



HAL
open science

Modélisation et résolution de grands problèmes stochastiques combinatoires, application à deux problèmes de gestion de production d'électricité

Nicolas Dupin

► **To cite this version:**

Nicolas Dupin. Modélisation et résolution de grands problèmes stochastiques combinatoires, application à deux problèmes de gestion de production d'électricité. Informatique [cs]. Université Lille 1 - Sciences et Technologies, 2015. Français. NNT: . tel-02463285

HAL Id: tel-02463285

<https://hal.science/tel-02463285v1>

Submitted on 31 Jan 2020

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

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

THÈSE

pour obtenir le grade de

Docteur de Lille 1

Spécialité : **Informatique**

préparée au laboratoire **CRISTAL**

dans le cadre de l'École Doctorale **Sciences Pour l'Ingénieur (SPI)**

présentée et soutenue publiquement
par

Nicolas DUPIN

le 5 octobre 2015

Titre:

**Modélisation et résolution de grands problèmes stochastiques combinatoires:
applications à la gestion de production d'électricité**

Directeur de thèse: **TALBI El-Ghazali**

Jury

Prof. HANAFI Saïd ,	Président du jury
Prof. DAUZERE-PERES Stéphane ,	Rapporteur
Prof. YALAOUI Farouk ,	Rapporteur
Dr. BENOIST Thierry ,	Examineur
LCL. BAZOT Pierre,	Invité
M. FLEISCHER Alex,	Invité
Prof. TALBI El-Ghazali ,	Directeur de thèse

L'université n'émet aucune approbation ni improbation aux opinions émises dans les thèses: ces opinions doivent être considérées comme propres à l'auteur. De même, les propos émis dans ce manuscrit ne relèvent pas de prises de positions officielles d'EDF R&D ou de la DGA, mais du travail personnel de l'auteur.

À un camarade polytechnicien corse vêtu de noir qui me ressemblait comme un frère,
À un mathématicien génial qui aurait été un thésard exceptionnel,
À un poète disparu trop tôt et à qui je pense souvent,
Arnaud, je ne t'oublierai jamais.

Voi dai nemici nostri,
A noi date vittoria
E poi l'eterna gloria
In paradisu.

Prolégomène

Aude sapere

Memento audere semper

We all have a great need for acceptance, but you must trust that your beliefs are unique, your own, even though others may think them odd or unpopular, even though the herd may go, that's baaaad. Robert Frost said, "Two roads diverged in the wood and I, I took the one less traveled by, and that has made all the difference."

Tom Schulman, *Dead Poet Society*

Victory has thousand fathers, but defeat is an orphan.

John Fitzgerald Kennedy,

Je forme une entreprise qui n'eut jamais d'exemple et dont l'exécution n'aura point d'imitateur. Je veux montrer à mes semblables un homme dans toute la vérité de la nature; et cet homme ce sera moi. Moi seul. Je sens mon cœur et je connais les hommes. Je ne suis fait comme aucun de ceux que j'ai vus; j'ose croire n'être fait comme aucun de ceux qui existent. Si je ne vaux pas mieux, au moins je suis autre. Si la nature a bien ou mal fait de briser le moule dans lequel elle m'a jeté, c'est ce dont on ne peut juger qu'après m'avoir lu. Que la trompette du Jugement dernier sonne quand elle voudra, je viendrai, ce livre à la main, me présenter devant le souverain juge. Je dirai hautement: "Voilà ce que j'ai fait, ce que j'ai pensé, ce que je fus. J'ai dit le bien et le mal avec la même franchise. Je n'ai rien tu de mauvais, rien ajouté de bon, et s'il m'est arrivé d'employer quelque ornement indifférent, ce n'a jamais été que pour remplir un vide occasionné par mon défaut de mémoire; j'ai pu supposer vrai ce que je savais avoir pu l'être, jamais ce que je savais être faux. [...] Être éternel, rassemble autour de moi l'innombrable foule de mes semblables; qu'ils écoutent mes confessions, qu'ils gémissent de mes indignités, qu'ils rougissent de mes misères. Que chacun d'eux découvre à son tour son cœur aux pieds de ton trône avec la même sincérité; et puis qu'un seul te dise, s'il l'ose: "Je fus meilleur que cet homme-là."

Jean-Jacques Rousseau, *Les Confessions*

Table des matières

Notations	7
Glossaire	9
Résumé	11
Abstract	12
Remerciements	13
Introduction	15
I Etat de l'art	17
1 PLNE, modélisation et résolution	19
1.1 Programmation Linéaire en Nombres Entiers (PLNE)	19
1.1.1 Définitions et premiers exemples	19
1.1.2 Programmation Linéaire	21
1.1.3 Coupes d'intégrité	23
1.1.4 Résolution générique d'un PLNE : algorithme de Branch&Bound	24
1.1.5 Résolutions spécifiques	25
1.2 Décomposition de Benders	26
1.2.1 Génération de coupes	27
1.2.2 Reformulation de Benders en Programmation Linéaire mixte	27
1.2.3 Cas partiellement bloc diagonal	29
1.2.4 Points d'implémentation	30
1.2.5 Cas des sous-problèmes entiers	32
1.3 Formulation de Dantzig-Wolfe et génération de colonne	33
1.3.1 Génération de colonne	33
1.3.2 Reformulation de Dantzig-Wolfe	34
1.3.3 Bornes lagrangiennes et décomposition de Dantzig Wolfe	36
1.3.4 Cas partiellement bloc diagonal	36
1.3.5 Résolution en nombres entiers	37
1.4 Optimisation linéaire dans l'incertain	37
1.4.1 Programmation Linéaire Stochastique avec recours	38
1.4.2 Robustesse en Programmation Linéaire	38
1.4.3 Approches robustes sur des problèmes discrets	40
1.4.4 Approches robustes multi-étapes	41

1.4.5	Autres approches robustes en ordonnancement	41
1.5	Heuristiques	42
1.5.1	Heuristiques constructives	42
1.5.2	Recherche locale	43
1.5.3	Métaheuristiques de populations	45
1.5.4	Heuristiques hybrides	46
1.5.5	Heuristiques et méthodes exactes en PLNE	46
2	Gestion de production électrique	49
2.1	Contexte national de production d'électricité	49
2.1.1	Demandes en électricité	49
2.1.2	Parc de production national	51
2.1.3	Processus décisionnels et problèmes d'optimisation	52
2.2	Unit Commitment Problem	53
2.2.1	Modélisation en PLNE de l'Unit Commitment Problem	53
2.2.2	Résolution de problèmes de type UCP, état de l'art	54
2.2.3	Contraintes dynamiques	55
2.3	UCPd un UCP discret pour un parc thermique	56
2.3.1	Description du problème UCPd	56
2.3.2	Formulations PLNE	56
2.3.3	Jeux de données	59
2.3.4	Etat de l'art de la résolution PLNE	59
2.4	Planification des arrêts des réacteurs nucléaires	60
2.4.1	Définition du problème du challenge ROADEF	60
2.4.2	Jeux de données du challenge ROADEF	62
2.4.3	État de l'art du challenge ROADEF	63
2.4.4	Etat de l'art de la résolution à EDF	67
II	Problème de planification journalière de la production	69
3	Résolution exacte d'UCPd en PLNE	71
3.1	Renforcements de la formulation par niveaux	71
3.1.1	Formulation renforcée par regroupement sur des cliques	71
3.1.2	Couplages entre variables binaires et démarrages	72
3.1.3	Formulation renforcée	73
3.2	Comparaison polyédrale et renforcements	73
3.2.1	Transformation des variables d'états en variables de paliers	73
3.2.2	Comparaison polyédrale des formulations	74
3.3	Reformulation étendue de Dantzig Wolfe	76
3.3.1	Formulation étendue pour UCPd	76
3.3.2	Problème maître restreint	76
3.3.3	Génération de colonne	77
3.4	Résultats d'implémentation	77
3.4.1	Relaxation linéaire des formulations compactes	78
3.4.2	Convergence Branch&Bound des formulations compactes	78
3.4.3	Résultats de la formulation étendue	80

3.5	Conclusions et perspectives	81
4	Matheuristiques dérivées pour UCPd	83
4.1	Heuristiques constructives	83
4.1.1	Heuristiques par fixation de variables	83
4.1.2	Approches de construction hiérarchique	84
4.1.3	Approche constructive parallélisable de type POPMUSIC	85
4.1.4	Résultats	85
4.2	Amélioration de solutions par descente VNS	86
4.2.1	L'algorithme général de descente VNS par PLNE	86
4.2.2	Définition des voisinages pour UCPd	88
4.2.3	Résultats	90
4.3	Synthèse des résultats et perspectives	91
III	Planification des arrêts nucléaires pour maintenances et rechargements	93
5	Formulations PLNE compactes pour le challenge ROADEF	95
5.1	Définition des variables et de la fonction objectif	95
5.1.1	Définition des variables	96
5.1.2	Considérations de planning initial	98
5.1.3	Fonction objectif	99
5.2	Ecriture des contraintes	99
5.2.1	Contraintes sur les niveaux de production	99
5.2.2	Contraintes liées aux quantités de combustible	99
5.2.3	Contraintes d'ordonnancement sur les dates d'arrêts	100
5.2.4	Gestion des contraintes de stretch CT6	102
5.3	Bilan des formulations	104
5.3.1	Formulation sans stretch	104
5.3.2	Formulation avec bornes supérieures du Stretch	105
5.3.3	Formulation sur un cas particulier de stretch	105
6	Résolution exacte et bornes duales du challenge ROADEF	107
6.1	Résultats de résolution frontale exacte	107
6.1.1	Preprocessing exact des fenêtres de temps	107
6.1.2	Résultats de la formulation sans stretch	108
6.1.3	Résultats de la formulation avec stretch	109
6.1.4	Influence de la solution initiale	110
6.1.5	Bilan des résultats de formulation frontale	111
6.2	Bornes duales pour le challenge ROADEF	111
6.2.1	Bornes duales par agrégation des pas de temps	112
6.2.2	Bornes duales en décomposant par scénarios	113
6.2.3	Bornes inférieures par relaxation de contraintes	116
6.2.4	Résultats	117
6.3	Conclusions et perspectives	119

7 Résolutions déterministes heuristiques	121
7.1 Heuristiques constructives	121
7.1.1 Fixation de variables sur la relaxation continue	121
7.1.2 Formulation sans variables de production nucléaire	122
7.1.3 Algorithmes de type gloutons/Relax and Fix	123
7.1.4 Résultats des approches constructives	125
7.2 Amélioration de solutions par descente VNS	126
7.2.1 Déploiement et enjeux des VNS avec voisinages PLNE	126
7.2.2 Définition des voisinages	126
7.2.3 Résultats des voisinages séparés	126
7.2.4 Résultats de l'approche VNS	128
7.3 Synthèse des résultats et perspectives	129
8 Planification robuste des arrêts nucléaires	131
8.1 Considérations de robustesse dans le modèle	131
8.1.1 Modélisation des aléas	131
8.1.2 Écriture générique des problèmes déterministes	132
8.1.3 Schéma robuste à 2 étapes	132
8.1.4 Formulation mixte robuste et stochastique	134
8.2 Décomposition de Benders déployée sur le modèle robuste	136
8.2.1 Contraintes d'ordonnancement en environnement robuste	136
8.2.2 Réécriture du modèle	137
8.2.3 Problème maître	138
8.2.4 Coupe de Benders sans stretch	139
8.3 Formulations "robustifiées"	140
8.3.1 Formulations de faisabilité d'ordonnancement robustes	140
8.3.2 Anticipations maximales robustes	141
8.3.3 Paramétrisation de la robustesse des contraintes CT14 et CT15	142
8.4 Résultats, conclusions et perspectives	143
IV Conclusions et perspectives	147
Conclusions	149
Perspectives ouvertes par la thèse	151
V Annexes	153
Résultats annexes des chapitres 5 à 7	155
Communications et publications réalisées dans le cadre de la thèse	157
Vie de thésard	159
Bibliographie	161

Notations

Notations mathématiques générales

$\llbracket a, b \rrbracket$	Ensemble des entiers dans l'intervalle $[a, b]$, c'est $[a, b] \cap \mathbb{Z}$.
$\mathbb{1}_A$	Fonction indicatrice de A dans $\{0, 1\}$, $\mathbb{1}_A(x) = 1$ ssi $x \in A$.
$\exp(x)$	Fonction exponentielle $\exp(x) = e^x$
\mathcal{K}	Ensemble quelconque, pouvant être $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ ou \mathbb{R}_+ .
\mathcal{K}^*	$\mathcal{K}^* = \mathcal{K} - \{0\}$.
\mathcal{K}_+	$\mathcal{K}_+ = \{k \in \mathcal{K}, k \geq 0\}$.
\mathcal{K}^p	Ensemble des vecteurs de dimension $p \in \mathbb{N}$ à coefficients dans \mathcal{K} .
$\mathcal{M}_{m,n}(\mathcal{K})$	Ensemble des matrices à $m \in \mathbb{N}$ lignes et $n \in \mathbb{N}$ colonnes à coefficients dans \mathcal{K} .
$\mathbb{N} = \mathbb{Z}_+$	Ensemble des nombres entiers positifs.
\mathbb{R}	Ensemble des nombres réels.
z^+	$z^+ = \max(0, z)$ pour $z \in \mathbb{R}$.

Notations communes pour UCP et UCPd

\mathcal{U}	Ensemble des unités de production.
\mathcal{T}	Ensemble des pas de temps. On note $\mathcal{T} = \llbracket 1, T \rrbracket$, $t = 0$ est l'instant initial
l_u	Durée minimale d'arrêt de l'unité u .
L_u	Durée minimale de fonctionnement de l'unité u .
D_t^P	Demande (prévisionnelle) de puissance à la date t .
D_t^{R1}	Demande de réserve primaire à la date t .
D_t^{R2}	Demande de réserve secondaire à la date t .
C_u^{dem}	Coût de démarrage de l'unité u .
C_u^{fix}	Coût de production fixe lorsque l'unité u est active.
C_u^{pprp}	Coût de production proportionnel à la puissance produite de l'unité u .

Notations spécifiques pour UCPd

\mathcal{I}_u	Ensemble des points de fonctionnement de la tranche u . (0 pour un arrêt).
N_u	Nombre de points de fonctionnement de l'unité u .
$i_u^{(0)}$	Point de fonctionnement initial, avant l'optimisation.
$P_{u,i}$	Puissance produite par l'unité u au point de fonctionnement i .
$R_{u,i}^1$	Capacité maximale en réserve primaire de l'unité u au point de fonctionnement i .
$R_{u,i}^2$	Capacité maximale en réserve secondaire de l'unité u au point de fonctionnement i .
$\Delta_{u,i}^+$	Durée minimale au point i de l'unité u avant une montée de puissance.
$\Delta_{u,i}^-$	Durée minimale au point i de l'unité u avant une baisse de puissance.

Notations pour le problème de planification des rechargements nucléaires

Ensembles et indexations

$t \in \mathcal{T} = \llbracket 1, T \rrbracket$	Ensemble des pas de temps, chaque période correspondant à $[t, t + 1]$
$w \in \mathcal{W} = \llbracket 1, W \rrbracket$	Ensemble des semaines de l'horizon d'optimisation.
$j \in \mathcal{J} = \llbracket 1, J \rrbracket$	Ensemble des moyens de production dits de type I).
$i \in \mathcal{I} = \llbracket 1, I \rrbracket$	Ensemble des réacteurs nucléaires (moyens de production de type II).
$k \in \mathcal{K}_i = \llbracket 0, K_i - 1 \rrbracket$	Ensemble des cycles d'un réacteur nucléaire i .
$s \in \mathcal{S} = \llbracket 1, S \rrbracket$	Ensemble des scénarios stochastiques du challenge ROADEF.
$\delta \in \Delta$	Ensemble des scénarios de prolongation d'arrêt
$m \in \mathcal{M}_{i,k} = \llbracket 1, \mathbf{Np}_{i,k} \rrbracket$	Ensemble des points de la courbe de stretch du réacteur i pour le cycle (i, k)

Notations temporelles

$\mathbf{Dem}^{t,s}$	Demande en puissance au pas de temps t pour le scénario s
\mathbf{D}_t	amplitude de temps entre les pas de temps t et $t + 1$
t_w	Premier pas de temps de la semaine w
w_t	Semaine auquel appartient le pas de temps t

Notations pour toute unité thermique j (type 1)

\mathbf{C}_{jt}^s	Coût de production proportionnel à la puissance produite à t au scénario s .
\mathbf{Pmin}_{jt}^s	Puissance minimale au pas de temps t et au scénario s .
\mathbf{Pmax}_{jt}^s	Puissance maximale au pas de temps t et au scénario s .

Notations pour toute unité nucléaire i (type 2)

$\mathbf{Amax}_{i,k}$	Anticipation maximale pour démarrer le cycle $k + 1$.
$\mathbf{Bo}_{i,k}$	Niveau de fuel au Bore nul du cycle k
\mathbf{C}_i	Valorisation dans l'objectif des niveaux de combustible finaux.
$\mathbf{c}_{i,k,m}$	Ordonnée des points de la courbe de profil stretch au mode m , ccycle k .
$\mathbf{Da}_{i,k}$	Nombre de semaines nominal de l'arrêt du cycle k .
$\overline{\mathbf{Da}}_{i,k}$	Nombre de semaines de l'arrêt du cycle k en cas de prolongation.
$\mathbf{f}_{i,k,m}$	Abscisse des points de la courbe de profil stretch au mode m , ccycle k .
$\mathbf{Mmax}_{i,k}$	Modulation maximale globale sur le cycle de production k .
\mathbf{Pmax}_i^t	Puissance maximale au pas de temps t .
$\mathbf{Q}_{i,k}$	Coefficient de pertes en rechargement au cycle k .
\mathbf{Rmin}_{ik}	Rechargement minimal à l'arrêt du cycle k .
\mathbf{Rmax}_{ik}	Rechargement maximal à l'arrêt du cycle k .
$\mathbf{Smax}_{i,k}$	Niveau maximal de combustible sur le cycle de production k .
\mathbf{Xi}_i	Niveau de combustible initial.

