Netherlands test case are the following :

[18]Because the feasible solution is just used as a feasible basis, the choice of Δ is not that important, it just needs to be high enough to make the problem easier to solve.

- We face a nonzero internal demand and energy cannot be sold, or to be more precise, it can be sold but at a zero price, which means producing more than demand is just equivalent to losing money;
- The generators now only have one or two slices instead of five or six;
- There are now at most 3 temperature states instead of at most 7;
- We face no reserve requirements whatsoever;
- More fuel contracts are included in the fuel network and we have more minimum / maximum off-take constraints;

With the observations made in previous section in mind, one can already imagine a problem we are going to face with this instance : an explosion of the solving time because zero selling prices are involved, meaning the profit margin made by generators is negative. Besides, the simplified models we solve are not as different from the initial full model as in the Dutch case because we now only have at most two slices and at most three temperature states, hence the simplifications have a smaller impact in comparison and the number of binaries removed is much smaller. On the other hand, one could argue that since there are fewer temperatures states and slices, the problem should be easier to solve.

The conclusions we can draw after this launch of our algorithm on the Singapore case are that the impact of zero selling prices outweigh that of the smaller number of temperatures states and slices. Indeed,

- The linear relaxation of the problem almost took an hour to solve;
- After 30 minutes, no initial feasible solution were found. However, if we tried to get an initial feasible commitment using the method presented in the previous section, that is with prices boosted up, a feasible commitment was found after 3 minutes (although it did not lead to a good feasible solution but the main objective of price boosting is to obtain an initial feasible solution fast enough);
- The search of a feasible solution if we fed a fixed commitment to the model solved as fast as in the Dutch test case.
- The low prices also made CPLEX struggle in the feasibility recovery part of the algorithm. No new feasible solutions were found during the course of the algorithm because even with a 20 minute time limit for this part of the algorithm, CPLEX still did not manage to finish the first step of its solution computation (root relaxation);
- The generator subproblems, as opposed to the rest of the algorithm, actually behaved quite well and always ended with an inferior to 0.1% gap within 3 minutes.

To finish up with the comments on these last robustness tests, the results from the second bullet seem to confirm that the complexity of this problem is strongly linked to the electricity prices on the market, because our price boosting method actually made CPLEX go from “no feasible solution found in 30 minutes” to “feasible solution found in 5 minutes”.

Conclusion and Perspectives

In this thesis we first presented the construction of an energy market model that tried to replicate the complexity occurring in operational models used by GDF-Suez, followed by the creation of a subgradient-based algorithm to try to obtain good solutions for this model in a time limit of approximately an hour. We then made some robustness tests using this algorithm and other datasets that resulted in some insights on the cause of model complexity as well as some attempts to circumvent this issue. The different observations made during the course of this thesis will first be briefly summed up, followed by some suggestions of future development for this model and the algorithm.

Summary of observations

Linear relaxation Although our choice of initial dual multipliers takes time because we first have to compute the linear relaxation of the full model, this choice always gave quite reasonable results with the Dutch dataset (less than 5% optimality gap) unless

1. the linear relaxation took more than the time limit to solve, in which case the dual multipliers used were only an approximation of what they were supposed to be;

2. the initial feasible solution search did not return any feasible solution before it reached the time limit, but we showed that for both cases when this occurred, a feasible solution found by boosting prices or with feasibility recovery also resulted in a less than 5% gap;

Initial feasible solution Because the model complexity was strongly linked to power selling prices, our algorithm did not always have time to find a feasible solution using the simplified problem. However, when the simplified model was not too hard to be solved efficiently, the feasible solution that was returned was quite good. Besides, when the model was in fact too hard because the profit margin that could be made was too low, we saw that a price boosting method could lead to results that were acceptable.

Upper bound computation This part seems to be the weak point of our algorithm. Even though the initial upper bound found is often quite good, the iterative method used to get better bounds rarely succeeded, except when using bundle methods, but we saw that those methods were actually too slow to be applicable in practice if everything else stays as is.

Feasibility recovery Although the feasibility recovery method never had time to get a solution in the Singapore test-case, it actually often behaved quite well in the Dutch test cases by either giving a feasible solution when we did not have any because the initial feasible solution did not have enough time to terminate, or by improving our current feasible solution available.

Algorithm in general One of the great advantages of our algorithm is the separability of some of its components. For example, the linear relaxation and the simplified initial solution, which we launched one after the other with a 20 minute time limit, could each be launched on two separate machines with a 40 minute time limit because they are completely independent from one another. The same rationale goes for all generator subproblems. We could also decide to do 2 or 3 upper bound computations before applying the feasibility recovery method, to get more options for generators plans when computing a new feasible solution. In addition, if we do not have too many generators, a possible separation in the feasibility recovery would be to try every schedule combination on different machines. This would make the algorithm avoid re-optimization on the full model (because we can immediately try the commitment combinations on the full problem).