Glossaire

ACO	Algorithme des colonies de fourmis
ASN	Autorité de Sûreté Nucléaire
ASR	Arrêt pour Simple Rechargement
B&B	Branch&Bound
Cplex	Solveur de PLNE développé par Ilog.
CSP	Cutting Stock Problem, problème de découpe de rouleaux de papier
CSP	Constraint Satisfaction Problem, problème de satisfaction de contraintes
EDF	Electricité de France.
EJP	Effacements Jours de Pointe
EURO	Association Européenne de Recherche Opérationnelle.
FP	Feasibility Pump
GRASP	Greedy Randomized Adaptive Search Procedure
GUB	Generalized Upper Bound, contraintes de type $\sum x_i \leq 1$.
INRIA	Institut National de Recherche en Informatique et en Automatique
JPOC	Journées Polyèdres, Optimisation Combinatoire
OPL	Modeleur et interface de Cplex développée par IBM Ilog.
ORION	Algorithme de placement des arrêts nucléaires en exploitation à EDF
PL	Programmation Linéaire ou Programme Linéaire.
PLNE	Programmation (ou Programme) Linéaire en Nombres Entiers.
PAN	Plannification des Arrêts Nucléaires (pour maintenances et rechargement)
PMR	Problème Maître Restreint.
POPMUSIC	Partial Optimization Metaheuristic under Special Intensification Conditions
PPC	Programmation Par Contraintes.
PSO	Particle Swarm Optimization, algorithme par essaims particulaires
Matheuristique	Heuristique utilisant la programmation mathématique
RINS	Relaxation Induced Neighborhood Search, heuristique pour la PLNE.
RTE	Réseau de Transport d'Electricité.
RO	Recherche Opérationnelle.
ROADEF	Association Française de Recherche Opérationnelle et d'Aide à la Décision.
SOS	Special Ordered Sets.
Stretch	Courbe de production décroissante nucléaire en fin de cycle
TANNAT	Modèle de résolution PLNE développé à EDF R&D pour PAN
UCP	Unit Commitment Problem.
UCPd	Unit Commitment Problem avec des puissances de productions discrètes.
VND	Variable Neighborhood Descent, variante déterministe de VNS
VNS	Variable Neighborhood Search
VRPTW	Vehicle Routing Problem with Time Windows

Résumé

Modélisation et résolution de grands problèmes stochastiques combinatoires : application à la gestion de production d'électricité

Le formalisme de la Programmation Linéaire en Nombres Entiers (PLNE) est couramment utilisé dans le monde industriel pour modéliser des problèmes d'optimisation complexes. Ce succès provient de la facilité à modéliser globalement des problèmes d'optimisation avec de nombreux jeux de contraintes et de variables décisionnelles dont des variables discrètes. La résolution générique en PLNE par l'algorithme de Branch&Bound permet de concentrer le travail sur les questions de modélisation, pour une résolution de type boîte noire. Malgré les importants progrès réalisés par la PLNE ces dernières décennies, la résolution est souvent limitée pour des problèmes de taille réelle. Les méthodes heuristiques sont alors utilisées pour trouver des solutions de bonne qualité sans preuve d'optimalité, ni même de majoration de l'écart à la solution optimale.

Il était question d'étudier les limites de la résolution exacte et des heuristiques sur des problèmes industriels d'optimisation d'EDF R&D à modéliser en PLNE, en vue de leur insertion dans le processus décisionnel opérationnel. L'application principale concerne la planification des arrêts de maintenance et de rechargement des centrales nucléaires, qui a fait l'objet du Challenge EURO/ROADEF 2010. Nous avons aussi traité un problème discrétisé de production journalière d'un parc thermique à flamme. Ces deux problèmes d'applications ont des structures et des caractéristiques de résolution frontale Branch&Bound très différentes.

La méthodologie suivie est analogue pour les deux cas. On modélise tout d'abord le problème avec une formulation compacte PLNE, pour en analyser les limites de la résolution frontale par Branch&Bound. Des méthodes de décomposition peuvent alors être étudiées. On dérive ensuite les méthodes exactes en matheuristiques pour résoudre des instances de taille réelle. Dans cette optique, l'hybridation de Variable Neighborhood Search (VNS) avec des voisinages définis par PLNE a donné des résultats très probants sur les deux problèmes en termes de qualités de solutions.

Le fait d'avoir travaillé avec des méthodes exactes a permis également de chiffrer l'impact d'hypothèses de résolution, de répondre à des considérations opérationnelles, mais également d'obtenir des bornes inférieures. Pour le problème du Challenge EURO/ROADEF 2010, nous avons pu obtenir des bornes qui amélioreraient les meilleurs bornes de l'état de l'art.

Mots-clés : Recherche Opérationnelle, Programmation Linéaire en Nombres Entiers (PLNE), Programmation Mathématique, Matheuristiques, Variable Neighborhood Search (VNS), Optimisation sous incertitude, Gestion de production d'électricité, Unit Commitment Problem, Planification des maintenances et des rechargements des centrales nucléaires, Challenge EURO/ROADEF 2010.

Abstract

MILP Modelling and resolution of industrial stochastic and combinatorial optimization problems : application to energy management problems

Mixed Integer Linear Programming (MILP) is a very popular and useful framework to model industrial optimization problems. This success is due to the facility to model complex optimization problems, with numerous sets of constraints and variables, including discrete variables. The work can be focused on modeling, with a black box generic resolution to optimality with Branch&Bound (B&B) algorithm, or with a specialized decomposition algorithm. If MILP made lots of progresses on the last decades, it is often not sufficient to tackle real world size instances. In such cases, heuristic methods are commonly used to find good quality solutions, without any guarantee to reach the optimum and any proven bound to the optimum.

Our work focus on two complex industrial problems from energy management optimization, with different structures and resolution characteristics. First application is a discretized daily Unit Commitment Problem of thermal units with specific dynamic constraints. Second application comes from the EURO/ROADEF 2010 challenge, scheduling problem of nuclear power plants' outages for maintenances and refueling.

In both cases the methodology was first to model efficiently the considered problem with a MILP compact formulation, and analyze the frontal resolution's limits with B&B. Decomposition methods could also be investigated, before the exact methods are derived in a matheuristic, to be able to tackle real size instances. In particular, Variable Neighborhood Search (VNS) with MILP neighbourhoods gave outstanding results on our problems. Our work allowed to answer natural modeling questions, estimating the impacts of usual hypothesis. Furthermore, we derived dual bounds for these optimizations problems.

Keywords : Operations Research, Optimization, Mathematical Programming, Mixed Integer Linear Programming (MILP), Matheuristics, Variable Neighborhood Search(VNS), Unit Commitment Problem, Maintenance scheduling of Nuclear Outages, Challenge EURO/ROADEF 2010.

Remerciements

Mes remerciements s'adressent tout d'abord à mon directeur de thèse El-Ghazali Talbi et à l'université de Lille 1. Grâce à vous, ma thèse a pu se finaliser, merci de m'avoir soutenu et fait confiance tout au long du projet, dans les moments fastes comme dans les difficultés. Merci au personnel administratif de l'université pour leur accueil chaleureux et leur aide dans l'accomplissement de démarches administratives à distance. Je remercie les membres du jury pour la qualité de leur relecture, qui m'a permis d'améliorer la qualité de ce manuscrit.

Je remercie la DGA de m'avoir permis de réaliser ce projet qui me tenait à cœur, en me détachant pour me permettre d'approfondir les méthodes d'optimisation. Ce parcours a été bénéfique pour moi, au-delà du côté technique, la thèse est une aventure humaine dont on ressort grandi. Plus personnellement, je remercie tout ceux qui ont été à l'écoute d'un tel parcours. Merci à Audrey Hermant-Ledoux, ma marraine à la DGA, d'avoir toujours été de bons conseils pour ma thèse et mes choix de parcours à la DGA, je suis très fier d'avoir une marraine avec un tel parcours académique. Merci à Pierre-Edouard Adenot et Jérôme de Dinechin pour leur soutien dans les moments difficiles de ma thèse, merci de m'avoir remis en selle alors que je pensais abandonner. Merci à Pierre Bazot pour avoir finalisé l'encadrement de ma thèse et permis d'enclencher une dynamique collaborative avec le CATOD.

Pendant mes années de thèse, j'ai beaucoup apprécié les événements organisés par la ROADEF, l'association Française de Recherche Opérationnelle et d'Aide à la Décision, une communauté familiale, où l'ouverture d'esprit de ses membres a permis des échanges gratifiants et bénéfiques pour l'avancement de mes réflexions. J'ai particulièrement apprécié les communautés plus spécialisées POC (Polyèdre et Optimisation Combinatoire) et META pour les métaheuristiques, et leurs congrès bisannuels associés. J'adresse des remerciements particuliers à Frédéric Gardi, Nabila Remli, Amel Benhamiche avec qui j'ai eu le plaisir d'avoir des échanges académiques de qualité, démarrés lors de tels congrès et poursuivis par correspondance.

J'adresse une pensée particulière à ceux qui m'ont transmis la passion de la Recherche Opérationnelle, en plus des savoir-faire en optimisation et en informatique. Ils m'ont conduit à vouloir approfondir dans le domaine et à choisir la voie de la thèse. Un grand merci à Stéphane Gaubert, Frédéric Bonnans, Frédéric Meunier, pour leur enseignement à l'École Polytechnique. Merci à Francis Sourd et Yohan Firmy pour ce qu'ils m'ont appris lorsque j'étais en stage à la SNCF. Merci à Marie-Christine Costa, Maurice Diamantini et Olivier Klopfenstein pour leur enseignement à l'ENSTA. Merci à Tomas Simovic et Pascale Bendotti pour leur encadrement lorsque j'étais en stage de fin d'études à EDF R&D, pour l'intelligence de leur management, qui a conduit à de fructueux résultats et mes meilleurs moments à EDF R&D.

Avec mon profil de mathématicien, je remercie les informaticiens qui m'ont épaulé pour mon codage, mes validations numériques et plus généralement pour la maîtrise indispensable de l'informatique. Je remercie particulièrement Romain Leguay pour son aide précieuse en programmation à Bordeaux et Alex Fleischer et l'équipe d'IBM pour leur aide sur Cplex. Je dois beaucoup de mes résultats à l'utilisation d'OPL qui m'a permis de me concentrer sur le travail de modélisation mathématique en me libérant de difficultés d'implémentation. J'adresse mes remerciements et mon profond respect pour la communauté des développeurs bénévoles et dévoués du logiciel libre, ayant utilisé Linux Ubuntu et les logiciels libres

pendant toute ma thèse. Merci aussi à Donald Knuth et Leslie Lamport pour $\text{T}_{\text{E}}\text{X}$ et $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Je tiens à adresser ma gratitude à tous ceux qui m'ont donné le goût des maths bien avant ma spécialisation en Recherche Opérationnelle, le travail en amont est rarement mis en avant à sa juste valeur. Merci à mon institutrice de CM1 au Lycée Français de Bruxelles, Mme Poncet, d'avoir éveillé en moi le goût des mathématiques. Merci à Serge Francinou, mon modèle en enseignement, de m'avoir révélé l'esthétique des mathématiques, durant une année mémorable en HX3 au Lycée Henri IV. Merci enfin à celle qui m'a transmis le souci de la rigueur mathématique depuis tout petit et qui jusqu'au bout a tenu à apporter sa pierre à ma relecture de thèse, mon professeur de mathématiques de toujours, ma maman.

Je remercie tous ceux qui m'ont soutenu et encouragé pendant ma thèse, dans le milieu de la recherche et les communautés de doctorants, ou mes amis non doctorants qui ont dû composer avec mes disponibilités de ministre entre Paris, Lille et Bordeaux. Merci Romain, Gaëlle, Baptiste, Frédéric, Jean-Christophe, Amel, Matthieu, Hubert, Jérémy, Adriana. Je n'oublie pas l'association des doctorants de mathématiques de Bordeaux LAMBDA, dont j'ai eu l'honneur d'être vice-président, avec l'organisation de soirées festives nécessaires sur un travail à long terme pour savoir décompresser. Merci à Lorie pour sa positive attitude, et à Günther pour ses chansons ding ding dong qui ont égayé mes soirées.

J'adresse mes remerciements à toute ma famille pour leur soutien indéfectible et inconditionnel qui m'a aidé à surmonter mes difficultés. J'adresse particulièrement ma gratitude à mon père d'avoir tracé ma voie. Je n'aurais sans doute pas eu un tel parcours académique sans son ascension sociale. Issu d'un milieu modeste, étant le premier de sa famille à rentrer au lycée et intégrer l'École Polytechnique, une telle histoire n'était pas improbable du temps où l'ascenseur social fonctionnait.

Merci à Virginie, l'élue de mon coeur. Ce n'est pas facile de vivre avec un thésard en fin de thèse, merci pour ton Amour qui m'a donné des ailes et la force de terminer.

Introduction

Cette thèse a été initiée sous l'impulsion d'EDF R&D, pour combiner des méthodes de résolution exacte de programmation mathématique (INRIA Bordeaux) avec des techniques de résolution heuristique (INRIA Lille), sur des problèmes d'applications d'EDF R&D. Il en résultait un partenariat EDF/INRIA Lille/INRIA Bordeaux pour un montage CIFRE de thèse industrielle. Avec l'accord de la DGA, j'ai pu m'engager dans ce projet comme "formation par la recherche", pour un montage DGA/EDF/INRIA où la DGA a la charge de mon financement tandis qu'EDF R&D finançait le contrat d'accompagnement avec l'INRIA, et mes déplacements dans le cadre de la thèse.

L'application majeure plébiscitée par EDF R&D était le problème de planification des arrêts de maintenance et de rechargement des centrales nucléaires. La thèse faisait suite à un travail de modélisation pour le challenge EURO/ROADEF 2010. Le besoin opérationnel n'était pas complètement défini et bien que le problème du challenge ROADEF comporte suffisamment de difficultés, avec divers jeux de variables et de contraintes. Les objectifs de modélisation d'EDF R&D étaient encore plus ambitieux.

Le but premier de cette thèse était d'étendre la formulation du challenge ROADEF en intégrant les problématiques de robustesse liées à des aléas de prolongation d'arrêts. Le besoin opérationnel préconisait des solutions robustes vis à vis des contraintes d'ordonnancement. En outre, la réoptimisation mensuelle des plannings de rechargements nucléaires engendrait un besoin de stabilité du planning au fil du temps.

La volonté d'EDF était de se baser sur des approches de résolution exacte. Le choix des méthodes de résolution pour la formulation PLNE du problème était une question à mûrir, les approches utilisées à EDF étant principalement des méthodes d'optimisation continue telles que la décomposition par les prix.

Les méthodes de modélisation PLNE et d'hybridations avec des métaheuristiques étant génériques, il s'agissait d'étudier dans quelle mesure ces méthodes pouvaient s'appliquer à d'autres problématiques d'EDF modélisables en PLNE. Le problème journalier d'optimisation de la production électrique en est un cadre adapté. Cette problématique ayant fait l'objet de mon projet de fin d'étude à EDF R&D dans les cas du parc thermique à flammes, cette modélisation a ainsi été réutilisée. Une de mes motivations initiales pour approfondir ce travail était d'implémenter un schéma de génération de colonne sur ce problème.

Cette thèse est organisée en quatre parties. La première partie introduit le contexte académique de Recherche Opérationnelle et le contexte national de la gestion de production d'électricité. La seconde partie s'applique sur un problème de l'optimisation journalière du parc thermique à flammes, analysant tout d'abord la résolution exacte par PLNE et décomposition, avant de dériver les formulations PLNE en matheuristiques. La troisième partie s'applique au problème de planification des arrêts de maintenance et de rechargement des centrales nucléaires, suivant la même méthodologie. De plus, nous avons étudié sur ce problème l'introduction de considérations de robustesse. La dernière partie détaille les conclusions et apports de cette thèse, ainsi que les perspectives ouvertes pour la suite.

Première partie

Etat de l'art

Chapitre 1

PLNE, modélisation et résolution

Résumé : Ce chapitre réintroduit les bases de modélisation mathématique et de résolution nécessaires à la compréhension des développements de la thèse. Le formalisme de la Programmation Linéaire en Nombres Entiers est présenté en développant les aspects théoriques et pratiques de la résolution. Les techniques de décomposition de Benders et de Dantzig-Wolfe sont développées avec leurs algorithmes de résolution exacte par génération de coupes et de colonnes. Pour les problématiques d'optimisation sous incertitude, nous présentons les formalismes de la programmation stochastique et de l'optimisation robuste. Enfin, nous présentons les approches heuristiques, en insistant sur leur potentiel d'hybridation avec des approches exactes.

La Recherche Opérationnelle (RO) se définit comme l'ensemble des méthodes rationnelles traitant de problèmes décisionnels du monde réel. En particulier, les formalismes d'optimisation sous contraintes permettent de modéliser des problématiques décideurs confrontés à des prises de décisions face à un grand nombre d'informations, de contraintes, et d'aléas, que la puissance informatique traite plus aisément que des opérateurs humains. Ce chapitre réintroduit les bases de RO nécessaires pour la suite de la thèse.

1.1 Programmation Linéaire en Nombres Entiers (PLNE)

Le cadre de la Programmation Linéaire en Nombres Entiers (PLNE) est couramment utilisé pour modéliser et résoudre des problèmes industriels. Un tel succès provient d'une part de la facilité à modéliser des problèmes d'optimisation complexes, et d'autre part de l'existence d'algorithmes efficaces de résolution. Cette section introduit le cadre de la PLNE, dans l'optique de définir un cadre théorique mais aussi pratique, pour donner les clés d'une résolution efficace de problèmes modélisés en PLNE.

1.1.1 Définitions et premiers exemples

La PLNE traite des problèmes qui s'écrivent mathématiquement de la forme suivante :

$$\begin{aligned} \min_{x \in \mathbb{N}^m \times \mathbb{R}_+^p} \quad & cx \\ \text{s.c. :} \quad & Ax = b \\ & Mx \geq d \end{aligned} \tag{1.1}$$

Le vecteur décisionnel x comprend des décisions discrètes et des variables continues. On lui associe un coût linéaire $c \in \mathbb{R}^{m+p}$. L'ensemble de réalisabilité est décrit par des contraintes linéaires d'égalité et d'inégalité, représentées par les matrices $A \in \mathcal{M}_{n,m+p}(\mathbb{R})$, $M \in \mathcal{M}_{n',m+p}(\mathbb{R})$ et des seconds membres $b \in \mathbb{R}^n$, $d \in \mathbb{R}^{n'}$. Dans le cas où toutes les variables x sont continues, on parle de Programme Linéaire (PL). Dans le cas où l'ensemble de réalisabilité défini par $Ax = b$ et $Mx \geq d$ est borné et non vide, il est compact et l'existence d'une solution atteignant le minimum global est assurée.

On note que l'on peut se ramener à l'écriture (1.1) par transformations d'écriture. Pour des variables $x \leq 0$, on utilise le changement de variable $y = -x \geq 0$. Pour des variables x non signées, on définit $y^{(+)}, y^{(-)} \geq 0$ et on remplace les occurrences de x par $y^{(+)} - y^{(-)}$. Pour un problème de maximisation, on se ramène à une minimisation en considérant $(-c)x$ comme fonction objectif. Les contraintes d'égalité $Ax = b$ peuvent s'écrire dupliquées $Ax \leq b$ et $(-A)x \leq -b$. Inversement, les contraintes d'inégalités $Mx \geq d$ s'écrivent $Mx = d + s$ en rajoutant des variables $s \geq 0$. De nombreux problèmes quadratiques se linéarisent en PLNE, nous référons à [5], [6]. Les contraintes logiques et/ou/implication se linéarisent également en PLNE, nous référons à [71]. Nous référons également à [66] pour une introduction pédagogique à la modélisation PLNE.

Le cadre de la PLNE permet de modéliser de nombreux problèmes complexes d'optimisation, ce qui a contribué à sa popularité dans le monde industriel. On pourra se reporter à [24] pour des applications industrielles de la PLNE. L'exemple canonique de la PLNE est le *problème de sac à dos* : disposant de N objets, chaque objet i ayant une masse m_i et une valeur c_i , d'un sac à dos de contenance maximale M , il s'agit de remplir le sac à dos en maximisant la valeur totale. Cela donne la formulation PLNE suivante :

$$\begin{aligned} & \max_{x \in \{0,1\}^N} \sum_{i=1}^N c_i x_i \\ \text{s.c : } & \sum_{i=1}^N m_i x_i \leq M \end{aligned} \tag{1.2}$$

Définition 1.1.1 (Borne primale, solution primale) *Tout $x \in \mathbb{N}^m \times \mathbb{R}_+^p$ vérifiant les contraintes, ie tel que $A.x = b$ et $M.x \geq d$, est appelé solution primale. La valeur $c.x$ est alors un majorant de la solution optimale (avec l'écriture en tant que problème de minimisation), on parle génériquement de borne primale.*

Définition 1.1.2 (Borne duale) *Tout majorant (resp minorant) de la solution optimale d'un problème de maximisation (resp minimisation) est appelée borne duale.*

Les bornes primales s'obtiennent en exhibant des solutions réalisables, ce qui peut être obtenu à l'aide des méthodes heuristiques, nous référons à la section 1.5. Les bornes duales s'obtiennent par des relaxations du problème, et nécessitent des preuves mathématiques.

Définition 1.1.3 (Polyèdre, Polytope) *Un polyèdre est une intersection d'un nombre fini de demi-espaces affines fermés. En d'autres termes $P \subset \mathbb{R}^n$ est un polyèdre s'il existe un entier $p \in \mathbb{N}$ et une matrice de taille $p \times n$, $b \in \mathbb{R}^p$ tels quel $P = \{x \in \mathbb{R}^n, A.x \leq b\}$. Un polyèdre est donc un sous-ensemble fermé de \mathbb{R}^n . Un polytope est un polyèdre borné.*

Les polyèdres interviennent en PLNE, les équations de contraintes $A.x \leq b$ ou $A.x = b$ définissant des polyèdres. La fonction objectif $c.x$ définit par le vecteur c une direction d'optimisation. Les lignes de niveau de même valeur $c.x$ sont des hyperplans, et comme illustré sur la figure 1.1, l'optimum d'un PL est atteint soit en un point unique, point extrême du polyèdre, soit sur toute une facette. A fortiori un point extrême est solution, minimiser sur un polyèdre, ensemble continu et non dénombrable, revient à minimiser sur nombre fini de points, les points extrêmes du polyèdre. Lorsque des variables discrètes interviennent, les solutions optimales peuvent être à l'intérieur strict du polyèdre défini par les équations, point extrême du polyèdre correspondant l'enveloppe convexe des solutions entières.

Définition 1.1.4 (Enveloppe convexe) *Soit $A \subset \mathbb{R}^n$. L'enveloppe convexe de A est le plus petit convexe de \mathbb{R}^n contenant A : $\text{conv}(A) = \bigcap_{A \subset B \subset \mathbb{R}^n} B$. $\text{conv}(A)$ est l'ensemble des combinaisons convexes de familles finies d'éléments de A : $\text{conv}(A) = \{x \in \mathbb{R}^n | x = \sum_{i=1}^m \lambda_i a_i\}$ avec $m \in \mathbb{N}$ et des familles $(a_1, \dots, a_m) \in A^m$ et $(\lambda_1, \dots, \lambda_m) \in \mathbb{R}_+^m$ telles que $\sum_i \lambda_i = 1$. L'enveloppe convexe d'un nombre fini de points est un polytope.*

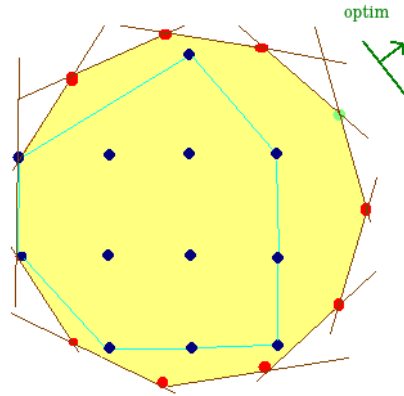


FIGURE 1.1 – Point de vue géométrique d'un programme linéaire

1.1.2 Programmation Linéaire

Un cas particulier majeur de la PLNE est le cas de la Programmation Linéaire (PL), ne comprenant aucune variable discrète. Dans ce cas, le calcul de la solution (ou la preuve de non existence de solution réalisable) est un problème polynomial, la solution se calcule ex nihilo en temps polynomial.

Dualité en programmation linéaire On associe à tout PL, son PL dual selon les règles suivantes :

Programme primal, version symétrique

$$\begin{aligned} \max_{x \geq 0} c^T \cdot x \\ \text{s.c : } A \cdot x \leq b \end{aligned} \quad (1.3)$$

Programme primal, version asymétrique

$$\begin{aligned} \min_{y \geq 0} c \cdot y \\ \text{s.c : } B \cdot y = d \end{aligned} \quad (1.5)$$

Programme dual, version symétrique

$$\begin{aligned} \min_{y \geq 0} b^T \cdot y \\ \text{s.c : } A^T \cdot y \geq c \end{aligned} \quad (1.4)$$

Programme dual, version asymétrique

$$\begin{aligned} \max_{x \in \mathbb{R}^n} d^T \cdot x \\ \text{s.c : } B^T \cdot x \leq c^T \end{aligned} \quad (1.6)$$

Par transformation d'écritures, on peut toujours se ramener à une de ces écritures, suivant les transformations qui ont déjà été mentionnées à la section 1.1.

Théorème 1.1.1 (Théorème de dualité faible) *La valeur du programme dual est toujours inférieure à celle du programme primal.*

Théorème 1.1.2 (Théorème de dualité forte) *Les propriétés suivantes sont équivalentes :*

- (i) les programme primal et dual sont réalisables.
- (ii) le programme primal a une solution.
- (iii) le programme dual a une solution.

Dans ce cas, les valeurs des programmes dual et primal sont égales.

Algorithmes de résolution d'un PL La première résolution efficace et générique d'un PL a été l'*algorithme du simplexe*, utilisant que les solutions d'un PL sont des points extrêmes du polyèdre défini par les contraintes. Il s'agit d'un procédé de descente en parcourant ces points extrêmes partant d'un point extrême initial ([33]). Cet algorithme est exponentiel dans de rares cas, polynomial en moyenne.

L'*algorithme du simplexe dual* effectue l'algorithme du simplexe sur le problème dual. Le premier intérêt est de gagner en temps de calcul dans le cas où le nombre de variables du primal est inférieur au nombre de contraintes, l'algorithme du simplexe étant plus efficace avec moins de contraintes. Le second intérêt du simplexe dual vient lorsqu'on réalise une succession de calculs de PL en rajoutant des contraintes, la

solution obtenue par le précédent PL est encore dual-réalisable, et peut initialiser un démarrage à chaud du simplexe de l'itération suivante, pour un fort gain en temps de calculs.

Des *algorithmes de points intérieurs* ont été élaborés, ne fournissant de solutions réalisables qu'à la dernière itération. Ces algorithmes sont polynomiaux, mais étaient pendant longtemps plus lents que les simplexes. De récents développements ont amélioré ces méthodes en pratique, nous référons à [33].

Relaxation linéaire d'un PLNE

Définition 1.1.5 (Relaxation linéaire) *La relaxation linéaire d'un PLNE est le programme linéaire obtenu en changeant le domaine de définition des variables entières par \mathbb{R}_+ .*

Minimiser sur le polyèdre de relaxation continue revient à minimiser sur un ensemble plus grand contenant les points entiers, et donc donne une borne inférieure :

Proposition 1.1.1 *La relaxation linéaire d'un PLNE donne une borne duale, calculable en temps polynomial. Dans le cas d'un problème de minimisation :*

$$\begin{array}{l} \min_{x \geq 0} c.x \\ s.c : A.x = b \end{array} \leq \begin{array}{l} \min_{x \in \mathbb{N}^m \times \mathbb{R}_+^p} c.x \\ s.c : A.x = b \end{array} \quad (1.7)$$

Relaxation lagrangienne On considère un PLNE sous l'écriture suivante :

$$\begin{array}{l} v(x) = \min_{x \in \mathbb{N}^n} cx \\ s.c : Ax \geq a \\ s.c : Bx \geq b \end{array} \quad (1.8)$$

Les contraintes $Ax \geq a$ sont dans l'usage les contraintes "difficiles", là où les équations $Bx \geq b$ permettent une résolution plus facile, par la reconnaissance d'un problème connu, ou présentant des problèmes découplés et indépendants. Dans ce dernier cas, les contraintes $Ax \geq a$ sont les contraintes couplantes. La relaxation lagrangienne dualise les contraintes couplantes en les faisant porter dans la fonction objectif :

$$v(x) = \min_{x \in \mathbb{N}^n, Bx \geq b} \max_{\lambda \geq 0} cx + \lambda(a - Ax) \quad (1.9)$$

Par inversion min-max,

$$v(x) \geq \max_{\lambda \geq 0} \min_{x \in \mathbb{N}^n, Bx \geq b} cx + \lambda(a - Ax) \quad (1.10)$$

Pour tout $\lambda \geq 0$, on note

$$l(\lambda) = \min_{x \in \mathbb{N}^n, Bx \geq b} cx + \lambda(a - Ax) \quad (1.11)$$

Définition 1.1.6 (Bornes lagrangiennes, borne lagrangienne) *Pour tout $\lambda \geq 0$, $l(\lambda)$ est une borne duale, dite borne lagrangienne. La borne lagrangienne est $\max_{\lambda} l(\lambda)$.*

Si la borne lagrangienne est plus difficile à calculer que la relaxation continue, elle a l'avantage théorique d'être toujours de meilleure qualité (on fournira une preuve de ce théorème section 1.3) :

Théorème 1.1.3 (Théorème de Geoffrion) *La borne lagrangienne est toujours meilleure que la borne de relaxation continue.*

Plusieurs algorithmes permettent de calculer des bornes lagrangiennes. Utilisant la convexité de l , une méthode de *descente de sous gradient* ([33]) peut être utilisée. La *méthode des volumes* ([19]) en est une variante efficace. La *méthode des faisceaux*, cf [86] résout le problème de maximisation par génération de coupes, tandis que la *reformulation de Dantzig-Wolfe* utilise la génération de colonnes (cf section 1.3). Ces quatre méthodes sont itératives et présentent des similitudes. À chaque itération, un signal de variables duales λ est réactualisé dans ces méthodes. Des calculs d'optimisation sur le domaine défini par $Bx \geq b$ sont effectués, et permettent de déduire une borne lagrangienne $l(\lambda)$. Les difficultés dans les cas proviennent de la stabilisation des variables duales λ , qui ont un comportement très erratique. Ces méthodes peuvent être comparées et combinées collaborativement, on pourra se reporter à [36].

1.1.3 Coupes d'intégrité

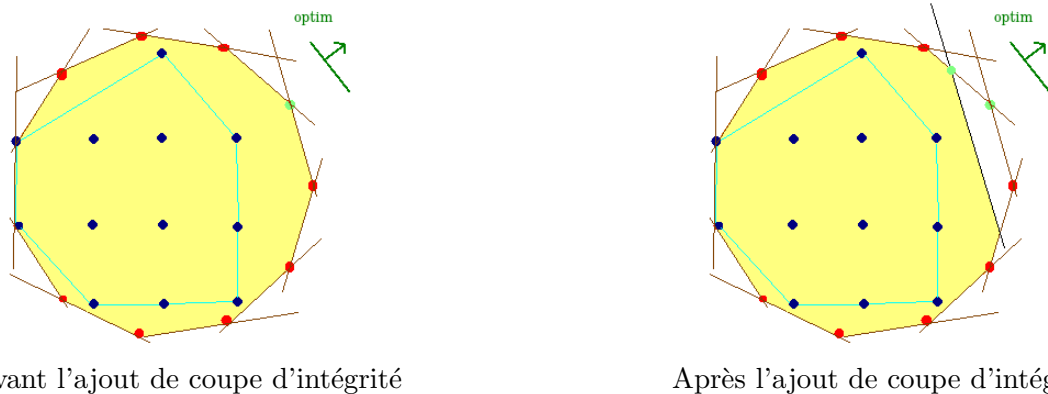


FIGURE 1.2 – Illustration des actions de coupes d'intégrité sur les domaines des solutions continues

Définition 1.1.7 (Coupe d'intégrité) Une coupe d'intégrité est une contrainte satisfaite par les points entiers d'un PLNE, permettant de réduire l'ensemble des solutions continues réalisables.

On détaille ici les types de coupes qui ont une utilisation directe dans la thèse. Pour une présentation plus exhaustive, on pourra se reporter à [31], [89], [65] et [88].

Coupes d'arrondis de Chvatal La procédure d'arrondi de Chvatal est basée sur le principe suivant. Pour des questions de granularité, si $x_1, x_2 \in \{0, 1\}$ une équation du type $2x_1 + 3x_2 \leq 4.9$ définit le même ensemble de points réalisables que $2x_1 + 3x_2 \leq 4$. Cette dernière contrainte, plus forte, réduit l'ensemble des solutions admissibles en préservant les solutions entières, et donc améliore la qualité de la borne duale obtenue par relaxation continue. Ce processus est utilisé pour générer les *coupes de Gomory* dans le cadre d'un PLNE avec des variables discrètes, et les *coupes mixtes de Gomory* dans le cas général. Ces coupes sont génériques pour tout PLNE, sans utiliser de structure particulière. Leur inconvénient est leur efficacité souvent limitée. La procédure d'arrondi de Chvatal peut s'effectuer sur une combinaison linéaire bien choisie de contraintes. C'est le cas des *Coupes "zero Half Cuts"* (terminologie de Cplex), qui applique le processus d'arrondi sur la moyenne de deux contraintes. Ainsi, des inégalités $x_1 + 4x_2 \leq 5$ et $3x_1 + 2x_2 \leq 4$, on obtient par moyenne $2x_1 + 3x_2 \leq 4.5$, ce qui donne la coupe $2x_1 + 3x_2 \leq 4$.

Coupes de couvertures Les *coupes de couvertures* s'appliquent pour des contraintes $\sum_i a_i x_i \leq d, a_i \geq 0$, dites de sac à dos. Si on a un sous ensemble de variables C , dite couverture, telle que $\sum_{i \in C} a_i > d$, tous les objets de C ne peuvent être pris en même temps, et cela impose la coupe $\sum_C x_i \leq |C| - 1$. Ces coupes peuvent être renforcées avec des contraintes GUB (Generalized Upper Bound), de type $\sum x_i \leq 1$. Dans ce

cas, le problème associé est un problème de sac à dos à choix multiple, où on ne peut prendre un seul objet d'un sous ensemble défini, les coupes de couvertures renforcées sont dites *coupes de couvertures GUB*.

Coupes de cliques Les *coupes de cliques* explicitent des relations d'incompatibilités entre variables. Un graphe, dit graphe de dépendance, est construit lors du prétraitement des solveurs de PLNE, associant un nœud pour chaque variable binaire x et sa complémentaire $\bar{x} = 1 - x$. Deux nœuds sont reliés si et seulement si les variables associées sont incompatibles. En particulier x et \bar{x} sont reliés dans le graphe de dépendance. Une contrainte de type $x_1 + x_2 \leq 1$ exprime l'incompatibilité entre x_1 et x_2 : une variable à 1 fixe l'autre variable à 0, et les événements $x_1 = 1$ et $x_2 = 1$ sont incompatibles. Sur une contrainte du type $x_1 \leq x_2$, on peut se ramener au cas précédent d'incompatibilité par la transformation suivante : $x_1 \leq x_2 \iff x_1 + 1 - x_2 \leq 1 \iff x_1 + \bar{x}_2 \leq 1$. Il s'agit alors d'une incompatibilité, donc un arc tracé entre x_1 et \bar{x}_2 . De manière générale, des relations d'incompatibilités entre variables binaires sont écrites lorsqu'une contrainte formalise une des quatre configurations d'implication suivante : $x_1 = 0 \implies x_2 = 0$, $x_1 = 1 \implies x_2 = 0$, $x_1 = 0 \implies x_2 = 1$ ou $x_1 = 1 \implies x_2 = 1$.

Une clique \mathcal{C} du graphe de dépendance permet d'écrire la coupe : $\sum_{x \in \mathcal{C}} x \leq 1$.

1.1.4 Résolution générique d'un PLNE : algorithme de Branch&Bound

Une raison du succès applicatif de la PLNE est l'existence d'une résolution générique par l'*algorithme de Branch&Bound* (B&B). Dans la suite, nous explicitons l'algorithme B&B sur un problème de minimisation, en développant les aspects pratiques pour une résolution efficace.

Points généraux Il n'est pas envisageable en général d'énumérer toutes les solutions d'un PLNE. L'algorithme B&B consiste à énumérer par un parcours d'arbre un nombre réduit de solutions guidé par une relaxation. Lorsque l'algorithme B&B converge, la solution retournée est prouvée optimale.

L'exploration nécessite une fonction d'évaluation donnant une borne inférieure pour tout sous-problème obtenu en cours d'exploration, comme les relaxations linéaire et lagrangienne. L'évaluation est combinée à des étapes de "séparation", partitionnant l'ensemble réalisable d'un problème en sous-problèmes plus faciles. L'algorithme itère avec une liste de sous-problèmes, la valeur de la meilleure solution trouvée, et la valeur de la borne inférieure prouvée du problème. Le critère d'arrêt prouvant l'optimalité de la meilleure solution trouvée est d'avoir l'égalité de ces deux bornes.

Lors d'une itération, on considère un sous-problème et on calcule la borne inférieure à l'aide de la fonction d'évaluation. Si le problème est infaisable ou donne une valeur supérieure à la meilleure solution trouvée, il est inutile d'explorer ce sous-problème comme l'optimum ne s'y trouve pas. Si la solution duale est entière et réalisable, la valeur duale est une solution primale donnant le minimum du sous problème, on actualise la meilleure solution courante et on supprime le sous problème. Enfin, dans les autres cas, on effectue une séparation en sous-problèmes que l'on rajoute à la liste à traiter, et on actualise la valeur de la borne inférieure. Dans ces trois cas, on a fini d'itérer, et on peut commencer une nouvelle itération en considérant un nouveau sous-problème, tant que le test d'arrêt n'est pas vérifié.

Algorithmes de branchement La séparation usuelle se fait de part et d'autre d'un hyperplan, ce qui préserve la structure PLNE après branchement en ajoutant une contrainte linéaire. En utilisant le simplexe dual, la solution PL précédente est dual-réalisable après l'ajout de la contrainte pour un démarrage à chaud efficace. Le simplexe dual est ainsi adapté à la résolution de PL à chaque nœud de l'arbre de B&B. La séparation canonique, *branchement sur la variable x* , fixe une variable x fractionnaire dans la relaxation continue, en regardant les sous-problèmes avec $x \leq \lfloor \tilde{x} \rfloor$ et $x \geq \lceil \tilde{x} \rceil$. Ce type de séparation n'est pas

toujours efficace, comme sur l'exemple des contraintes GUB de la forme $\sum_{i \in I} x_i \leq 1$. Par branchement sur les variables binaires, le nombre de nœuds visités est en $O(2^n)$ dans le pire des cas où $n = |I|$. Dans ce cas, on applique un algorithme de séparation spécifiques (dit SOS de type 1, nous référons à [20] et [124]) : on considère I_1 et I_2 une partition de I , de cardinaux $\lfloor \frac{n}{2} \rfloor$ et $\lceil \frac{n}{2} \rceil$. Les sous-problèmes sont construits en ajoutant les contraintes $\sum_{i \in I_1} x_i \leq 1$ et $\sum_{i \in I_2} x_i \leq 0$ et $\sum_{i \in I_2} x_i \leq 1$ et $\sum_{i \in I_1} x_i \leq 0$. Cette stratégie "diviser pour régner" engendre un arbre de recherche équilibré, visitant $O(n)$ nœuds.

Paramétrisations avancées L'algorithme de B&B regorge de paramètres à connaître pour utiliser au mieux un solveur sur un problème spécifique. On utilise dans cette thèse **Cplex** d'IBM Ilog par l'interface de modélisation OPL. On détaille ici les paramètres avec leur nom dans l'interface OPL.