Model Complexity The model complexity is strongly linked to the power selling prices or contract prices that are used, whether it is for the linear relaxation, the simplified problems or the generator subproblems. However, using a price boosting method can reduce this complexity to compute solutions faster, keeping in mind that these solutions are optimal for another problem and therefore are mainly useful as a feasible basis.

Future work

Model improvement The model we built was in fact a simplification (in terms of features that are included in the model) of what is really solved by GDF-Suez though its fidelity is much higher since it represents the full horizon of the optimization. A few examples of such additional features that are used are a maximum number of startups for the time horizon, joint minimum/maximum off-take limits on multiple contracts, or the distinction of fuel into *fuel types*, where some power plants can only use certain types of fuel and not others, and fuel type proportions sometimes have to be ensured when electricity is produced. The inclusion of those features into our model could complicate the applicability of our algorithm because, for example, additional constraints would have to be relaxed for example.

Generator subproblems After discussion with GDF-Suez, it turns out that the generators subproblems could be solved very efficiently using dynamic programming. If such a method can indeed be implemented, the upper bound computation could be largely improved because the generator subproblem optimality gap due to branch-and-bound would disappear. These subproblems could be solved within seconds, enabling us to perform much more upper bound iterations before moving on to recover feasible solutions. This would actually also allow us to use bundle methods, which we showed gave better results but needed multiple iterations before getting close to convergence.

Algorithm improvements Because our algorithm can be seen as a combination of three blocks (initialization, iterative upper bound and feasibility recovery), any improvements for one of those blocks (e.g. the iterative upper bound block could be improved using dynamic programming as stated in the previous paragraph) could improve the algorithm as a whole without having to rethink the whole concept behind our method.

Bibliography

[Boy12] Stephen Boyd. Ee364b: Convex optimization ii : Subgradient methods. Course slideshow, Stanford Univertisy, 2012.

[CBFL+09] Santiago Cerisola, Álvaro Baíllo, José M. Fernández-López, Andrés Ramos, and Ralf Gollmer. Stochastic power generation unit commitment in electricity markts: A novel formulation and a comparison of solution methods. *Operations Research, Vol. 57, No. 1*, pages 32–46, 2009.

[Eli] Elia.

[Kle98] Joel B. Klein. The use of heat rates in production cost modeling and market modeling. Technical report, Electricity Analysis Office, California Energy Commission, April 1998.

[Lem00] Claude Lemaréchal. Lagrangian relaxation. *Papers from the Spring School held in Schloß Dagstuhl*, May 2000.

[Ned08] Angelia Nedich. Ie 598 an: Convex optimization : Subgradient projection method. Lecture notes, University of Illinois, 2008.

[Pap13a] Anthony Papavasiliou. Ancillary services, February 2013. LINMA 2415 Quantitative Energy Economics.

[Pap13b] Anthony Papavasiliou. Final exam, June 2013. LINMA 2491 Operations Research.

[SBB10] Christos K. Simoglou, Pandelis N. Biskas, and Anastasios G. Bakirtzis. Optimal self-scheduling of a thermal producer in short-term electricity markets by milp. *IEEE Transactions On power Systems, Vol. 25, No. 4*, November 2010.

[Sch10] Janine Schmidt. Renewable energies and base load power plants : Are they compatible? *Renews Special, Issue 35*, page 6, June 2010. Background information from the German Renewable Energies Agency.

[Sä04] Reine Säljö. Implementation of a bundle algorithm for convex optimization. Master's thesis, Mathematical Sciences, Göteborg University, May 2004.

[TB00] Samer Takriti and John R. Birge. Using integer programming to refine lagrangian-based unit commitment solutions. *IEEE Transactions on Power Systems, Vol. 15, No. 1*, February 2000.

Appendix A

Notations

A.1 Sets

- T : set of time periods $(t_1, \ldots, t_{end})$;
- T_j : subset of time periods $(t_j, \ldots, t_{end})$;
- T^k : subset of time periods $(t_1, \ldots, t_{end-k})$;
- T_j^k : subset of time periods $(t_j, \ldots, t_{end-k})$;
- G : set of generators;
- M_g : set of modes associated to a generator g with Off mode included $(\forall g \in G)$;
- M_g^0 : set of modes associated to a generator g without Off mode $(\forall g \in G)$;
- A_g : set of possible transitions between modes of generator g;
- Θ_{gm} : set of possible temperature states for generator g in working mode m;
- S_{gm} : set of production slices associated to generator g in working mode m;
- MR_g : set of time periods when generator g must be on
- MS_g : set of time periods when generator g must be off
- G^{SUP} : set of generators with startup profiles;
- G^{SDP} : set of generators with shutdown profiles;
- M_g^{SUP} : set of working modes with startup profiles for generator g;
- M_g^{SDP} : set of working modes with shutdown profiles for generator g;
- Θ_g^{SUP} : set of temperature states with startup profiles for generator g in working mode m;
- Θ_g^{SDP} : set of temperature states with shutdown profiles for generator g in working mode m;
- C : set of contracts;
- FD : set of delivery spots for fuel;