Les solveurs modernes implémentent en fait un "Branch&Cut", des coupes d'intégrité sont ajoutées aux nœuds de l'arbre de branchement pour améliorer les bornes duales et explorer moins de nœuds. L'allongement des temps de calcul des bornes duales dû à l'ajout de coupes doit être négligeable devant la diminution du nombre de nœuds explorés. Cela peut être paramétré sous **Cplex**. La majeure partie des coupes sont générées au nœud racine, coupes valides pour tous les nœuds. Le paramètre **cutpass** peut limiter le nombre d'itérations d'ajouts de coupes à la racine. Pour chaque type de coupe, on peut paramétrer l'intensité de la génération de coupes. Connaissant les structures spécifiques d'un problème, on peut désactiver les coupes les moins pertinentes. Il s'agit respectivement des paramètres **gubcovers**, **cliques** et **zerohalfcuts** pour les coupes de couverture GUB, de cliques, et "zero half cuts".

De même, améliorer les bornes primales permet de couper plus tôt des branches de l'arbre de B&B. Des heuristiques de calcul de solution primale à chaque nœud ont été élaborées, nous y reviendrons plus loin. La fréquence d'appel de ses heuristiques peut être paramétrée, avec les paramètres **heurfreq** ou **rinsfreq**. Ces dernières fonctionnalités permettent d'utiliser Cplex dans un mode de résolution heuristique, juste dans le but de fournir d'excellentes solutions. Dans cette optique, les paramètres **tilim** et **odelim** permettent de stopper le calcul en temps ou nombre de nœuds limités. En temps limité, les calculs de preprocessing et de génération de coupes d'intégrité peuvent être désactivés, pour concentrer le temps de calcul sur la recherche et l'amélioration de solutions primales. En outre, des paramétrages automatiques sont disponibles avec Cplex selon que l'on veuille favoriser la recherche de solutions primales, la croissance de la borne duale, ou une stratégie mixte. Cela se fait par le paramètre **mipemphasis**. Il est également possible et souvent très intéressant de fournir à Cplex une bonne solution initiale pour un démarrage à chaud, ce qui active dès le début les heuristiques de recherche locale de Cplex nécessitant une première solution.

Notons que le critère d'arrêt de Cplex n'est pas exact, une tolérance est appliquée, par défaut 0,01% entre la borne duale prouvée et la meilleure solution primale, ie Cplex s'arrête quand la solution trouvée est prouvée à moins de 0,01% de l'optimum. Cela peut se modifier avec le paramètre **epgap**, critère d'écart relatif. Un critère d'écart absolu peut aussi être défini avec le paramètre **epagap**, ce qui peut être intéressant lorsque l'on connaît la granularité des solutions. Ainsi, pour un problème à valeurs entières, avoir prouvée une borne duale à 11,01 et avoir une solution primale à 12 permet d'avoir exactement prouvé que la solution du PLNE est 12. La granularité connue du problème se paramétrise alors avec **epagap=1**.

1.1.5 Résolutions spécifiques

Avec un algorithme de type Branch&Bound, la résolution d'un PLNE a une complexité exponentielle. Sur des cas particuliers, des résolutions spécifiques peuvent être plus adaptées, et mener à une complexité polynomiale, et également permettre une résolution sans passer par un solveur de PLNE, nécessitant une licence commerciale. De tels cas particuliers sont détaillés dans cette section.

Algorithmes de graphes spécifiques Historiquement, les problèmes d’optimisation issus de la théorie des graphes ont été activement étudiés dans la littérature, fournissant de nombreux algorithmes polynomiaux pour des problèmes d’optimisation basés sur des graphes. Nous citerons le problème de plus court chemin entre deux nœuds d’un graphe (algorithme de Dijkstra), le problème de l’arbre de poids couvrant minimum (algorithme de Kruskal), le problème de flot maximal (resp de coupes minimale) dans un graphe avec l’algorithme de Ford-Fulkerson (resp l’algorithme d’Edmonds-Karp), ou le problème d’affectation d’un graphe biparti avec l’algorithme hongrois. Nous référons à [61] pour plus de détails.

Algorithme de programmation dynamique La programmation dynamique s’applique à des problèmes composés de plusieurs sous-problèmes de même nature. La programmation dynamique s’appuie sur le principe d’optimalité de Bellman : toute solution optimale combine des solutions optimales d’une série de sous-problèmes selon une stratégie ”diviser pour mieux régner”. Le problème de plus court chemin dans un graphe rentre dans ce cadre, l’algorithme de Dijkstra étant une programmation dynamique ([32]). La programmation dynamique s’applique aussi en ordonnancement, nous référons à [41],[50].

Programmation par contraintes Dans le cas où un PLNE à valeurs entières bornées n’a pas de fonction objectif, on se trouve dans le cadre des problèmes de satisfaction de contraintes (CSP pour Constraint Satisfaction Problem en Anglais). Un CSP se modélise formellement par un triplet (X, D, C) tel que $X = \{X_1, X_2, \dots, X_n\}$ désigne l’ensemble des variables (à valeurs discrètes) du problème, D est une fonction qui associe à chaque variable X_i son domaine $D(X_i)$, l’ensemble (fini) des valeurs que peut prendre X_i , $C = \{C_1, C_2, \dots, C_k\}$ est l’ensemble des contraintes. Chaque contrainte C_j est une relation entre un sous ensemble de variables de X , restreignant les valeurs que ces variables peuvent prendre simultanément. Le cadre des contraintes de CSP est bien plus général que les contraintes linéaires de la PLNE. Comme en PLNE, un problème est modélisé globalement, avec une résolution générique par la Programmation Par Contraintes (PPC) permettant une démarche ”model&run”. La PPC garantit de trouver une solution, quand elle existe, et prouve qu’il n’existe pas de solution à un problème lorsque la recherche exhaustive se termine sans avoir de solution.

La PPC est une recherche arborescente, présentant les mêmes types de branchements que l’algorithme de Branch&Bound. Le *filtrage* ou *propagation de contraintes* évite l’énumération exhaustive, en réduisant les domaines des variables à l’aide des contraintes et des valeurs déjà fixées. On peut alors aboutir à une preuve d’absence de solution lorsqu’une variable n’a plus de valeur admissible. Les calculs de relaxation continue du B&B font office de filtrage. Les algorithmes de filtrages de la PPC sont en général moins gourmand en espace mémoire et en temps de calcul, ce qui permet d’envisager des résolutions sur des problèmes de plus grande taille que la PLNE. De même que pour le B&B, la paramétrisation efficace de la PPC tient d’un bon compromis à trouver entre le temps nécessaire au filtrage et son efficacité.

La PPC s’est retrouvée être une approche alternative à la PLNE pour des problèmes d’optimisation très contraints et/ou de grande taille, nous référons par exemple à [35],[70]. La PPC a permis des avancées sur la résolution PLNE par Branch&Bound, mise en évidence dans [2], notamment en y incorporant des algorithmes de propagation de contraintes de la PPC.

1.2 Décomposition de Benders

Cette section présente le cadre de la décomposition de Benders, avec sa résolution par génération de coupes. Deux types d’hypothèses sur le PLNE initial sont envisagés : le cadre où les variables de second niveau sont continues et les variables de premier niveau quelconques, et le cadre où les variables de premier niveau sont binaires.

1.2.1 Génération de coupes

Certains problèmes peuvent se formuler en PL/PLNE avec un nombre exponentiel de contraintes, comme le problème du voyageur de commerce ([68]). Une génération dynamique de coupes permet d'ajouter dynamiquement des contraintes afin de traiter une succession de problèmes de taille raisonnable :

Algorithme de génération de coupes

Initialisation : on sélectionne un ensemble initial de contraintes (de taille raisonnable)

Étape 1 : On calcule une solution du PL ou PLNE avec les contraintes sélectionnées.

Étape 2 : Existe-t'il une contrainte non écrite qui viole la solution partielle ?

Étape 3 : Si oui, on exhibe et on ajoute la contrainte dans la sélection et on revient à l'étape 1.

Étape 4 : Si non, l'algorithme s'arrête, la solution partielle est la solution du problème global.

Par ce processus incrémental, à chaque calcul intermédiaire, on obtient une borne duale, croissante pour un problème de minimisation. La valeur de la borne duale peut être constante d'une itération à l'autre, il s'agit de cas de dégénérescence, en aucun cas cela ne garantit la terminaison de l'algorithme.

Le point crucial est l'étape 2 : savoir déceler rapidement dans un nombre exponentiel de contraintes si elles seront toutes réalisées dans la solution courante ou sinon savoir détecter et exhiber rapidement parmi un nombre exponentiel de contraintes une coupe violée à ajouter. Il s'agit de l'algorithme de séparation :

Définition 1.2.1 (Algorithme de séparation) *L'algorithme de séparation est l'algorithme qui garantit qu'aucune contrainte non écrite est satisfaite par la solution partielle et qui dans le cas contraire exhibe une coupe violée à ajouter.*

Théorème 1.2.1 (Théorème de Lovasz-Schrijver-Grötschel) *Pour un PL au nombre exponentiel de contraintes, si il existe un algorithme de séparation polynomial, le calcul du PL est aussi polynomial.*

1.2.2 Reformulation de Benders en Programmation Linéaire mixte

On présente ici la décomposition de Benders dans le cadre de la PLNE. Pour un cadre plus général, on pourra se reporter à [32]. On opère une partition des variables, $X = \{x \in \mathbb{N}^p \times \mathbb{R}^{n-p} \mid Ax \geq a\}$ comprenant toutes les variables discrètes. On réécrit le problème sous la forme suivante avec une variable de coût $\alpha \geq 0$:

$$\begin{array}{ll}
 \min_{x \in X, y \geq 0} & cx + dy \\
 \text{s.c :} & Ax \geq a \\
 & Tx + Wy \geq h \\
 & By \geq b
 \end{array}
 =
 \begin{array}{ll}
 \min_{x \in X, y \geq 0} & cx + \alpha \\
 \text{s.c :} & Ax \geq a \\
 & Tx + Wy \geq h \\
 & By \geq b \\
 & dy \leq \alpha
 \end{array}
 \quad (1.12)$$

A (x, α) fixé, la question de faisabilité des contraintes se ramène à un calcul de PL, donc polynomial, en ajoutant à chaque contrainte originelle une variable positive η , chiffrant la pénalisation des contraintes à satisfaire, et en minimisant la pénalisation :

$$\begin{array}{ll}
 \eta^* = & \min_{y, \eta \geq 0} \eta \\
 \text{s.c :} & Wy + \eta \mathbf{1} \geq h - Tx \quad (\lambda_1) \\
 & By + \eta \mathbf{1} \geq b \quad (\lambda_2) \\
 & -dy + \eta \mathbf{1} \geq -\alpha \quad (\lambda_3)
 \end{array}
 \quad (1.13)$$

La faisabilité des contraintes équivaut à $\eta^* = 0$. Avec la dualité forte :

$$\begin{aligned}
\eta^* &= \max_{\lambda_1, \lambda_2, \lambda_3 \geq 0} \lambda_1(h - Tx) + \lambda_2 b - \lambda_3 \alpha \\
s.c : & \lambda_1 W + \lambda_2 B - \lambda_3 d \leq 0 \\
& \lambda_1 \mathbb{1} + \lambda_2 \mathbb{1} + \lambda_3 \mathbb{1} \leq 1
\end{aligned} \tag{1.14}$$

Soit J l'ensemble des directions extrêmes de $(WBd)^T$. $\eta^* = 0$ équivaut à :

$$\forall (\lambda_1, \lambda_2, \lambda_3) \in J, \quad \lambda_1(h - Tx) + \lambda_2 b - \lambda_3 \alpha \leq 0 \tag{1.15}$$

On remarquera que l'on retrouve des coupes de faisabilité si $\lambda_3 = 0$, et des coupes d'optimalité sinon. Le problème initial est alors équivalent à la reformulation de Benders suivante :

$$\begin{aligned}
& \min_{x \in X, \alpha \geq 0} cx + \alpha \\
s.c : & Ax \geq a \\
\forall (\lambda_1, \lambda_2, \lambda_3) \in J, & \lambda_1(h - Tx) + \lambda_2 b - \lambda_3 \alpha \leq 0
\end{aligned} \tag{1.16}$$

Il s'agit d'une formulation ayant un nombre exponentiel de contraintes, portant uniquement sur les variables en x . La résolution se fait ainsi par génération de coupes.

Problème maître de Benders Le Problème Maître Restreint (PMR) est la reformulation de Benders avec les coupes déjà générées :

$$\begin{aligned}
& \min_{x \in X, \alpha \geq 0} cx + \alpha \\
s.c : & Ax \geq a \\
\forall (\lambda_1, \lambda_2, \lambda_3) \in J, & \lambda_1(h - Tx) + \lambda_2 b - \lambda_3 \alpha \leq 0
\end{aligned} \tag{1.17}$$

La valeur du PMR est une borne inférieure du problème. Pour la solution partielle x , on peut calculer une borne supérieure par un calcul de PL, donc polynomialement par :

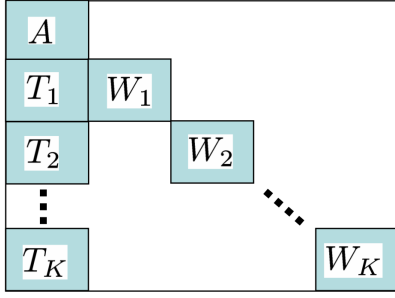
$$\begin{aligned}
cx + & \min_{y \geq 0} dy \\
s.c : & Tx + Wy \geq h \\
& By \geq b
\end{aligned} \tag{1.18}$$

Problème esclave, séparation de Benders La séparation de Benders, problème esclave, se formule comme la question : A une solution partielle (x, α) , peut on générer des coupes violées ?

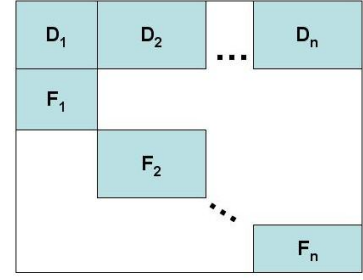
$$\begin{aligned}
\eta^* &= \min_{y, \eta \geq 0} \eta \\
s.c : & Wy + \eta \mathbb{1} \geq h - Tx \quad (\lambda_1) \\
& By + \eta \mathbb{1} \geq b \quad (\lambda_2) \\
& -dy + \eta \mathbb{1} \geq -\alpha \quad (\lambda_3)
\end{aligned} \tag{1.19}$$

Les contraintes du problème sont réalisables pour la solution partielle (x, α) si et seulement si $\eta^* = 0$. Dans le cas contraire, les valeurs duales donnent une coupe violée, à ajouter au PMR :

$$\lambda_1(h - Tx) + \lambda_2 b - \lambda_3 \alpha \leq 0 \tag{1.20}$$



Pour la décomposition de Benders



Pour la reformulation de Danzig-Wolfe

FIGURE 1.3 – Structures matricielles privilégiées pour les reformulations de Benders et de Danzig-Wolfe

1.2.3 Cas partiellement bloc diagonal

Le cadre d'application naturel pour la reformulation de Benders est sur un problème comme illustré sur la figure 1.3, où des variables liantes x couplent K sous-problèmes indépendants à variables y^k :

$$\begin{aligned}
 & \min_{x, y^k \geq 0} c^T x + \sum_{k \in \mathcal{K}} q^k y^k \\
 \text{s.t. :} & \quad Ax \geq a \\
 & \quad \forall k \in \mathcal{K}, \quad T^k x + W^k y^k \geq h^k \\
 & \quad \forall k \in \mathcal{K}, \quad B^k y^k \geq b^k \\
 & \quad x \in \mathbb{N}^m \times \mathbb{R}_+^n
 \end{aligned} \tag{1.21}$$

Approche classique de génération de coupes Dans l'approche historique, issue de la décomposition de Benders écrite comme dans la section 2, la séparation se déroule par opérations successives. On regarde d'abord si les sous-problèmes pour $k \in \mathcal{K}$ sont réalisables. Si ce n'est pas le cas, on génère pour chaque sous-problème irréalisable une coupe de faisabilité en calculant le PL suivant :

$$\begin{aligned}
 \eta_k^* = & \min_{y, \eta \geq 0} \eta \\
 \text{s.c. :} & \quad W^k y + \eta \mathbf{1} \geq h^k - T^k x \quad (\lambda_1^k) \\
 & \quad B^k y^k + \eta \mathbf{1} \geq b^k \quad (\lambda_2^k)
 \end{aligned} \tag{1.22}$$

La coupe d'infaisabilité du sous bloc k tel que $\eta_k^* > 0$ s'écrivant alors :

$$\lambda_1^k (h - T^k x) + \lambda_2^k b^k \leq 0 \tag{1.23}$$

Sinon, on regarde si on doit générer une coupe d'optimalité par le PL :

$$\begin{aligned}
 \eta^* = & \min_{y^k, \eta \geq 0} \eta \\
 \text{s.c. :} & \quad - \sum_{k \in \mathcal{K}} q^k y^k + \eta \geq -\alpha \quad (\lambda_3) \\
 & \quad \forall k \in \mathcal{K}, \quad W^k y^k \geq h^k - T^k x \quad (\lambda_1^k) \\
 & \quad \forall k \in \mathcal{K}, \quad B^k y^k \geq b^k \quad (\lambda_2^k)
 \end{aligned} \tag{1.24}$$

Si $\eta^* = 0$, la solution courante est l'optimum. Sinon, on rajoute la coupe d'optimalité :

$$\sum_k \lambda_1^k (h^k - T^k x) + \sum_k \lambda_2^k b^k - \lambda_3 \alpha \leq 0 \tag{1.25}$$

Version multi-coupes Précédemment, on générerait soit autant de coupes de faisabilités que de scénarios irréalisables, soit une seule coupe d'optimalité. Une approche multi-coupes permet de générer à toute

itération autant de coupes que de sous bloc k . La reformulation du problème s'écrit en introduisant pour tout sous bloc k une variables $z^k \in \mathbb{R}_+$:

$$\begin{aligned}
 & \min_{x, y^k \geq 0} c^T x + \sum_{k \in \mathcal{K}} z_k \\
 \text{s.t. :} & Ax \geq a \\
 & \forall k \in \mathcal{K}, \quad T^k x + W^k y^k \geq h^k \\
 & \forall k \in \mathcal{K}, \quad B^k y^k \geq b^k \\
 & \forall k \in \mathcal{K}, \quad q^k y^k \geq z_k \\
 & x \in \mathbb{N}^m \times \mathbb{R}_+^n
 \end{aligned} \tag{1.26}$$

La décomposition de Benders considère les variables x et z_k comme variables de premier niveau, ce qui ramène le problème à une question de faisabilité :

$$\begin{aligned}
 \eta^* = & \min_{y_k, \eta \geq 0} \eta \\
 \forall k \in \mathcal{K}, & W^k y^k + \eta \geq h^k - T^k x \quad (\lambda_1^k) \\
 \forall k \in \mathcal{K}, & B^k y^k \geq b^k \quad (\lambda_2^k) \\
 \forall k \in \mathcal{K}, & q^k y^k + \eta \geq z_k \quad (\lambda_3^k)
 \end{aligned} \tag{1.27}$$

La faisabilité de tous les sous-problèmes équivaut à $\eta^* = 0$. On remarque qu'on a $\eta^* = 0$, si et seulement tous les sous-problèmes en k sont satisfaits indépendamment les uns des autres. Cela se décompose en K problèmes indépendants, avec $\eta^* = \min_k \eta_k^*$:

$$\begin{aligned}
 \eta_k^* = & \min_{\eta, y^k \geq 0} \eta \\
 \text{s.c. :} & W^k y + \eta \geq h^k - T^k x \quad (\lambda_1^k) \\
 & B^k y^k \geq b^k \quad (\lambda_2^k) \\
 & q^k y^k + \eta \geq z_k \quad (\lambda_3^k)
 \end{aligned} \tag{1.28}$$

Si pour tout k , $\eta_k^* = 0$, la valeur courante x, z^k fournit l'optimum. Sinon, pour chaque sous bloc k tel que $\eta_k^* > 0$, la coupe à générer s'écrit :

$$\sum_k \lambda_1^k (h^k - T^k x) + \sum_k \lambda_2^k b^k - \lambda_3^k z_k \leq 0 \tag{1.29}$$

1.2.4 Points d'implémentation

Cette section traite de points spécifiques d'implémentations de la décomposition de Benders. Trois types d'implémentations sont décrits, l'implémentation séquentielle, l'implémentation Branch&Cut dans le cas d'un PLNE, et l'implémentation heuristique tronquée.

Implémentation séquentielle L'implémentation séquentielle d'une décomposition de Benders calcule à l'optimalité le PMR à chaque itération avec les coupes déjà générées. La séparation s'applique à la solution optimale x obtenue, calculant autant de PL que de problème esclaves. Si des coupes violées peuvent être ajoutées, on réitère le processus en recalculant le problème maître avec les coupes générées, sinon la solution obtenue est prouvée optimale, l'algorithme s'arrête. Chaque itération fournit une borne duale du problème après un calcul de PMR. La borne duale est croissante au cours des itérations, pas toujours strictement. À partir de la solution optimale du PMR obtenue x , on peut reconstituer une solution primale par (1.18).

Dans le cas où le PMR est continu, une succession PL est calculée, les PMR s'obtenant en rajoutant un petit nombre en contraintes. Une telle implémentation est efficace en calculant l'algorithme le PMR par simplexe dual avec un démarrage à chaud utilisant de la dernière solution, toujours dual-réalisable. Dans

le cas où le PMR est entier, chaque itération demande un calcul à l’optimalité d’un PLNE. Un tel schéma n’est envisageable en pratique que si le PMR se résout aisément.

Implémentation Branch&Cut Quand le PMR est entier, une implémentation en Branch&Cut permet de parcourir un seul arbre de Branch&Bound sur les variables du PMR. L’algorithme de séparation est valide pour une solution x fractionnaire, comme dans la décomposition de Benders du relâché continu. L’idée est alors d’ajouter à chaque nœud toutes les coupes de Benders violées par la solution fractionnaire en cours. Quand la solution fractionnaire d’un nœud ne viole plus de coupes de Benders, on effectue des branchements, pour aboutir à des solutions entières. Lorsqu’une solution entière trouvée ne viole pas de coupe de Benders, c’est une solution réalisable du problème, on a une borne primale. Les bornes duales obtenues par relaxation continue et décomposition de Benders à chaque nœud fournissent des bornes duales du problème conditionnellement aux branchements réalisés. Les itérations de décomposition de Benders portent alors sur des problèmes continus, où l’implémentation séquentielle avec un démarrage à chaud par l’algorithme de simplexe dual est efficace.

Potentiellement, un nombre important de coupes de Benders peut être généré, ce qui peut devenir prohibitif en terme d’espace mémoire. Pour éviter cela, une implémentation classique en Branch&Cut définit des coupes locales : une coupe de Benders générée à un nœud n’est écrite que pour les nœuds successeurs. Cette approche a été utilisée dans [17], où l’implémentation est réalisée sous cplex à l’aide de callbacks. Des coupes de Benders non utilisées depuis un grand nombre d’itérations sont retirées pour libérer de l’espace mémoire. Un tel algorithme de Branch&Cut est compatible avec la génération de coupes d’intégrité à chaque nœuds de l’arbre de Branch&Cut, similairement à la section 1.1.4.

Approche heuristique tronquée Dans le cas d’un PLNE, l’approche séquentielle est limitante en temps de calculs quand l’approche Branch&Cut est limitante en espace mémoire. Dans les deux cas, les tailles des problèmes résolubles à l’optimalité sont limitées. Un compromis est souvent élaboré sur de grands problèmes, par exemple dans [101] et [91], aboutissant à des heuristiques basées sur la décomposition de Benders. La relaxation continue est calculée exactement par décomposition de Benders suivant l’implémentation séquentielle, et le calcul de PLNE est poursuivi avec les coupes générées sur la relaxation PL, sans génération de nouvelles coupes de Benders, pour une approche *Cut&Branch*. Les bornes duales obtenues par l’approche Cut&Branch fournissent des bornes duales du problème complet.

Génération de coupes de Benders Pour générer des coupes de Benders, seule une solution partielle (x, α) est nécessaire. Le calcul à l’optimalité du PMR permet de prouver que la solution partielle est optimale si elle respecte toutes les coupes de Benders. Ainsi, pour les premières itérations de la décomposition de Benders, le calcul à l’optimalité est facultatif. Lorsque le PMR est un PLNE, pour économiser du temps de résolution sur les premières itérations d’une implémentation séquentielle. Dans un tel cadre, la PPC a été utilisée pour fournir des résolutions accélérés de PMR dans [70].

Dans le cas de sous-problèmes indépendants, on peut accélérer les temps de calcul en résolvant un nombre déterminé de sous-problèmes aux premières itérations. Cela permet aussi de ne pas générer de coupes redondantes. En effet, dans le cas où deux sous blocs sont identiques, les coupes de Benders générées sont identiques. L’application associant aux coefficients matriciels et à la solution de premier niveau la coupe de Benders associée est linéaire et donc continue. Pour une solution de premier niveau donnée, des valeurs de coefficients matriciels ”proches” induisent des coupes de Benders ”proches”.

Un point crucial pour accélérer la terminaison d’un algorithme de génération de coupes est la stabilisation, diminuer le nombre d’itérations nécessaires. La stabilisation d’une décomposition de Benders a été peu étudiée spécifiquement dans la littérature. Nous référons à [21] pour des techniques générales de stabil-

isation de génération de coupes ou de colonnes, et à [76] pour des techniques de stabilisation de coupes de Benders avec une application à un problème d'ordonnement. Nous notons enfin que la PPC peut être utile pour fournir des coupes de Benders renforcées, en nous référant à [110].

Instabilité numérique et coupes de Benders Un point crucial pour implémenter la décomposition de Benders est que les coupes de Benders dépendent des solutions duales. Les erreurs d'arrondis numériques s'accumulent dans le PMR avec l'ajout de coupes de Benders, un grand soin doit être apporté aux arrondis numériques : une coupe $\lambda_1(h - Tx) + \lambda_2b \leq \lambda_3\alpha$ avec λ_3 très petit, se trouvant arrondi à 0, devient une coupe de faisabilité fautive. Pour minimiser les instabilités numériques d'arrondis, il est souhaitable d'avoir des coefficients de même ordre de grandeur. Pour ce faire, on peut tout d'abord normaliser les variables. Il est également souhaitable que les variables duales des sous-problèmes aient le même ordre de grandeur. Là encore, on peut opérer une opération de normalisation. Pour écrire les variables duales, on avait introduit η pour mesurer les violations de contraintes sans avoir normalisé. Pour chaque contrainte j , cela on remplace $1 \times \eta_j$ par $\tilde{\eta}_j = M_j\eta_j$ où $M_j > 0$ indique l'ordre de grandeur de la contrainte j .

1.2.5 Cas des sous-problèmes entiers

Pour des applications pratiques, des variables binaires peuvent apparaître dans le second membre y . Une possibilité est d'appliquer la décomposition de Benders précédente en ajoutant les variables binaires dans le problème maître. Un schéma de décomposition alternatif peut être proposé quand les variables x sont toutes binaires (un problème entier à valeurs bornées peut toujours se ramener à ce cas).

$$\begin{aligned} & \min_{x,y} cx + dy \\ \text{s.c : } & Ax \geq a \\ & Tx + Wy \geq h \\ & By \geq b \\ & x \in \{0, 1\}^m, y \in \mathbb{N}^n \times \mathbb{R}_+^p \end{aligned} \tag{1.30}$$

Soit $\mathcal{D} = \{x \in \{0, 1\}^m, Ax \geq a\}$. Pour tout $x \in \mathcal{D}$, on définit :

$$\begin{aligned} Q(x) = \min_{y \in \mathbb{N}^n \times \mathbb{R}_+^p} & dy \\ & x + Wy \geq h \\ & By \geq b \end{aligned} \tag{1.31}$$

Le problème se réécrit comme $\min_{x \in \mathcal{D}} Q(x)$. On suppose disposer pour Q d'un minorant L :

$$\forall x \in \mathcal{D}, \quad Q(x) \geq L \tag{1.32}$$

Soit $x^0 \in \mathcal{D}$. On distingue les cas sur la réalisabilité de sous-problèmes associés à x^0 , pour en déduire des contraintes valides pour tout $x \in \mathcal{D}$. Si $x^0 \in \mathcal{D}$ ne fournit pas de solution réalisable, ie $Q(x^0)$ n'est pas défini, une première condition nécessaire pour que $x \in \mathcal{D}$ est qu'une de ses variables au moins a une valeur différente dans x^0 . Cela s'écrit avec la contrainte linéaire :

$$\forall x \in \mathcal{D}, \psi_{x^0}(x) = \sum_{i:x_i^0=0} x_i + \sum_{i:x_i^0=1} (1 - x_i) \geq 1 \tag{1.33}$$

Ces coupes sont analogues à des coupes de faisabilités. On notera la faiblesse de ces coupes. Si on sait détecter qu'un sous ensemble de variables mène à une infaisabilité quelque soient les valeurs des autres variables, on a une contrainte renforcée en considérant dans une coupe (1.33) un ensemble de cardinalité minimale induisant une contradiction. Un tel point est crucial pour ne pas induire de coupes trop faibles.

Si $x^0 \in \mathcal{D}$ fournit une solution réalisable, ie $\mathcal{Q}(x^0)$ est bien défini, on définit ϕ_{d^0} par l'égalité suivante :

$$\phi_{x^0}^L(x) = \mathcal{Q}(x^0) + (L - \mathcal{Q}(x^0)) \left(\sum_{i:x_i^0=0} x_i + \sum_{i:x_i^0=1} (1 - x_i) \right).$$
 On remarque alors que $\phi_{x^0}^L(x^0) = \mathcal{Q}(x^0)$ et $\phi_{x^0}(x^0) \leq L$ pour $x \in \mathcal{D} \setminus \{x^0\}$. De manière générale, pour tout $x \in \mathcal{D}$, tel que $\mathcal{Q}(x)$ est défini, $\mathcal{Q}(x) = \max_{x^0 \in \mathcal{D}} \phi_{x^0}^L(x)$. La distinction de cas permet de reformuler le problème comme :

$$\begin{aligned} & \min_{x, \Theta} cx + \Theta \\ & Ax \geq a \\ & \Theta \geq L \\ & \forall d^0 \in \mathcal{D}, \mathcal{Q}(d^0) < \infty, \quad \phi_{d^0}(x) \leq \Theta \\ & \forall d^0 \in \mathcal{D}, \mathcal{Q}(d^0) = \infty, \quad \psi_{d^0}(x) \geq 1 \end{aligned} \tag{1.34}$$

Les coupes $\phi_{d^0}(x) \leq \Theta$ sont analogues aux coupes d'optimalité, tandis que les coupes $\psi_{d^0}(x) \geq 1$ sont analogues aux coupes de réalisabilité. Un algorithme similaire à la décomposition de Benders peut être implémenté, où on itère entre un problème maître PLNE en considérant un nombre restreint de coupes, avec des calculs de PLNE, $\mathcal{Q}(x)$ pour x solution du problème maître, pour générer des coupes violées.

1.3 Formulation de Dantzig-Wolfe et génération de colonne

La reformulation de Dantzig-Wolfe transforme un PLNE compact sous une forme étendue avec un grand nombre de variables, permettant de calculer de meilleures bornes duales que la relaxation continue du PLNE compact. La résolution de telles bornes duales se fait par génération de colonnes. Cette approche a fourni d'excellents résultats sur les problèmes de découpe de rouleaux de papiers ([40]), de problèmes de tournées de véhicules ([52]), de Bin-Packing ([43]) et de Lot-sizing ([123]).

1.3.1 Génération de colonne

L'algorithme de génération de colonne s'applique à un PL, où le nombre de colonnes est très grand devant le nombre de contraintes, exponentiel en général, et parfois non énumérable dans sa totalité. On considère un PL de la forme suivante, où $A \in \mathcal{M}^{m,n}(\mathbb{R})$, $x, c \in \mathcal{M}^{n,1}(\mathbb{R})$, $b \in \mathcal{M}^{m,1}(\mathbb{R})$, avec $m \ll n$:

$$\begin{aligned} & \min_{x \geq 0} c^T . x \\ s.c : & Ax \geq b \end{aligned} \tag{1.35}$$

Algorithme de génération de colonne Dans la résolution d'un tel PL, la majorité des variables est hors base et donc nulle sur les points extrêmes. L'idée est de considérer le problème restreint à un nombre de variables raisonnable, et de générer dynamiquement les variables selon le besoin. À chaque étape, on a une solution réalisable du problème complet en considérant les autres variables comme nulles. Le point clé est de savoir comment ajouter des variables, et prouver avoir atteint l'optimum du problème complet avec un nombre restreint de variables. Pour cela, le coût réduit interviendra :

Définition 1.3.1 Le coût réduit associé à la variable x_i est défini par $CR(x_i) = c_i - \sum_{j=1}^m \lambda_j A_{i,j}$.

Pour contrer des arrondis numériques, le critère d'arrêt est $\min_i CR(x_i) \leq -\varepsilon$ avec une tolérance $\varepsilon > 0$.

Algorithme de génération de colonnes

Initialisation : on sélectionne un ensemble initial de variables

Étape 1 : On calcule une solution du problème avec les variables sélectionnées.

Étape 2 : Existe-t'il une variable au coût réduit strictement négatif? ie, a-t'on $\min_i CR(x_i) < 0$?

Étape 3 : Si oui, on exhibe et on ajoute la variable dans la sélection de variables et on revient à l'étape 1.

Étape 4 : Si non, l'algorithme s'arrête, la solution partielle est la solution du problème global.

Justification, pourquoi le coût réduit ? On justifie ici la validité de l'algorithme. Dans le cadre de la dualité forte, le PL précédent a la même valeur que son dual, qui s'écrit :

$$\begin{aligned} & \max_{\lambda \geq 0} \sum_{j=1}^m b_j \lambda_j \\ s.c : & \forall i \in \llbracket 1, n \rrbracket \quad \sum_{j=1}^M A_{i,j} \lambda_j \leq c_i \end{aligned} \quad (1.36)$$

Le dual comprend un grand nombre de contraintes ($n \gg m$), générer une variable dans le primal restreint revient à générer une contrainte dans le dual. Le dual se calcule par génération de coupes, comme précédemment, et une contrainte doit être rajoutée si et seulement si il y a violation, ie il existe $i \in \llbracket 1, n \rrbracket$, tel que $\sum_{j=1}^m \lambda_j A_{i,j} > c_i$, autrement dit $CR(x_i) = c_i - \sum_{j=1}^m \lambda_j A_{i,j} < 0$. On a ainsi retrouvé le coût réduit. La génération de coupe dans le dual produit des contraintes à ajouter tant que $\min_i CR(x_i) < 0$. Quand $\min_i CR(x_i) \geq 0$, la solution dans le dual restreint est la solution du dual comprenant toutes les contraintes. Par passage à la dualité, le primal restreint a la même valeur que le primal complet.

1.3.2 Reformulation de Dantzig-Wolfe

La reformulation de Dantzig-Wolfe se définit littéralement comme une reformulation étendue d'un PLNE, obtenue par dualisation lagrangienne, et permettant de calculer également des bornes lagrangiennes. La relaxation continue de cette formulation se calcule par génération de colonne. Mathématiquement, on considère un PLNE de la forme suivante où on dualise les contraintes difficiles ou liantes $Ax \geq a$:

$$\begin{aligned} v(x) &= \min_{x \in \mathbb{N}^n \times \mathbb{R}_+^p} cx \\ s.c : & \quad Ax \geq a \\ s.c : & \quad Bx \geq b \end{aligned} \quad (1.37)$$

Comme cadre suffisant pour la suite, on traite ici le cas des problèmes discrets, ie $x \in \mathbb{N}^n$, où $P = \{x \in \mathbb{N}^n, Bx \geq b\}$ est non vide et borné. Pour le cas général, on se reportera à [112]. En énumérant les points de P , $\{x \in \mathbb{N}^n, Bx \geq b\} = \{x_q\}_{q \in \mathcal{Q}}$, un point entier de P s'écrit $x = \sum_q \lambda_q x_q$, avec $\lambda_q \in \{0, 1\}^{|\mathcal{Q}|}$ et $\sum_q \lambda_q = 1$. La reformulation de Dantzig Wolfe est la formulation suivante, indexée sur les points entiers de P :

$$\begin{aligned} v(x) &= \min_{\lambda \in \{0,1\}^{|\mathcal{Q}|}} \sum_q \lambda_q (cx_q) \\ s.c : & \quad \sum_q \lambda_q (Ax_q) \geq a \\ & \quad \sum_q \lambda_q = 1 \end{aligned} \quad (1.38)$$

Mise en œuvre de la procédure de génération de colonne La relaxation continue de la reformulation de Dantzig-Wolfe se calcule par génération de colonne. On définit alors le *Problème Maître Restreint* (PMR) comme le problème limité à un nombre restreint de points extrêmes de colonnes :

$$\begin{aligned} v(x) &= \min_{\lambda \in \mathbb{R}_+^{|\mathcal{Q}|}} \sum_q \lambda_q (cx_q) \\ s.c : \quad & \sum_q \lambda_q (Ax_q) \geq a \quad (\pi) \\ & \sum_q \lambda_q = 1 \quad (\sigma) \end{aligned} \tag{1.39}$$

σ est alors une variable duale non signée, contrairement à $\pi \geq 0$. Le coût réduit d'une colonne q est $CR_q = cx_q + \sigma - \pi Ax_q$ et le sous-problème de génération de colonne est ainsi :

$$\begin{aligned} & \min_{x \in \mathbb{N}^n} \sigma + (c - \pi A)x \\ s.c : \quad & Bx \geq b \end{aligned} \tag{1.40}$$

Il s'agit d'un PLNE, qui reste difficile (non polynomial) en général, une résolution spécifique est intéressante. Dans le cas où le sous-problème est polynomial, le théorème de Lovasz permet d'affirmer que le calcul de la relaxation continue de la formulation étendue reste polynomial.

Points d'implémentation L'implémentation de la génération de colonne nécessite des précautions. Tout d'abord, il est nécessaire d'initialiser l'algorithme avec un ensemble de colonnes garantissant l'existence d'une solution réalisable. Dans le cas contraire, le premier problème maître est infaisable, et cela ne permet pas de poursuivre l'algorithme. On pourra contourner ces difficultés en ajoutant des pénalisations pour garantir la réalisabilité du PMR.