- L : set of links between a contract and a delivery spot;
- PL : set of links between two delivery spots;
- $TF(c)$ set of timeframes over which minimum or maximum off-takes constraints exists for contract c;
- Pool(d) : subset of generators located at delivery spot d;

A.2 Parameters

- p_t^s : price for selling a unit of power to the market at time t;
- p_b^t : price for buying a unit of power from the market at time t;
- p_{ct}^{fuel} : price of a unit of fuel, taken from contract c at time t;
- VOM_{gm} : hourly maintenance cost for generator g in working mode m;
- TC_{ga} : fuel consumption associated to transition a for generator g;
- $p_t^{CO_2}$: price of CO_2 emission rights at instant t;
- E_{gm} : CO_2 emission rate of generator g running in mode m;
- UT_g : minimum up-time for generator g;
- DT_g : minimum down-time for generator g;
- ut_{gm} : minimum up-time for mode m in generator g;
- dt_{gm} : minimum down-time for mode m in generator g;
- τ_{gml}^{SUP} : time length of startup profile for generator g in working mode m at temperature state l;
- $\tilde{\tau}_{gm}^{SUP}$: minimum time length over all temperature states of startup profiles for generator g in working mode m;
- τ_{gm}^{SDP} : time length of shutdown profile for generator g in working mode m;
- $\overline{\tau}_{gml}$: time after which generator g can begin starting up in working mode m with temperature state l when it is shut down;
- $\underline{\tau}_{gml}$: time after which generator g cannot start up anymore in working mode m with temperature state l when it is shut down;
- D_t : demand at time t;
- P_{gms}^{+} : maximum output for production slice s for generator g in working mode g;
- $P_{gm}^{\pm}$: minimum/maximum output for generator g in working mode g;
- P_{gmly}^{SUP} : production level for the $y-$th hour of startup profile, for generator g in working mode m when the startup occurred in temperature state l;

- P_{gmy}^{SDP} : production level for the $y-$th hour of shutdown profile, for generator g in working mode m;
- R_{gm}^{+} : ramping up parameter for generator g in working mode m;
- R_{gm}^{-} : ramping down parameter for generator g in working mode m;
- M a large number;
- $\underline{O}_{c,\tau}$: minimum off-take of fuel for contract c for timeframe τ;
- $\overline{O}_{c,\tau}$: maximum off-take of fuel for contract c for timeframe τ;
- SUC_{gml} : startup consumption for generator g in working mode m when the startup occurs in temperature state l;
- NLC_{gm} : no-load consumption for generator g in working mode m;
- HR_{gms} : heat-rate for production slice s, associated to working mode m from generator g;
- MCR_{gmt}^{2+} : maximum contribution to upward secondary reserves by generator g working in mode m at instant t;
- MCR_{gmt}^{2-} : maximum contribution to downward secondary reserves by generator g working in mode m at instant t;
- MCR_{gmt}^{3} : maximum contribution to tertiary reserves by generator g working in mode m at instant t;
- $RR_{t}^{2\pm}$: upward/downward secondary reserves requirements for time t;
- RR_{t}^{3} : tertiary reserves requirements for time t;

A.3 Variables

- s_{gt} : amount of energy sold by generator g to the market at instant t;
- b_t : amount of energy bought from the market at instant t;
- o_{ct} : off-take of fuel from contract c at instant t;
- u_{gmt} : generator g in working mode m or not g at instant t;
- v_{gat} : occurrence of transition a for generator g at instant t;
- f_{gmt} : quantity of fuel used by generator g in working mode m at instant t;
- u_{gt} : on state of generator g at instant t;
- v_{gt} : startup of generator g at instant t;
- z_{gt} : shutdown of generator g at instant t;
- v_{gmt} : startup of generator g in working mode m at instant t;