Les instabilités numériques provenant d'erreurs d'arrondis numériques peuvent survenir avec les variables duales π et σ . Une erreur d'arrondi perturbe uniquement la fonction objectif des sous-problèmes et donc le critère d'arrêt. Les instabilités numériques sont contournées en générant les colonnes avec une tolérance $\varepsilon > 0$, ie en changeant le critère d'arrêt et de génération de colonne à $CR(x_i) \leq -\varepsilon$.

Le calcul à l'optimalité du sous-problème n'est pas nécessaire lorsque des colonnes de coût réduits négatif sont exhibées, l'optimalité du sous-problème servant uniquement à la dernière étape pour prouver l'optimalité du PMR. Ce constat permet de diminuer les temps de calcul des premières itérations.

Un point crucial de l'implémentation est la stabilité de la convergence, faire converger l'algorithme en un nombre le plus restreint d'itérations. En effet, sur les premières étapes, on observe en général un comportement erratique des variables duales, ce qui est propre aux méthodes lagrangiennes. Cela conduit à générer aux premières itérations de mauvaises colonnes au final, car basées sur un mauvais signal de valeurs duales. Générer plusieurs colonnes de coût réduit négatif ou partir d'un bon ensemble initial de colonnes peut aider à stabiliser la convergence. Le fond du problème reste que le calcul à l'optimalité du PMR, Programme Linéaire, conduit à des signaux de variables duales extrêmes. Le point clé est la stabilisation des variables duales. Pour ce faire, la méthode des "Box Step" ([93]) contraint les variables duales. D'autres méthodes de stabilisation basent la génération de colonne sur des signaux de variables duales pondérés, on citera [21],[22],[102]. Conceptuellement proche, la génération de colonne "primale-duale" ([63]), calcule le PMR par une méthode de points intérieurs, sans aller à l'optimalité pour les premières itérations.

Nous terminons par quelques idées reçues fréquentes sur la génération de colonne. L'algorithme de génération de colonne s'applique uniquement à des PL, en utilisant la dualité forte. Dans le cas de PLNE, c'est la relaxation linéaire qui est calculée. Les colonnes générées pour calculer la relaxation continue d'un PLNE ne fournissent pas l'optimum du PLNE, juste une borne primale. Notons également que si d'une itération à la suivante, la valeur du PMR ne varie pas, il s'agit d'un cas de dégénérescence, et en aucun cas il s'agit d'un critère de terminaison de l'algorithme de génération de colonne.

1.3.3 Bornes lagrangiennes et décomposition de Dantzig Wolfe



FIGURE 1.4 – Illustration géométrique de la reformulation de Dantzig-Wolfe sur les polytopes projetés

La reformulation de Dantzig-Wolfe permet d'obtenir des bornes duales de meilleure qualité que la relaxation continue du PLNE compact d'origine. Comme illustré sur la figure 1.4, en convexifiant le polyèdre $Bx \geq b$ des solutions fractionnaires sont éliminées. Plus formellement : $\{x \in \mathbb{R}^n, Bx \geq b\} \subset \{x \in \mathbb{R}^n | \exists \lambda_q \in [0, 1]^{|\mathcal{Q}|}, \sum_q \lambda_q = 1, x = \sum_q \lambda_q x_q\} \subset \{x \in \mathbb{N}^n, Bx \geq b\}$. Par cette inclusion, il s'en suit que $Z_{LP}^{compact} \leq Z_{LP}^{DW}$, en notant $Z_{LP}^{compact}$ la borne de relaxation continue de la formulation compacte originelle, et Z_{LP}^{DW} la borne de relaxation continue de la reformulation de Dantzig Wolfe. Dans le cas où on a l'égalité $\{x \in \mathbb{R}^n, Bx \geq b\} = \{x \in \mathbb{R}^n | \exists \lambda_q \in [0, 1]^{|\mathcal{Q}|}, \sum_q \lambda_q = 1, x = \sum_q \lambda_q x_q\}$, les bornes duales $Z_{LP}^{compact}$ et Z_{LP}^{DW} sont identiques. Cela se produit quand les points extrêmes du polytope $\{x \in \mathbb{R}^n, Bx \geq b\}$ sont entiers.

Reformulation de Dantzig-Wolfe et relaxation lagrangienne sont liées, la reformulation de Dantzig-Wolfe permettant de calculer des bornes lagrangiennes. On applique la dualisation lagrangienne à la formulation étendue, de même valeur sur les points entiers, et on obtient les bornes lagrangiennes suivantes :

$$\begin{aligned}
 l(x, \pi) &= \min_{\lambda \in \{0,1\}^{|\mathcal{Q}|}} \sum_q \lambda_q (cx_q) + \pi(a - \sum_q \lambda_q (Ax_q)) \\
 s.c : & \quad \sum_q \lambda_q = 1
 \end{aligned} \tag{1.41}$$

Cette expression est similaire à la valeur optimale du problème esclave à chaque itération : en notant $\xi(\pi) = \min_{x \in \mathbb{N}^n, Bx \geq b} \sigma + (c - \pi A)x$ la valeur optimale du problème esclave, on a $l(x, \pi) = \pi a + \xi(\pi) - \sigma$. En calculant le problème esclave à l'optimalité à une itération, on peut donc en déduire une borne lagrangienne.

1.3.4 Cas partiellement bloc diagonal

La reformulation de Dantzig Wolfe s'applique usuellement au cas où la matrice B est bloc diagonale, les contraintes $Ax \geq a$ comprenant alors les contraintes couplantes. Pour une meilleure convergence de la génération de colonne, on utilise une autre reformulation basée sur la convexification des sous blocs F^k . Pour tout k , on énumère les solutions $\{x \in \mathbb{N}^n, F^k x \geq b^k\} = \{x_q^k\}_{q \in \mathcal{Q}}$, une solution du problème complet se recrée en combinant des colonnes issus de chaque sous-problème. On considère la reformulation suivante :

$$\begin{aligned}
 v(x) &= \min_{\lambda_q^k \in \{0,1\}} \sum_k \sum_q \lambda_q^k (c^k x_q^k) \\
 s.c : & \quad \sum_k \sum_q \lambda_q^k (D^k x_q^k) \geq a \quad (\pi) \\
 \forall k, & \quad \sum_q \lambda_q^k = 1 \quad (\sigma)
 \end{aligned} \tag{1.42}$$

Les colonnes des différents sous-problèmes indépendants se combinent pour former une colonne du problème complet. Pour chaque bloc k , on a le sous-problème de génération de colonne suivant :

$$\begin{aligned}
& \min_x \sigma^k + (c^k - \pi D^k)x \\
s.c : & F^k x \geq b^k \\
& x \in \mathbb{N}^n
\end{aligned} \tag{1.43}$$

Pour chaque sous-problème de valeur strictement négative, on introduit la colonne partielle générée. Si tous ces problèmes ont une valeur positive, les itérations s'achèvent, l'optimum étant donné par le PMR.

1.3.5 Résolution en nombres entiers

Méthode de Branch&Price Pour la résolution d'un PLNE en utilisant la reformulation de Dantzig Wolfe, l'algorithme B&B peut être adapté, en obtenant les bornes à chaque nœud par génération de colonnes, on parle alors de *Branch&Price*. A chaque nœud, les colonnes déjà générées sur les nœuds ancêtres sont valides et permettent un démarrage à chaud. Le choix d'une résolution par Branch&Bound ou Branch&Price est une affaire de compromis : le gain en qualité de borne duale sur la formulation étendue et ses implications sur le nombre de branchements doivent compenser le temps de calcul de relaxation continue à chaque nœud, supérieur sur la formulation étendue.

Branchements en Branch&Price Les branchements en Branch&Price nécessitent un soin particulier. Tout d'abord, des difficultés viennent du fait qu'après branchement, les colonnes restantes après branchements peuvent mener à un calcul de PMR infaisable, sans que le nœud soit irréalizable, et cela ne permet pas de générer de colonnes en suivant l'algorithme. De même que pour l'initialisation d'une décomposition de Dantzig Wolfe, l'ajout de pénalisations permet de contourner de telles difficultés. Les règles de branchements sont cruciales sur la convergence. Brancher sur les variables de la reformulation de DW mène à des arbres de branchement non équilibrés. La stratégie usuelle efficace est de brancher sur l'équivalent des variables de la formulation compacte. Pour une variable compacte qui s'écrit $z_i = \sum_q z_{i,q} \lambda_q$, de valeur fractionnaire calculée \tilde{z}_i , le branchement est réalisé en ajoutant la contrainte $\sum_q z_{i,q} \lambda_q \geq \lceil \tilde{z}_i \rceil$ ou la contrainte $\sum_q z_{i,q} \lambda_q \leq \lfloor \tilde{z}_i \rfloor$. Pour plus de détails sur les branchements en Branch&Price, on se reportera à [122].

Méthode de Branch&Cut&Price De même que pour le Branch&Bound, l'ajout de coupes d'intégrité spécifiques peut être intéressant pour améliorer les bornes duales et couper plus prématurément des nœuds de l'arbre de branchement. Une telle méthode de génération de colonnes et de coupes est alors qualifiée de *Branch&Cut&Price*. Comme expliqué dans [52], cela pose des difficultés lorsque les sous-problèmes sont calculés par un algorithme spécifique, les coupes génèrent des variables duales qui doivent être remontées dans les sous-problèmes, et cela peut changer la nature du sous-problème (qui reste un PLNE). Cela vaut également pour les stratégies de branchement.

1.4 Optimisation linéaire dans l'incertain

En pratique, les coefficients d'un PLNE sont souvent mal incertains, issues de mesures physiques ou d'estimations statistiques. Ces incertitudes peuvent avoir de fortes incidences sur la solution optimale d'un PLNE, qui peut se retrouver infaisable ou avec un coût rédhibitoire dans le problème réel. Cette section fournit les outils d'optimisation dans l'incertain utilisés dans la suite, les approches stochastiques et robustes. Les formalismes de logique floue et de contraintes en probabilité ne sont pas traitées, nous référons à [8] et [13].

1.4.1 Programmation Linéaire Stochastique avec recours

Dans cette section, on supposera connaître la distribution de probabilité des aléas. Nous expliciterons ces approches dans le cas de scénarios probabilistes discrets \mathcal{S} où chaque scénario \mathcal{S} a une probabilité π_s . Dans un tel cadre, la structure de PLNE est préservée, pour le cas général, on se reportera à [78]. On considère un PLNE déterministe de la forme suivante, où les variables y sont soumises aux incertitudes :

$$\begin{aligned} & \min_{x,y} cx + qy \\ \text{s.t. : } & Ax \geq a \\ & Tx + Wy \geq h, \\ & x \in \mathbb{N}^m \times \mathbb{R}_+^n, y \geq 0 \end{aligned} \tag{1.44}$$

Cela intervient naturellement pour des problèmes à deux niveaux. Le premier niveau des variables x est un niveau stratégique, correspondant aux décisions "here and now" soumises aux incertitudes et implémentées avant la connaissance de l'aléa. Le second niveau des variables y correspond aux décisions "wait and see" opérationnelles réalisées après connaissance de l'incertitude, agissant comme recours aux décisions de premier niveau. Le second niveau adresse des questions de coût et de faisabilité, impliquées par les décisions de premier niveau. Le problème stochastique avec recours s'écrit :

$$\begin{aligned} & \min_{x,y^s \geq 0} c^T x + \sum_{s \in \mathcal{S}} \pi_s q^s y^s \\ \text{s.t. : } & Ax \geq a \\ \forall s \in \mathcal{S}, & T^s x + W^s y^s \geq h^s \\ & x \in \mathbb{N}^m \times \mathbb{R}_+^n \end{aligned} \tag{1.45}$$

La décision de premier niveau minimise l'espérance probabiliste des coûts attendus sur le second niveau. Au niveau des contraintes, cela induit la faisabilité des décisions de premier niveau pour tous les aléas. La résolution d'un tel problème rentre dans le cadre de la décomposition de Benders à la section 1.2.3, avec des sous blocs formés par les différents scénarios. Cette méthode dans le cadre stochastique est aussi appelée algorithme L-shaped ([83]). Dans le cadre où des variables de second niveau sont entières, et où toutes les variables de premier niveau sont entières, l'algorithme de résolution "integer L-shaped" ([39]) correspond à l'algorithme développé à la section 1.2.6.

1.4.2 Robustesse en Programmation Linéaire

L'optimisation robuste est une approche alternative de gestion de l'incertain, ne nécessitant pas la connaissance de densités de probabilité. Le terme "robuste" présente des ambiguïtés dans le langage courant, "robuste" est souvent employé pour qualifier une solution résistante à des aléas. Suivant le contexte, la définition de robustesse se doit d'être précisée pour éviter les ambiguïtés. Dans le cadre de la programmation mathématique, nous qualifions de robuste une solution réalisable pour toutes les réalisations considérées des aléas, et dont le coût est optimisé sur la pire réalisation de l'aléa pouvant survenir dans un *ensemble d'incertitude*. La description des incertitudes se restreint au domaine de réalisation prévisionnel des aléas, appelé *ensemble d'incertitude* noté Ω .

Formalisation dans le cadre de la Programmation Linéaire Dans le cadre de la programmation linéaire, nous considérons un PL de la forme suivante :

$$\begin{aligned} \mathcal{P}^{det} = & \min_{x \geq 0} c^T x \\ \text{s.c. : } & Ax \leq b \end{aligned} \tag{1.46}$$

où x est un vecteur colonne de taille n qui représente les variables du problème. Le coût c est un vecteur ligne de taille n , le second membre b est un vecteur colonne de taille m et la matrice A des contraintes de taille $m \times n$. La définition de robustesse invoquée mène au problème min-max suivant :

$$\begin{aligned} \mathcal{P}^{rob} = & \min_{x \geq 0} \max_{\omega \in \Omega} c^T x \\ \text{s.c. : } & \forall \omega \in \Omega, \quad A(\omega)x \leq b(\omega) \end{aligned} \quad (1.47)$$

Cela s'interprète comme un paradigme de théorie des jeux, comme si un adversaire fictif choisissait la réalisation aléatoire après la prise de la décision des variables x . Le problème ci dessus correspond au choix optimal en x , en considérant que l'adversaire fictif choisira la réalisation dans Ω la plus défavorable.

En toute généralité, on peut supposer que l'incertitude porte uniquement sur la matrice des contraintes A . On peut se ramener à un tel cas en ajoutant les variables x_{n+1} et x_{n+2} , avec l'écriture suivante :

$$\begin{aligned} \mathcal{P}^{det} = & \min_{x \geq 0} x_{n+2} \\ \text{s.t. : } & x_{n+1} = 1 \\ \forall i \in \llbracket 1, m \rrbracket, & \sum_{j=1}^n A_{i,j} x_j - b_i x_{n+1} \leq 0 \\ & \sum_{j=1}^n c_j x_j - x_{n+2} \leq 0 \end{aligned} \quad (1.48)$$

Approche robuste de Soyster L'approche de Soyster, introduite en 1973 dans [115], est l'une des premières approches historiques de détermination de solutions robustes d'un PL. L'incertitude est écrite sur la matrice des contraintes A , avec l'hypothèse que les colonnes A_j décrivent des domaines $K_j \subset \mathbb{R}^m$ convexes. Le problème robuste est alors équivalent au problème suivant, où $\bar{A}_j = \max_{A_j \in K_j} A_{i,j}$:

$$\begin{aligned} & \min_{x \geq 0} cx \\ \text{s.t. : } & \sum_j \bar{A}_j \cdot x_j \leq b \end{aligned} \quad (1.49)$$

On est alors ramené au cas déterministe, dans le cas où l'on sait calculer aisément $\max_{A_j \in K_j} A_{i,j}$. Un cas particulier est le cas d'un produit d'intervalle, ie pour tout (i, j) , $A_{i,j} \in [\bar{A}_{i,j}, \underline{A}_{i,j}]$, cela s'écrit :

$$\begin{aligned} & \min_{x \geq 0} cx \\ \text{s.t. : } & \sum_j \underline{A}_{i,j} \cdot x_j \leq b_i, \forall i \end{aligned} \quad (1.50)$$

Approches paramétriques Les solutions robustes selon l'approche de Soyster sont qualifiées de solutions "conservatives" dans la littérature. En effet, la décision est basée sur le cas le plus défavorable, avec une probabilité d'occurrence potentiellement très faible. Des approches paramétriques ont alors été développées pour contrôler le degré de "conservatisme" de la solution robuste tout en garantissant une forte probabilité de satisfaction des contraintes.

L'ensemble d'incertitude adopté pour modéliser les coefficients de la matrice A est un modèle par intervalles : pour tout $i \in \llbracket 1; m \rrbracket$ et $j \in \llbracket 1; n \rrbracket$, le paramètre $A_{i,j}$ appartient à $[\hat{A}_{i,j} - \hat{\Delta}_{i,j}, \hat{A}_{i,j} + \hat{\Delta}_{i,j}]$, où $\hat{A}_{i,j}$ représente la valeur nominale du coefficient $A_{i,j}$ et $\hat{\Delta}_{i,j} \geq 0$ est sa déviation maximale. En d'autres termes, $A_{i,j} = \hat{A}_{i,j} + \xi_{i,j} \hat{\Delta}_{i,j}$ avec $\xi_{i,j} \in [-1, 1]^{n \times m}$. L'approche paramétrique de Bertsimas et Sim introduite dans [30] introduit des paramètres Γ_i pour chacune des contraintes $i \in \llbracket 1; m \rrbracket$, qui représentent la somme des déviations totales par rapport aux valeurs nominales de tous les coefficients incertains de la même contrainte i . Il s'agit d'un modèle d'incertitude en lignes, là où l'approche de Soyster était un modèle d'incertitude en colonnes. L'ensemble d'incertitude noté $\Phi_i^{BS}(\Gamma_i)$ est alors $\Phi_i^{BS}(\Gamma_i) = \{\xi_{i,j} \in [-1, 1]^n \mid \sum_j |\xi_{i,j}| \leq \Gamma_i\}$. Le problème robuste s'écrit alors :

$$\begin{aligned}
\mathcal{P}^{rob} = & \min_{x \geq 0} \sum_{j=1}^n c_j x_j \\
s.c : \forall i \in \llbracket 1, m \rrbracket, & \sum_{j=1}^n \hat{A}_{i,j} x_j + \max_{\xi_{i,j} \in \Phi_i^{BS}(\Gamma_i)} \sum_{j=1}^n \xi_{i,j} \hat{A}_{i,j} x_j \leq b_i
\end{aligned} \tag{1.51}$$

Ainsi, pour une contrainte i , la valeur de Γ_i permet de contrôler la déviation totale des paramètres incertains de leur valeur nominale. Notons que Γ_i n'est pas nécessairement un entier, mais à valeurs dans l'intervalle $[0, n]$. Pour $\Gamma_i = 0$, aucune déviation n'est autorisée sur les coefficients de la contrainte i et celle-ci est équivalente à la contrainte nominale. Par contre, si $\Gamma_i = n$ tous les paramètres sont susceptibles de dévier, et l'on revient à la formulation pire cas de Soyster. En dualisant le problème de maximisation présent dans la matrice des contraintes, il est prouvé dans [30] que l'on se ramène au PL suivant :

$$\begin{aligned}
\mathcal{P}^{rob} = & \min_{x, \pi, \lambda \geq 0} \sum_{j=1}^n c_j x_j \\
s.c : \forall i \in \llbracket 1, m \rrbracket, & \sum_{j=1}^n \hat{A}_{i,j} x_j + \pi_i \Gamma_i + \sum_{j=1}^n \lambda_{i,j} \leq b_i \\
\forall i \in \llbracket 1, m \rrbracket, \forall j \in \llbracket 1, n \rrbracket, & \pi_i \lambda_{i,j} \geq \hat{A}_{i,j} x_j
\end{aligned} \tag{1.52}$$

L'approche "Multiband Robust Optimization" introduite dans [37] a permis d'étendre les résultats et la modélisation de Bertsimas et Sim, en considérant des modèles d'incertitude basés sur des histogrammes.