- u_{gmt}^{SUP} : occurence of a startup profile for generator g in working mode m at instant t;
- u_{gmt}^{DISP} : absence of startup or shutdown profile for generator g in working mode m at instant t;
- u_{gmt}^{SDP} : occurrence of a shutdown profile for generator g in working mode m at instant t;
- v_{gmlt} : startup of generator g in working mode m occurring in temperature state l at instant t;
- p_{gt} : energy production of generator g at instant t;
- p_{gmt} : energy production of generator g in working mode m at instant t;
- p_{gmst} : energy production in segment s of generator g in working mode m at instant t;
- u_{gmst} : authorization to produce in segment s for generator g in working mode m at instant t;
- p_{gmt}^{SUP} : energy production at time t if generator g in working mode m is following a startup profile;
- p_{gmt}^{SDP} : energy production at time t if generator g in working mode m is following a shutdown profile;
- q_{lt} : fuel transiting between a contract and a delivery spot using connexion l at instant t;
- $\tilde{q}_{ht}$: fuel transiting between two delivery spots using pipeline h at instant t;
- f_{gt} : amount of fuel used by generator g at instant t;
- $r_{gmt}^{2\pm}$: contribution of generator g in working mode m at instant t to upward/downward secondary reserves;
- r_{gmt}^{3S} : contribution of active generator g in mode m at instant t to tertiary reserves;
- r_{gmt}^{3NS} : contribution of inactive generator g in mode m at instant t to tertiary reserves;
- u_{gmt}^{3NS} : possibility of contribution to non-spinning tertiary reserves from generator g in working mode m at instant t;

A.4 Operators

- $F(a)$: Origin mode of transition a;
- $T(a)$: Destination mode of transition a;
- $F^*(l)/F(h)$: Origin spot/contract of fuel connexion l/h;
- $T^*(l)/T(h)$: Destination spot of fuel connexion l/h;
- $\mathrm{Spot}(g)$: Delivery spot where generator g is located;

Appendix B

Full Model

$$
\begin{aligned}
\max \quad & \sum_{t\in T}\Bigg(p^s_t \sum_{g\in G} s_{gt} - p^b_t b_t - \sum_{c\in C} p^{\text{fuel}}_{ct} o_{ct} - \sum_{g\in G}\Bigg(\sum_{m\in M^0_g} VOM_{gm} u_{gmt} \\
& \qquad\qquad + \sum_{a\in A_g} TC_{ga} v_{gat} + p^{CO_2}_t \sum_{m\in M_g} E_{gm} f_{gmt}\Bigg)\Bigg) \\
\text{s.t.} \quad & u_{gt} = u_{g,t-1} + v_{gt} - z_{gt} && \forall g\in G, t\in T_1 \\
& u_{gt} = 1 && \forall g\in G, t\in MR_g \\
& u_{gt} = 0 && \forall g\in G, t\in MS_g \\
& 1 = \sum_{m\in M_g} u_{gmt} && \forall g\in G, t\in T \\
& u_{gt} = \sum_{m\in M^0_g} u_{gmt} && \forall g\in G, t\in T \\
& u_{gmt} = u_{gm,t-1} + \sum_{\substack{a\in A_g \\ T(a)=m}} v_{gat} - \sum_{\substack{a\in A_g \\ F(a)=m}} v_{gat} && \forall g\in G, m\in M_g, t\in T \\
& 1 \geq \sum_{a\in A_g} v_{gat} && \forall g\in G, t\in T \\
& v_{gmt} = v_{gat} && \forall g\in G, m\in M^0_g, a=(\text{Off}, m), t\in T \\
& v_{gt} = \sum_{\substack{a\in A_g \\ F(a)=\text{Off}}} v_{gat} && \forall g\in G, t\in T \\
& u_{gt} \geq \sum_{y=t-UT_g+1}^{t} v_{gy} && \forall g\in G, t\in T_{UT_g} \\
& u_{gt} \leq 1 - \sum_{y=t-DT_g+1}^{t} z_{gy} && \forall g\in G, t\in T_{DT_g}
\end{aligned}
$$

$$u_{gmt} \geq \sum_{y=t-ut_{gm}+1}^{t} \sum_{\substack{a \in A \\ T(a)=m}} v_{gay} \qquad \forall g \in G, m \in M_g, t \in T_{ut_{gm}}$$

$$u_{gmt} \leq 1 - \sum_{y=t-dt_{gm}+1}^{t} \sum_{\substack{a \in A \\ F(a)=m}} v_{gay} \qquad \forall g \in G, m \in M_g, t \in T_{dt_{gm}}$$

$$u_{gmt} = u_{gmt}^{SUP} + u_{gmt}^{DISP} + u_{gmt}^{SDP} \qquad \forall g \in G, m \in M_g^0, t \in T$$

$$u_{gmt}^{SUP} = \sum_{l \in \Theta_{gm}^{SUP}} \sum_{\substack{y=t-\tau_{gml}^{SUP}+1 \\ y \geq 1}}^{t} v_{gmly} \qquad \forall g \in G^{SUP}, m \in M_g^{SUP}, t \in T_{\tilde{\tau}_{gm}^{SUP}}$$

$$u_{gmt}^{SDP} = \sum_{y=t}^{t+\tau_{gm}^{SDP}-1} v_{gat} \qquad \forall g \in G^{SDP}, m \in M_g^{SDP}, a = (m, \mathrm{Off})$$

$$t \in T^{\tau_{gm}^{SDP}+1}$$

$$v_{gmt} = \sum_{l \in \Theta_{gm}} v_{gmlt} \qquad \forall g \in G, m \in M_g^0, t \in T$$

$$v_{gmlt} \leq \sum_{y=t-\underline{\tau}_{gml}+1}^{t-\overline{\tau}_{gml}} z_{gy}, \qquad \forall g \in G, m \in M_g^0, l \in \Theta_{gm}, t \in T_{\underline{\tau}_{gml}}$$

$$0 = \sum_{g \in G} p_{gt} + b_t - D_t - \sum_{g \in G} s_{gt} \qquad \forall t \in T$$

$$0 \leq b_t \qquad \forall t \in T$$

$$b_t \leq D_t \qquad \forall t \in T$$

$$s_{gt} \geq 0 \qquad \forall g \in G, t \in T$$

$$s_{gt} \leq p_{gt} \qquad \forall g \in G, t \in T$$

$$p_{gt} = \sum_{m \in M_g^0} p_{gmt} \qquad \forall g \in G, t \in T$$

$$p_{gmt} = \sum_{s \in S_{gm}} p_{gmst} \qquad \forall g \in G, m \in M_g^0, t \in T$$

$$p_{gmst} \geq 0 \qquad \forall g \in G, m \in M_g^0, s \in S_{gm}, t \in T$$

$$p_{gmst} \leq P_{gms}^{+} u_{gmst} \qquad \forall g \in G, m \in M_g^0, s \in S_{gm}, t \in T$$

$$p_{gmt}^{SUP} = \sum_{l \in \Theta_{gm}^{SUP}} \sum_{\substack{y=t-\tau_{gml}^{SUP}+1 \\ y \geq 1}}^{t} v_{gmly} P_{gml,t-y+1}^{SUP} \qquad \forall g \in G^{SUP}, m \in M_g^{SUP}, t \in T_{\tilde{\tau}_{gm}^{SUP}}$$

$$p_{gmt}^{SDP} = \sum_{y=t+1}^{t+\tau_{gml}^{SDP}} v_{gat} P_{gm,y-t}^{SDP} \qquad \forall g \in G^{SDP}, m \in M_g^{SDP}, a = (m, \mathrm{Off})$$

$$t \in T^{\tau_{gm}^{SDP}}$$

$$
\begin{aligned}
p_{gmt} &\geq p^{SUP}_{gmt} + p^{SDP}_{gmt} + P^{-}_{gm} u^{DISP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T \\
p_{gmt} &\leq p^{SUP}_{gmt} + p^{SDP}_{gmt} + P^{+}_{gm} u^{DISP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T \\
u_{gms_1t} &= u_{gmt} && \forall g \in G, m \in M_g, t \in T \\
u_{gm,s+1,t} &\leq \frac{p_{gmst}}{P^{+}_{gms}} && \forall g \in G, m \in M^0_g, s \in S_{gm} \backslash \{s^{end}_{gm}\}, t \in T \\
p_{gmt} &\leq p_{gm,t-1} + R^{+}_{gm} + M \sum_{\substack{a \in A_g \\ T(a)=m}} v_{gat} \\
&\qquad + M u^{SUP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T_1 \\
p_{gmt} &\geq p_{gm,t-1} - R^{-}_{gm} - M \sum_{\substack{a \in A_g \\ F(a)=m}} v_{gat} \\
&\qquad - M u^{SDP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T_1 \\
o_{ct} &= \sum_{\substack{l \in L \\ F^*(l)=c}} q_{lt} && \forall c \in C, t \in T \\
0 &= \sum_{\substack{g \in G \\ g \in \mathrm{Pool}(d)}} f_{gt} + \sum_{\substack{h \in PL \\ F^{**}(l)=d}} \tilde{q}_{ht} - \sum_{\substack{l \in L \\ T^*(l)=d}} q_{lt} \\
&\qquad - \sum_{\substack{h \in PL \\ T^{**}(l)=d}} \tilde{q}_{ht} && \forall d \in FD, t \in T \\
\underline{O}_{c\tau} &\leq \sum_{y \in \tau} o_{cy}, && \forall c \in C, \tau \in TF(c) \subseteq T \\
\overline{O}_{c\tau} &\geq \sum_{y \in \tau} o_{cy}, && \forall c \in C, \tau \in TF(c) \subseteq T \\
f_{gmt} &= \sum_{l \in \Theta_{gm}} v_{gmlt} SUC_{gml} + NLC_{gm} u_{gmt} \\
&\quad + \sum_{\substack{a \in A_g \\ T(a)=m}} TC_{ga} v_{gat} + \sum_{s \in S_{gm}} HR_{gms} p_{gmst} \\
f_{gt} &= \sum_{m \in M^0_g} f_{gmt} && \forall g \in G, t \in T \\
r^{2+}_{gmt} &\geq 0 && \forall g \in G, m \in M^0_g, t \in T \\
r^{2-}_{gmt} &\geq 0 && \forall g \in G, m \in M^0_g, t \in T \\
r^{2+}_{gmt} &\leq p^{SUP}_{gmt} + p^{SDP}_{gmt} + P^{+}_{gm} u^{DISP}_{gmt} - p_{gmt} && \forall g \in G, m \in M^0_g, t \in T \\
r^{2-}_{gmt} &\geq p_{gmt} - p^{SUP}_{gmt} - p^{SDP}_{gmt} - P^{-}_{gm} u^{DISP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T \\
r^{2+}_{gmt} &\leq MCR^{2+}_{gmt} u^{DISP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T \\
r^{2-}_{gmt} &\leq MCR^{2-}_{gmt} u^{DISP}_{gmt} && \forall g \in G, m \in M^0_g, t \in T
\end{aligned}
$$