Cas particuliers Si on a un cadre commun, les problématiques spécifiques de robustesses peuvent être mises à profit. Dans le cas où seule la fonction objectif présente des incertitudes, le domaine de faisabilité robuste est le même que le problème déterministe. Dans un tel cas, d'autres critères que le pire cas ont été développés avec une résolution efficace, on citera le critère de regret maximal, cf [16].

Dans de nombreuses applications, l'incertitude affecte uniquement le second membre, via des contraintes de demandes ou de ressources. Dans ce cas, l'approche robuste naturelle est l'approche d'incertitude en colonnes de Soyster, où seule la colonne du au second membre définit une incertitude non nulle. Réduire le conservatisme en appliquant l'approche de Bertsimas et Sim est sans issue : le budget d'incertitude de la ligne étant consommé uniquement sur le second membre, cela revient à réduire directement l'incertitude correspondant au second membre. Ce cas applicatif courant a conduit à des travaux spécifiques dans [106].

1.4.3 Approches robustes sur des problèmes discrets

Les progrès réalisés par la résolution PLNE ces dernières décennies, ont permis d'envisager l'extension de l'optimisation robuste aux PLNE. Le modèle d'incertitude robuste de Bertsimas et Sim a été étendu dans l'article [29] aux problèmes discrets. Le modèle robuste correspondant à un PLNE déterministe est toujours un PLNE, utilisé pour une implémentation générique dans [79]. Sur le cas particulier l'incertitude porte uniquement sur la fonction de coût et où les n variables sont toutes binaires, la résolution se ramène à résoudre $n + 1$ instances du problème d'origine. Il en résulte que si le problème déterministe est polynomial avec un algorithme spécifique, le problème robuste est également polynomial. [14] s'est intéressé à des reformulations renforcées dans un tel cadre. Ces résultats sont également valables avec l'extension Multiband Robust Optimization de [37].

Auparavant, l'article [82] avait étudié l'état de l'art des problèmes robustes discrets avec des modèles d'incertitudes basés sur des scénarios discrets. Dans un tel cadre, la majorité des problèmes discrets polynomiaux ont une extension robuste devient NP-difficile.

1.4.4 Approches robustes multi-étapes

Dans le contexte multi-étapes, les décisions du problème sont fractionnées en plusieurs groupes, suivant l'instant où l'incertitude est révélée. Les approches robustes bi-étapes, introduites de manière générales dans [23], [95] et [96], sont fortement inspirées de l'optimisation stochastique avec recours. Les décisions de la première étape sont définies telles qu'il existe toujours un recours à la seconde étape. Le problème de recours est dans ce cas basé sur la pire évaluation du problème compte tenu des incertitudes, donnant un problème de type min max min. Les approches robustes bi-étapes s'écrivent sur des problèmes déterministes de la forme suivante, où les décisions relatives aux variables x sont mises en œuvre avant la réalisation de l'aléa, tandis que les variables y sont mises en œuvre après la réalisation de l'aléa :

$$\begin{aligned} & \min_{x \in \mathbb{N}^m \times \mathbb{R}_+^n, y \geq 0} cx + qy \\ \text{s.t. : } & Ax \geq a \\ & Tx + Wy \geq h \end{aligned} \tag{1.53}$$

En définissant Ω l'ensemble d'incertitude, le problème robuste à deux étapes s'écrit :

$$\begin{aligned} & \min_{x \in \mathbb{N}^m \times \mathbb{R}_+^n} cx + \mathcal{Q}(x) \quad \text{avec} \quad \mathcal{Q}(x) = \max_{\omega \in \Omega} \min_{y \geq 0} q(\omega)y \\ \text{s.c. : } & Ax \geq a \qquad \qquad \qquad \text{s.c. : } T(\omega)x + W(\omega)y \geq h(\omega) \end{aligned} \tag{1.54}$$

Cela s'interprète à nouveau comme un paradigme de théorie des jeux, comme s'il existait un adversaire fictif qui choisissait la réalisation aléatoire qui serait la plus défavorable après la prise de la décision des variables x , sachant que nous pourrions réagir au coup suivant avec le recours en jouant une réalisation des variables y . Le problème $\mathcal{Q}(x)$ de calcul de perturbation maximale de l'adversaire fictif est ainsi un problème de type max-min. Le problème ci-dessus correspond au choix optimal en x , au premier coup de ce jeu, avant que les décisions suivantes soient prises de manière optimale après la décision x .

Le cadre applicatif usuel de la littérature est un ensemble d'incertitude polyédral, avec l'incertitude portant uniquement sur le second membre. Ce formalisme se retrouve ainsi en ordonnancement PERT [95], dans [106] pour une application à un problème de localisation-transport, dans [15] pour un problème de design de réseau télécom, et dans [28] pour un problème de type Unit Commitment Problem.

Pour ce type de problème, une méthode de résolution est proposée dans [120] avec les hypothèses que seul le second membre est soumis à des aléas, et que l'incertitude prend des valeurs continues, dans Ω un polyèdre, et où les variables du second niveau sont aussi continues. Cette approche consiste alors à utiliser la dualité pour transformer le dernier problème de minimisation en problème de maximisation, et à traiter le problème min-max par une approche de plan coupant en utilisant l'algorithme de Kelley. Cette approche revient à une génération dynamique de scénarios, sur les points extrêmes du domaine d'incertitude. Si le nombre de scénarios était fixé, cela serait analogue à la décomposition de Benders.

1.4.5 Autres approches robustes en ordonnancement

Dans un contexte opérationnel, "robuste" signifie souvent être résistant en terme de faisabilité aux aléas considérés dans l'ensemble d'incertitude. En ce sens, des solutions issues de l'approche stochastique, garantissent la réalisabilité de la solution sur tous les scénarios stochastiques. La robustesse est aussi souvent interprétée comme la présence de marges dans des contraintes. La définition de robustesse se doit d'être systématiquement définie avec précision pour lever toute ambiguïté.

De nombreux travaux ont introduit un tel cadre de robustesse pour des problèmes d'ordonnancement, on se reportera à [69]. La robustesse est utilisée souvent dans le cadre de la réoptimisation dynamique d'un problème d'ordonnancement selon l'avancée du temps et de la réalisation d'aléas. Les approches proactives

se basent sur une solution courante de fil rouge calculée de manière robuste, de manière à être le plus résistante possible a priori, avant de s'adapter à la réalisation d'aléas. Une approche robuste suivant l'état de l'art de 1.4.3 ou 1.4.4 s'avère souvent limitée sur les contraintes de temps de calcul et sur les tailles de problèmes exigées. L'enjeu est alors de développer des approches robustifiées praticables, fournissant des solutions plus robustes que la solution déterministe sans aléas. Nous référons à [56] pour un tel travail dans le cadre de la planification des horaires de train.

De manière générale, la notion de "Light robustness" développée dans [55], répond à de telles considérations. Considérant la solution déterministe sans aléa, il s'agit pour un coût de solution donné supérieur, de fournir la solution de coût inférieur à ce seuil la plus robuste, ie maximisant les marges sur la faisabilité, écrites des variables d'écarts supplémentaires. Cela peut s'interpréter comme une décomposition hiérarchique du problème min-max de 1.4.2, avec des résolutions similaires au problème déterministe.

Dans le cas où un algorithme de programmation dynamique est utilisé pour la résolution déterministe, une définition robuste analogue à la section 1.4.2 suivant un formalisme min-max peut être résolu en étendant la programmation dynamique, comme expliqué dans l'article [72]. Cela permet de traiter le cas non considéré précédemment, où les incertitudes ne prennent que des valeurs discrètes. Des applications à un tel formalisme se trouvent en pratique dans la planification des horaires de trains traitée dans [7], ou dans l'ordonnancement de pistes d'aéroports, nous référons alors à [41].

1.5 Heuristiques

Les méthodes heuristiques s'utilisent pour trouver des solutions primales de bonne qualité sans preuve d'optimalité, ni même de majoration de l'écart à la solution optimale. Il existe une large taxonomie d'heuristiques, nous en faisons un aperçu général dans cette section. On distingue tout d'abord les heuristiques constructives, des heuristiques perturbatives, suivant que l'on dispose ou non d'une solution initiale. Enfin, on analysera le lien et les hybridations entre approches heuristiques et exactes.

1.5.1 Heuristiques constructives

Les heuristiques constructives fournissent rapidement une solution réalisable pour toutes les contraintes sans partir d'une solution réalisable. On traite dans cette section des algorithmes gloutons et de la variante GRASP, ainsi que des techniques de clustering. L'algorithme des colonies de fourmis (ACO) est également constructif, et sera traité dans la section 1.5.3.

Algorithme glouton L'algorithme glouton construit progressivement une solution, par choix locaux optimaux pour ajouter un morceau de solution à la solution partielle courante. Cela ne garantit pas en général de construire une solution globalement optimale, ni même in fine une solution réalisable. Dans le cas du problème du sac à dos défini par (1.2), l'algorithme glouton trie les objets suivant leur rapport utilité/masse décroissant, et ajoute des objets le long de la liste ordonnée tant que la capacité M du sac à dos le permet. Pour le calcul de la relaxation continue, en considérant l'objet i' limite, ie le premier objet qui ne rentre plus entièrement dans le sac à dos, ie $m_{i'}$ dépasse la capacité résiduelle du sac à dos, en affectant à la variable $x_{i'}$ la valeur fractionnaire maximale pour respecter la contrainte de capacité, et des valeurs nulles aux autres variables, l'algorithme glouton a fourni une solution optimale du problème, en temps linéaire. Dans la version PLNE, en affectant les objets qui peuvent rentrer dans la capacité résiduelle du sac à dos selon les rapports c_i/m_i décroissants, on obtient une heuristique gloutonne non optimale.

Algorithme GRASP L'algorithme GRASP (Greedy Randomized Adaptive Search Procedure) est une métaheuristique qui combine les méthodes gloutonnes et aléatoires. La construction d'une solution se

déroule par étapes et à chacune de celles-ci, l'ensemble des solutions partielles qu'il est possible d'ajouter est placé dans une liste appelée RCL (Restricted Candidate List). Cette liste est triée, c'est la partie gloutonne de l'algorithme. Ce n'est pas nécessairement la meilleure solution partielle qui est ajoutée à la solution courante, on tire aléatoirement parmi les meilleurs possibilités celle qui sera sélectionnée.

Heuristiques de preprocessing/clustering La taille des instances peut s'avérer rédhibitoire pour des méthodes exactes comme pour des heuristiques constructives, pour de grands problèmes d'optimisation. Des techniques de preprocessing/clustering permettent de réduire la taille des instances à des problèmes résolubles avec un algorithme constructif. Les techniques de preprocessing permettent de réduire les instances, en fixant des variables. Une approche exacte de preprocessing a déjà été citée dans [113], fixant des variables à leur valeur à l'optimum. Une fixation de variables peut être heuristique, comme dans [51] et [91]. Les techniques de clustering morcellent un grand problème en plus petits sous-problèmes indépendants, sur lesquels la résolution est envisageable par un algorithme constructif. La solution est obtenue en recollant les diverses solutions partielles engendrées par les sous-problèmes. L'algorithme POPMUSIC (Partial Optimization Metaheuristic under Special Intensification Conditions ([116])) rentre dans un tel cadre.

1.5.2 Recherche locale

Un algorithme de recherche locale part d'une solution initiale et la déplace de façon itérative vers une solution voisine, jusqu'à la réalisation d'un critère d'arrêt. Ces algorithmes s'articulent autour de notions communes : les voisinages, la diversification, l'intensification, et des mécanismes de mémoire. Le *voisinage* d'une solution est le sous-ensemble de solutions atteignables par transformations locale, obtenues en général par permutations, éjections ou optimisations partielles. Le point crucial de la définition d'un voisinage est de pouvoir calculer rapidement les coûts des solutions du voisinage, la capacité d'exploration d'une recherche locale en dépend fortement. Les notions d'*intensification* et de *diversification* sont cruciales dans la conception des méta-heuristiques, pour atteindre un équilibre entre ces deux dynamiques de recherche. La diversification (ou exploration) récolte de l'information sur le problème, tandis que l'intensification (ou exploitation) vise à utiliser l'information récoltée pour parcourir les zones intéressantes de l'espace de recherche. La mémoire est le support de l'apprentissage, pour éviter les zones où l'optimum global n'est pas susceptible de se trouver. A un voisinage donné, "hill climbing" (escalade de colline) ou "steepest descent" (descente la plus abrupte) procèdent en maximisant (ou minimisant) le critère sur le voisinage, sans mémoire ni diversification, convergeant vers un optimum local, aucune solution du voisinage ne permet d'améliorer le critère. Dans un tel cadre, utiliser de grands voisinages permet d'avoir des optimums locaux de meilleure qualité.

Recherche avec tabous La recherche avec tabous a été introduite en 1986 dans l'article [59] et développée par la suite, on citera [60] comme ouvrage de référence. La recherche avec tabous se distingue de l'heuristique hill climbing par la notion de mémoire, qui permet de sortir et de ne pas revenir dans des extrema locaux. Les positions déjà explorées sont conservées dans une file FIFO (appelée liste tabou) d'une taille donnée (paramètre ajustable). L'idée de la recherche tabou consiste, à partir d'une position donnée, à explorer le voisinage et à choisir la position non interdite par la liste tabou dans ce voisinage qui minimise la fonction objectif. Dans le cas où tous les points admissibles du voisinage ont une valeur plus élevée, la valeur du critère est dégradée. C'est à partir de ce mécanisme que l'on sort d'un minimum local.

Algorithme de recuit simulé L'algorithme de recuit simulé a été introduit dans l'article [81] par analogie au refroidissement naturel en sidérurgie. La température est utilisée comme paramètre de commande, le recuit consiste à chauffer le matériau, lui conférant une énergie élevée, avant de refroidir lentement le

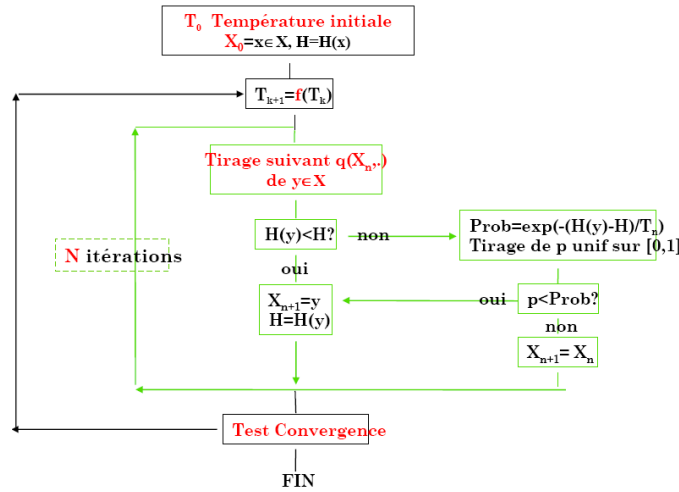


FIGURE 1.5 – Algorithme de recuit simulé

matériau. À la température T , l'algorithme de recuit utilise que la probabilité pour un système physique de posséder l'énergie donnée E est proportionnelle au facteur de Boltzmann $\exp\left(-\frac{E}{k_B T}\right)$, k_B étant la constante de Boltzmann. Le recuit simulé progresse comme illustré sur la figure 1.5. À l'itération k , où la solution courante à une valeur H , on actualise la température T_k , et on tire aléatoirement une solution y dans le voisinage de la solution courante. Si $H(y) \leq H$, y devient la solution courante. Sinon, on effectue un tirage aléatoire de loi binomiale de paramètre $p = \exp\left(\frac{H-H(y)}{T_k}\right)$. Si la réalisation aléatoire est 1, la solution courante est acceptée, ce qui fait remonter le coût de solution et diversifie la recherche. Les phases de recuits et de descente de température sont paramétrées par la loi d'évolution de T_k . A haute température, $\exp\left(\frac{H-H(y)}{T_k}\right) \approx 1$, la plupart des mouvements sont acceptés, l'algorithme équivaut à une marche aléatoire. À basse température, $\exp\left(\frac{H-H(y)}{T_k}\right) \approx 0$, la plupart des mouvements augmentant l'énergie sont refusés, l'algorithme se ramène à des itérations "hill climbing". Le recuit simulé converge en probabilité vers un optimum global, sous les hypothèses d'ergodicité (toute solution peut être atteinte avec une probabilité non nulle sur une marche aléatoire basée sur les voisinages), avec une loi de température où T_k décroissant au plus vite en $O\left(\frac{1}{\log(k)}\right)$. La loi de décroissance logarithmique de la température n'est pas utilisée en pratique, car inefficace. Plusieurs types de parallélisation ont été envisagés pour le recuit simulé, on se reportera à [1] et [108]. L'intérêt de la parallélisation est qu'à basse température, les mouvements acceptés deviennent rares, et aucune acceptation peut survenir sur des processus indépendants.

Algorithme VNS Les heuristiques VNS (Variable Neighbourhood Search), ont été introduites en 1997 dans [97], et se sont rapidement développées avec succès, nous nous référons à [98] et [67] pour une revue détaillée des variantes de VNS développées et des problèmes d'applications. L'idée de l'algorithme VNS est de considérer différents types de voisinages et d'en changer systématiquement au cours de la recherche locale. On sort d'un extremum local pour un voisinage donné en changeant de voisinage, la solution courante n'étant plus forcément un extremum local du nouveau voisinage considéré. Par cette méthode, un extremum local est un extremum local de tous les voisinages considérés. Cela induit donc moins d'extremums locaux et de meilleure qualité que dans le cas d'une recherche locale simple.

La variante la plus simple est Variable Neighborhood Descent (VND), une exploration déterministe sans diversification, où l'on choisit toujours la meilleure solution du voisinage courant, le choix des voisinages étant également imposé par une règle déterministe. Il s'agit d'un algorithme de type "steepest descent". Le critère d'arrêt de la recherche locale peut être qu'aucun voisinage ne permettent d'améliorer la solution, si bien que la solution courante est un optimum local pour tous les types de voisinages.

1.5.3 Métaheuristiques de populations

Dans cette section, nous introduisons des métaheuristiques manipulant des populations de solutions. Ces approches permettent un haut degré de diversification, et une parallélisation naturelle. Un autre intérêt de ces approches est de pouvoir aboutir à un front de Pareto de solutions dans le cadre de l'optimisation multicritère. Nous présentons les algorithmes évolutionnaires, et l'algorithme de colonie de fourmis.

Algorithmes évolutionnaires Les algorithmes évolutionnaires s'inspirent de la théorie de l'évolution des êtres vivants, en considérant que la nature tend à produire et sélectionner des organismes les plus adaptés à leur environnement. Un algorithme évolutionnaire doit être initialisé avec une première population d'individus, pour itérer selon des étapes et opérateurs de sélection, croisements, mutations, suivant des processus aléatoires. La première étape de *sélection* permet de séparer les individus qui participeront à la reproduction de ceux qui n'y participeront pas. Les individus sélectionnés (les "parents") se reproduisent (on dit aussi que l'on effectue des *croisements*), donnant un ensemble d'"enfants" partageant une partie des caractéristiques de leurs ascendants. Ces enfants subissent alors une étape de *mutation*, qui modifie aléatoirement leur génotype. Les nouveaux individus sont alors évalués par la fonction objectif. Enfin, on choisit un nombre d'individus déterminé parmi l'ensemble parents + enfants, pour former la génération suivante. Dans le cadre des algorithmes évolutionnistes, nous citerons les algorithmes génétiques, les stratégies d'évolution, les algorithmes à évolution différentielle, les algorithmes à estimation de distribution, les systèmes immunitaires artificiels, et l'algorithme BAO (Bayesian Optimization Algorithm).