$$
\begin{aligned}
RR_t^{2+} &\leq \sum_{g\in G}\sum_{m\in M_g^0} r_{gmt}^{2+} && \forall t \in T\\
RR_t^{2-} &\leq \sum_{g\in G}\sum_{m\in M_g^0} r_{gmt}^{2-} && \forall t \in T\\
r_{gmt}^{3S} &\geq 0 && \forall g \in G, m \in M_g, t \in T\\
r_{gmt}^{3S} &\leq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{+} u_{gmt}^{DISP} - p_{gmt} - r_{gmt}^{2+} && \forall g \in G, m \in M_g^0, t \in T\\
u_{gmt}^{3NS} &\leq 1 - u_{gt} && \forall g \in G, m \in M_g^0, t \in T\\
1 &\geq \sum_{m\in M_g} u_{gmt}^{3NS} && \forall g \in G, t \in T\\
u_{gmt}^{3NS} &= 0 && \forall g \in G, m \in M_g^0, t \in MS_g\\
r_{gmt}^{3NS} &\geq 0 && \forall g \in G, m \in M_g^0, t \in T\\
r_{gmt}^{3NS} &\leq P_{gm}^{+} u_{gmt}^{3NS} && \forall g \in G, m \in M_g^0, t \in T\\
MCR_{gmt}^{3} &\geq r_{gmt}^{3S} + r_{gmt}^{3NS} && \forall g \in G, m \in M_g^0, t \in T\\
RR_t^{3} &\leq \sum_{g\in G}\sum_{m\in M_g^0} (r_{gmt}^{3S} + r_{gmt}^{3NS}) && \forall t \in T
\end{aligned}
$$

Appendix C

Subproblems

C.1 Trivial subproblem

$$
\begin{aligned}
\max \quad & \sum_{t \in T} \left(-p^b_t b_t - \lambda^{PB}_t (b_t - D_t) - \lambda^{R2+}_t RR^{2+}_t - \lambda^{R2-} RR^{2-}_t - \lambda^{R3}_t RR^3_t \right) \\
\text{s.t.} \quad & 0 \le b_t && \forall t \in T \\
& b_t \le D_t && \forall t \in T
\end{aligned}
$$

C.2 Network subproblem

$$
\begin{aligned}
\max \quad & \sum_{t \in T} \left[-\sum_{c \in C} p^{\text{fuel}}_{ct} o_{ct} - \sum_{d \in FD} \lambda^{FE}_{dt} \left(-\sum_{\substack{l \in L \\ T^*(l)=d}} q_{lt} + \sum_{\substack{h \in PL \\ F^{**}(l)=d}} \tilde{q}_{ht} - \sum_{\substack{h \in PL \\ T^{**}(l)=d}} \tilde{q}_{ht} \right) \right] \\
\text{s.t.} \quad & o_{ct} = \sum_{\substack{l \in L \\ F^*(l)=c}} q_{lt} && \forall c \in C, t \in T \\
& \underline{O}_{c\tau} \le \sum_{y \in \tau} o_{cy}, && \forall c \in C, \tau \in TF(c) \subseteq T \\
& \overline{O}_{c\tau} \ge \sum_{y \in \tau} o_{cy}, && \forall c \in C, \tau \in TF(c) \subseteq T
\end{aligned}
$$

C.3 Generator g subproblem

$$\max \sum_{t\in T}\Bigg[p_t^s s_{gt} - \sum_{m\in M_g^0} VOM_{gm}u_{gmt} - \sum_{a\in A_g} TC_{ga}v_{gat} - p_t^{CO_2}\sum_{m\in M_g^0} E_{gm}f_{gmt}$$

$$-\lambda_t^{PB}(p_{gt}-s_{gt}) + \lambda_{\text{Spot}(g),t}^{FE} f_{gt}$$

$$-\sum_{g\in G}\sum_{m\in M_g^0}\quad(\lambda_t^{R2+}r_{gmt}^{2+} + \lambda_t^{R2-}r_{gmt}^{2-} + \lambda_t^{R3}(r_{gmt}^{3S}+r_{gmt}^{3NS}))\Bigg]$$

$$\begin{aligned}
\text{s.t.}\quad & u_{gt} = u_{g,t-1} + v_{gt} - z_{gt} && \forall g\in G, t\in T_1\\
& u_{gt} = 1 && \forall g\in G, t\in MR_g\\
& u_{gt} = 0 && \forall g\in G, t\in MS_g\\
& 1 = \sum_{m\in M_g} u_{gmt} && \forall g\in G, t\in T\\
& u_{gt} = \sum_{m\in M_g^0} u_{gmt} && \forall g\in G, t\in T\\
& u_{gmt} = u_{gm,t-1} + \sum_{\substack{a\in A_g\\ T(a)=m}} v_{gat} - \sum_{\substack{a\in A_g\\ F(a)=m}} v_{gat} && \forall g\in G, m\in M_g, t\in T\\
& 1 \geq \sum_{a\in A_g} v_{gat} && \forall g\in G, t\in T\\
& v_{gmt} = v_{gat} && \forall g\in G, m\in M_g^0, a=(\text{Off}, m), t\in T\\
& v_{gt} = \sum_{\substack{a\in A_g\\ F(a)=\text{Off}}} v_{gat} && \forall g\in G, t\in T\\
& u_{gt} \geq \sum_{y=t-UT_g+1}^{t} v_{gy} && \forall g\in G, t\in T_{UT_g}\\
& u_{gt} \leq 1 - \sum_{y=t-DT_g+1}^{t} z_{gy} && \forall g\in G, t\in T_{DT_g}\\
& u_{gmt} \geq \sum_{y=t-ut_{gm}+1}^{t}\sum_{\substack{a\in A\\ T(a)=m}} v_{gay} && \forall g\in G, m\in M_g, t\in T_{ut_{gm}}\\
& u_{gmt} \leq 1 - \sum_{y=t-dt_{gm}+1}^{t}\sum_{\substack{a\in A\\ F(a)=m}} v_{gay} && \forall g\in G, m\in M_g, t\in T_{dt_{gm}}\\
& u_{gmt} = u_{gmt}^{SUP} + u_{gmt}^{DISP} + u_{gmt}^{SDP} && \forall g\in G, m\in M_g^0, t\in T\\
& u_{gmt}^{SUP} = \sum_{l\in\Theta_{gm}^{SUP}}\ \sum_{\substack{y=t-\tau_{gml}^{SUP}+1\\ y\geq 1}}^{t} v_{gmly} && \forall g\in G^{SUP}, m\in M_g^{SUP}, t\in T_{\tilde{\tau}_{gm}^{SUP}}
\end{aligned}$$

$$u_{gmt}^{SDP} = \sum_{y=t}^{t+\tau_{gm}^{SDP}-1} v_{gat} \qquad \forall g \in G^{SDP}, m \in M_g^{SDP}, a = (m, \mathrm{Off})$$
$$t \in T^{\tau_{gm}^{SDP}+1}$$
$$v_{gmt} = \sum_{l \in \Theta_{gm}} v_{gmlt} \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$v_{gmlt} \leq \sum_{y=t-\underline{\tau}_{gml}+1}^{t-\overline{\tau}_{gml}} z_{gy}, \qquad \forall g \in G, m \in M_g^0, l \in \Theta_{gm}, t \in T_{\underline{\tau}_{gml}}$$
$$s_{gt} \geq 0 \qquad \forall g \in G, t \in T$$
$$s_{gt} \leq p_{gt} \qquad \forall g \in G, t \in T$$
$$p_{gt} = \sum_{m \in M_g^0} p_{gmt} \qquad \forall g \in G, t \in T$$
$$p_{gmt} = \sum_{s \in S_{gm}} p_{gmst} \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$p_{gmst} \geq 0 \qquad \forall g \in G, m \in M_g^0, s \in S_{gm}, t \in T$$
$$p_{gmst} \leq P_{gms}^{+} u_{gmst} \qquad \forall g \in G, m \in M_g^0, s \in S_{gm}, t \in T$$
$$p_{gmt}^{SUP} = \sum_{l \in \Theta_{gm}^{SUP}} \sum_{\substack{y=t-\tau_{gml}^{SUP}+1 \\ y \geq 1}}^{t} v_{gmly} P_{gml,t-y+1}^{SUP} \qquad \forall g \in G^{SUP}, m \in M_g^{SUP}, t \in T_{\tilde{\tau}_{gm}^{SUP}}$$
$$p_{gmt}^{SDP} = \sum_{y=t+1}^{t+\tau_{gml}^{SDP}} v_{gat} P_{gm,y-t}^{SDP} \qquad \forall g \in G^{SDP}, m \in M_g^{SDP}, a = (m, \mathrm{Off})$$
$$t \in T^{\tau_{gm}^{SDP}}$$
$$p_{gmt} \geq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{-} u_{gmt}^{DISP} \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$p_{gmt} \leq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{+} u_{gmt}^{DISP} \qquad \forall g \in G, m \in M_g^0, t \in T$$
$$u_{gms_1t} = u_{gmt} \qquad \forall g \in G, m \in M_g, t \in T$$
$$u_{gm,s+1,t} \leq \frac{p_{gmst}}{P_{gms}^{+}} \qquad \forall g \in G, m \in M_g^0, s \in S_{gm} \backslash \{s_{gm}^{end}\}, t \in T$$
$$p_{gmt} \leq p_{gm,t-1} + R_{gm}^{+} + M \sum_{\substack{a \in A_g \\ T(a)=m}} v_{gat} + M u_{gmt}^{SUP} \qquad \forall g \in G, m \in M_g^0, t \in T_1$$
$$p_{gmt} \geq p_{gm,t-1} - R_{gm}^{-} - M \sum_{\substack{a \in A_g \\ F(a)=m}} v_{gat} - M u_{gmt}^{SDP} \qquad \forall g \in G, m \in M_g^0, t \in T_1$$

$$
\begin{aligned}
o_{ct} &= \sum_{\substack{l \in L \\ F^*(l)=c}} q_{lt} && \forall c \in C, t \in T \\
&\qquad - \sum_{\substack{h \in PL \\ T^{**}(l)=d}} \tilde{q}_{ht} && \forall d \in FD, t \in T \\
f_{gmt} &= \sum_{l \in \Theta_{gm}} v_{gmlt} SUC_{gml} + NLC_{gm} u_{gmt} \\
&\quad + \sum_{\substack{a \in A_g \\ T(a)=m}} TC_{ga} v_{gat} + \sum_{s \in S_{gm}} HR_{gms} p_{gmst} \\
f_{gt} &= \sum_{m \in M_g^0} f_{gmt} && \forall g \in G, t \in T \\
r_{gmt}^{2+} &\geq 0 && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{2-} &\geq 0 && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{2+} &\leq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{+} u_{gmt}^{DISP} - p_{gmt} && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{2-} &\geq p_{gmt} - p_{gmt}^{SUP} - p_{gmt}^{SDP} - P_{gm}^{-} u_{gmt}^{DISP} && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{2+} &\leq MCR_{gmt}^{2+} u_{gmt}^{DISP} && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{2-} &\leq MCR_{gmt}^{2-} u_{gmt}^{DISP} && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{3S} &\geq 0 && \forall g \in G, m \in M_g, t \in T \\
r_{gmt}^{3S} &\leq p_{gmt}^{SUP} + p_{gmt}^{SDP} + P_{gm}^{+} u_{gmt}^{DISP} - p_{gmt} - r_{gmt}^{2+} && \forall g \in G, m \in M_g^0, t \in T \\
u_{gmt}^{3NS} &\leq 1 - u_{gt} && \forall g \in G, m \in M_g^0, t \in T \\
1 &\geq \sum_{m \in M_g} u_{gmt}^{3NS} && \forall g \in G, t \in T \\
u_{gmt}^{3NS} &= 0 && \forall g \in G, m \in M_g^0, t \in MS_g \\
r_{gmt}^{3NS} &\geq 0 && \forall g \in G, m \in M_g^0, t \in T \\
r_{gmt}^{3NS} &\leq P_{gm}^{+} u_{gmt}^{3NS} && \forall g \in G, m \in M_g^0, t \in T \\
MCR_{gmt}^{3} &\geq r_{gmt}^{3S} + r_{gmt}^{3NS} && \forall g \in G, m \in M_g^0, t \in T
\end{aligned}
$$