Colonies de fourmis L'algorithme par optimisation de colonies de fourmi (en anglais ACO, Aint Colony Optimization) a été introduit en 1996 dans [47], sur un problème de plus court chemin dans un graphe, et a depuis été étendu comme une méta-heuristique naturellement parallélisable pour de nombreux problèmes d'optimisation, nous référons à [48] et [119]. L'idée originale vient de l'observation de l'exploitation des ressources alimentaires chez les fourmis. Une colonie de fourmis ayant le choix entre plusieurs chemins menant à une source de nourriture a tendance à utiliser le chemin le plus court. L'algorithme ACO est à la fois constructif et une recherche locale perturbative suivant une cartographie de phéromones. ACO se base sur la collaboration des individus entre eux, et s'appuie sur le concept d'auto-organisation, un groupe d'individus peu intelligents peut posséder une organisation globale complexe. Ce système repose sur des rétroactions positives (le dépôt de phéromone attire d'autres fourmis qui vont la renforcer à leur tour) et négatives (la dissipation de la piste par évaporation empêche le système de s'emballer).

La méthode ACO construit des solutions à partir des éléments qui ont été explorés par d'autres individus. Pour reprendre la dénomination naturelle, les fourmis se déplacent à la recherche de solutions et sécrètent des phéromones, qui indiqueront à leurs congénères si un chemin est intéressant ou non. En effet, les chemins fortement phéromonés indiquent les chemins vers de bonnes solutions, car traversés à une fréquence plus importante. Chaque fois qu'un individu découvre une solution, bonne ou mauvaise, il enrichit la connaissance collective de la colonie. Un tel schéma se classe dans les techniques d'exploration.

Une diversification est incorporée au modèle pour diminuer le risque d'enfoncer la colonie dans un minimum local, via un modèle d'évaporation des phéromones. Cela permet d'explorer des solutions moins marquées que la solution courante, pour rehausser l'intérêt des autres chemins. Le taux d'évaporation est un premier paramètre de diversification. On peut aussi contrôler le niveau de phéromone de chaque chemin pour le maintenir entre des bornes minimum et maximum, et éviter l'existence de chemins inondés de phéromones ou totalement invisibles, fournissant de nouveaux paramètres de diversification.

1.5.4 Heuristiques hybrides

On parle d'hybridation quand une méta-heuristique est composée de plusieurs méthodes qui se répartissent les tâches de recherche locale. Les méta-heuristiques sont propices aux hybridations, pour combiner les avantages de diverses méthodes. La taxonomie des méta-heuristiques hybrides se sépare en deux parties : une classification hiérarchique et une classification plate. On se reportera à [118] comme ouvrage de référence sur les hybridations des méta-heuristiques.

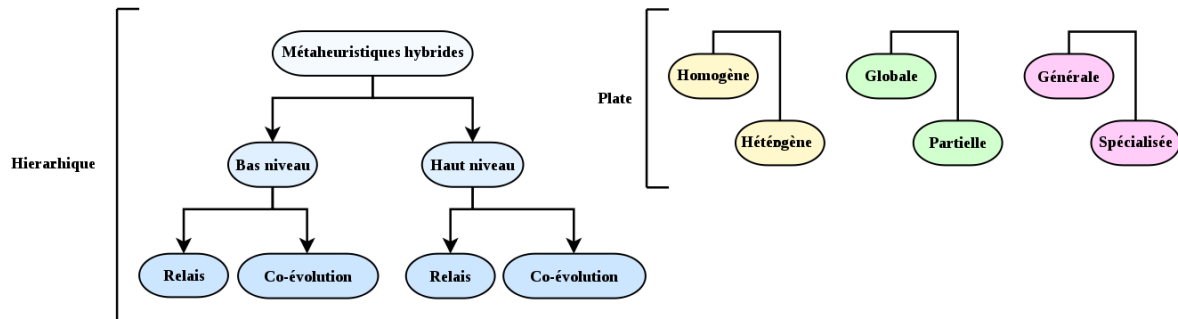


FIGURE 1.6 – Taxonomie des méta-heuristiques hybrides

La classification hiérarchique se fonde sur le niveau (bas ou haut) de l'hybridation et sur son application (en relais ou concurrente). Une hybridation de bas niveau remplace une fonction donnée d'une méta-heuristique. Dans le cas du haut niveau, le fonctionnement interne des méta-heuristiques n'est pas modifié. Dans une hybridation en relais, les méta-heuristiques sont lancées les unes après les autres, chacune prenant en entrée la sortie produite par la précédente. Dans la concurrence (ou coévolution), chaque algorithme utilise une série d'agents coopérants ensemble. Au bilan, quatre classes générales sont dégagées : les hybridations *bas niveau et relais* (abrégé LRH en anglais), les hybridations *bas niveau et coévolution* (abrégé LCH), les hybridations *haut niveau et relais* (HRH) et les hybridations *haut niveau et coévolution* (HCH).

Une seconde classification des hybridations peut être réalisée sur les caractéristiques d'homogénéité, de globalité ou de spécialisation. Si l'hybridation se fait entre plusieurs instances d'une même méta-heuristique, elle est dite *homogène*, sinon elle est dite *hétérogène*. Si les méthodes recherchent dans tout l'espace de recherche, on parlera d'*hybridation globale*, si elles se limitent à des sous-parties de l'espace, d'*hybridation partielle*. Si les algorithmes mis en jeu traitent tous le même problème, on parlera d'*hybridation générale*, s'ils sont lancés sur des problèmes différents, d'*hybridation spécialisée*.

1.5.5 Heuristiques et méthodes exactes en PLNE

Les méthodes exactes et heuristiques sont des approches complémentaires qui permettent de s'enrichir mutuellement : les méthodes exactes peuvent être utilisées avec succès dans un schéma d'hybridation, tandis que les heuristiques peuvent accélérer la convergence de méthodes exactes. De telles hybridations ont été engendré le terme de *matheuristique*. Nous traitons dans cette section les hybridations entre des méta-heuristiques et des méthodes de PLNE intervenant dans la suite de la thèse. La programmation dynamique présente également un potentiel hybride, nous référons à [44]) et [73].

Heuristiques primales en Branch&Bound L'amélioration des heuristiques primales a permis des avancées pratiques. Trouver plus rapidement d'excellentes solutions permet de couper plus tôt des branches de l'arbre de Branch&Bound, ce qui accélère la convergence. En outre, un solveur de PLNE peut alors être utilisé dans une visée heuristique en temps contraint. Les heuristiques primales du Branch&Bound sont à la fois constructives et perturbatives. On pourra se reporter à [27] pour plus de détails. Ces idées ont été

depuis étendues en Branch&Price dans [75].

La construction naturelle de solutions en Branch&Bound est d'effectuer des branchements jusqu'à obtenir une solution entière, ce qui revient à plonger dans l'arbre de Branch&Bound jusqu'à obtenir une feuille. Des telles heuristiques de plongements "diving heuristics", sont incorporées dans les règles de branchements, la fréquence de recherche de solutions par plongement est un paramètre de la résolution Branch&Bound. Les dernières versions de Cplex utilisent la parallélisation pour sélectionner les meilleurs branchements à considérer à partir d'une même situation, en permettant de collecter les meilleures informations des différentes explorations. En particulier, en utilisant des règles de branchements orientées sur la recherche de solutions en parallèle de branchements plus équilibrés pour la convergence globale permet de mieux utiliser les heuristiques de plongements. De telles heuristiques sont insuffisantes, il est avantageux d'avoir des bonnes solutions avant tout branchement. Disposer initialement d'une bonne solution, est souvent intéressant comme démarrage à chaud, ce qui fournit une première hybridation méta-heuristique/Branch&Bound.

Les heuristiques constructives "rounding heuristics" s'appuient sur la relaxation continue, pour déduire du PLNE initial un petit PLNE après avoir fixé toutes les variables entières (ou suffisamment proches d'entiers à la définition d'un seuil près) de la relaxation continue, pour lequel un petit calcul de Branch&Bound permet d'obtenir rapidement une solution. Une telle approche est souvent inefficace, en particulier en menant à des PLNE infaisibles. Une variante plus élaborée, "Feasibility Pump" (FP), a été introduite introduite dans [53] pour des PLNE à variables binaires, améliorée dans [4], et généralisée à tous les PLNE dans [26]. Feasibility Pump cherche à partir du vecteur de solution continue la solution entière la plus proche (au sens d'une distance définie avec la norme 1). Mathématiquement, pour le PLNE $\min_{Ax \geq b} cx$ où les variables x sont binaires de solution relâchée continue \tilde{x} , on considère le problème d'optimisation $\min_{Ax \geq b} \Delta(x, \tilde{x})$, où $\Delta(x, \tilde{x}) = \sum_i |x_i - \tilde{x}_i|$. Si un tel PLNE ne réduit pas l'ensemble des solutions réalisables, l'orientation de la fonction objectif rend la résolution Branch&Bound bien plus efficace, avec une exploration guidée autour de la solution continue. La linéarisation définit pour tout indice $j \in \mathcal{J}$ tel que $0 < \tilde{x}_j < 1$, des variables $x_j^+, x_j^- \geq 0$, telles que $x_j = \tilde{x}_j + x_j^+ - x_j^-$:

$$\begin{aligned} \min_{x \in \{0,1\}^m, x_j^+, x_j^- \geq 0} \quad & \sum_{i: \tilde{x}_i=0} x_i + \sum_{i: \tilde{x}_i=1} (1 - x_i) + \sum_{j \in \mathcal{J}} (x_j^+ + x_j^-) \\ \text{s.c :} \quad & Ax \geq b \\ \forall j \in \mathcal{J} \quad & x_j = \tilde{x}_j + x_j^+ - x_j^- \end{aligned} \tag{1.55}$$

Une fois qu'une solution primale est connue, des heuristiques perturbatives permettent d'améliorer la solution courante, à tout nœud de l'arbre de Branch&Bound. Nous citons tout d'abord l'heuristique Relaxation Induced Neighborhood Search (RINS) introduite dans [46]). L'idée est d'utiliser non seulement la solution de la relaxation continue, mais également la meilleure solution trouvée. Les variables communes à ces deux vecteurs sont fixées, et un petit calcul de PLNE est effectué sur les variables restantes, avec un faible temps de calcul alloué. L'heuristique "Local branching", introduite dans [54], est une recherche locale utilisant uniquement la solution courante. Les voisinages sont définis par une distance à la solution courante, en considérant un paramètre $k \in \mathbb{N}$ indiquant le nombre maximal de modifications par rapport à la solution courante. Similairement à l'approche "Feasibility Pump", cela se ramène à une formulations PLNE où la combinatoire est réduite par rapport au problème initial. Dans le cas d'un PLNE avec des variables binaires x_i for $i \in \mathcal{I}$ ayant la valeur x_i^0 dans la solution courante, les voisinages sont définis en ajoutant les contraintes linéaires : $\sum_{i: x_i^0=1} x_i + \sum_{i: x_i^0=0} (1 - x_i) \leq k$. Ces voisinages paramétriques ont pu être incorporés dans une VNS pour une meilleure efficacité dans [99]. Variable Neighborhood Decomposition Search (VNDS) a également abouti à des heuristiques primales efficaces, nous référons à [84].

Apport d’heuristiques dans la stabilisation d’une méthode de décomposition Dans le cadre d’une implémentation d’un schéma de décomposition, les approches heuristiques peuvent fournir des apports considérables en terme de temps de calcul (en réduisant les temps de calcul pour chaque itération) et de stabilisation (en diminuant le nombre d’itérations nécessaires à la preuve de la convergence).

Dans le cas d’un schéma de génération de colonne, les premières itérations servent uniquement à générer des colonnes de coût réduit négatif. Une heuristique fournissant rapidement de telles colonnes permet de limiter le temps de calcul des problèmes esclaves, prépondérants dans le temps de calcul de l’algorithme complet. Les heuristiques permettent également de diminuer le nombre d’itérations nécessaires, en générant plusieurs colonnes de coût réduit négatif. Une telle stabilisation est réalisée dans [94]. Dans ce cas, l’utilisation de méta-heuristiques de populations est également intéressante, Fournir de meilleures solutions initiales dans le processus de génération de colonne, avoir une population de solutions initiales réalisables permet d’avoir une meilleure approximation du dual.

Dans le cadre de la décomposition de Benders, l’utilisation de la PPC peut remplacer avantageusement un solveur de PLNE dans le calcul du problème maître, suivant l’approche mise en œuvre dans [25] pour un problème d’ordonnancement où les question de faisabilités s’avèrent prépondérantes. La PPC peut aussi être utilisée avantageusement pour renforcer des coupes de Benders, notamment pour détecter un ensemble minimal de variables engendrant un conflit, comme dans l’équation (1.33). On se reportera à [110] pour de telles applications appliquées à des coupes de Benders, et à [3] pour un cadre plus général en PLNE.

Apport de méthode exactes de PL/PLNE dans une méta-heuristique Les approches exactes peuvent se dériver en une méta-heuristique de multiples manières. Une première possibilité pour repousser les limites de calcul des approches exactes est de fixer préalablement des variables de manière heuristique pour dériver un calcul heuristique où l’approche exacte est envisageable, comme dans [51]. Les approches exactes sont aussi très bien adaptées pour une hybridation de bas niveau interne à une méta-heuristique, pour traiter des plus petits sous-problèmes. Les approches de type POPMUSIC ([116]) permettent de réduire les espaces de recherches à fournir à une approche exacte, ce qui fournit naturellement une hybridation partielle. Tout opérateur d’une méta-heuristique peut être remplacé par une recherche optimale réalisée par une méthode exacte, pour fournir des hybridations spécialisées. [45] remplace des opérateurs d’algorithmes évolutionnaires par de la recherche de solutions B&B. L’article [114] opère le B&B dans une recherche locale VNS, ce qui permet de définir de grands voisinages.

Les relaxations fournies par des méthodes exactes peuvent aussi servir à guider des méta-heuristiques. La relaxation lagrangienne était ainsi souvent utilisée dans un tel objectif, avant les progrès des capacités de calculs et algorithmiques permettant l’implémentation de Branch&Price efficaces. On parle alors d’*heuristique lagrangienne*, on référera à [18] pour une hybridation avec un algorithme particulière PSO. Plus récemment, il a été développé dans [104] un algorithme génétique pour le problème d’affectation ”generalized assignment”, où la relaxation continue est exploitée pour créer une population de solutions initiales par une procédure d’arrondis et après des procédures de réparation et d’améliorations locales.

Chapitre 2

Gestion de production électrique

Résumé : Ce chapitre introduit le contexte industriel de la gestion de production d'électricité, et le contexte académique de l'Unit Commitment Problem, avant de présenter les problèmes d'applications de la thèse et leur état de l'art spécifique. Le premier problème d'application, UCPd, se distingue de l'UCP académique par les contraintes de durées minimales sur des paliers et des puissances discrètes. Le second problème d'application s'intéresse à la planification des arrêts de maintenance et de rechargement des centrales nucléaires. Ce dernier problème a fait l'objet du challenge EURO/ROADEF 2010.

Ce chapitre introduit le contexte industriel et académique de gestion de production d'électricité, dans lequel s'inscrivent les problèmes d'application de la thèse. Après avoir décrit le contexte national de production d'électricité, nous introduisons l'Unit Commitment Problem (UCP), le problème académique d'optimisation de la production d'électricité, et l'état de l'art de sa résolution. Nous définissons ensuite les problèmes d'applications de la thèse : un UCP discret modélisant des unités thermiques à horizon journalier et la planification des arrêts de maintenances et de rechargement des centrales nucléaires à un horizon pluriannuel.

2.1 Contexte national de production d'électricité

Pour cerner les besoins de modélisation des problèmes d'applications, nous introduisons dans cette section le contexte national de la production d'électricité, et les processus décisionnels associés.

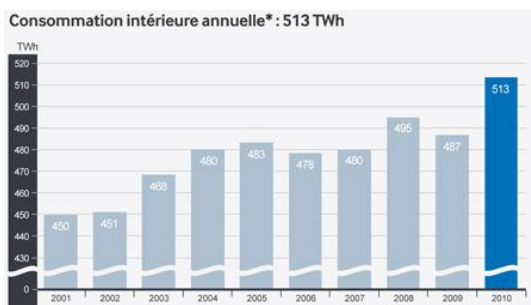


FIGURE 2.1 – Evolution annuelle de la consommation d'électricité en France

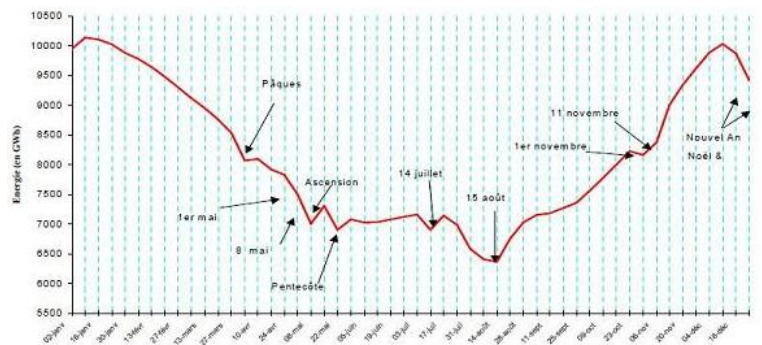


FIGURE 2.2 – Évolution de la demande au cours d'une année

2.1.1 Demandes en électricité

Comme l'électricité ne se stocke pas aux échelles de la consommation nationale, la production doit suivre la consommation électrique sur le réseau. De bonnes estimations de la demande sont ainsi indispensables pour réaliser un planning prévisionnel, ainsi que des mécanismes de réserve en cas d'écart entre le réalisé

et le prévisionnel. Les réserves se caractérisent par leur nature physique et leur délai de mise en œuvre. Le réglage primaire est un dispositif de commande automatique, pour une mise en action presque instantanée, répartie sur les unités de production. Le réglage primaire entraîne un écart de la fréquence du réseau avec la référence à $50Hz$. Pour la sécurité du système, l'écart de fréquence est limité à $1Hz$. La réserve secondaire ("télé-régulation") permet de rétablir la fréquence à $50Hz$, pour reconstituer la réserve primaire selon un processus automatisé. Le temps de réaction est d'au maximum 13 minutes.

En pratique, les demandes sont très variables sur différentes échelles de temps. La figure 2.1 illustre l'évolution de la consommation annuelle d'électricité de 2001 à 2010. Si une tendance globale à la hausse de consommation se dessine, la forme de cette courbe est assez imprédictible. L'évolution de la courbe de charge sur l'année 2010 à la figure 2.2 est plus prédictible. La saison et le climat sont directement reliés à la consommation d'électricité. De plus, les jours non travaillés causent des points anguleux, avec la réduction d'activité d'entreprises grandes consommatrices d'électricité, cela impacte la consommation à l'échelle globale du pays. Les graphes de la figure 2.7 précisent cela à l'échelle de la journée.

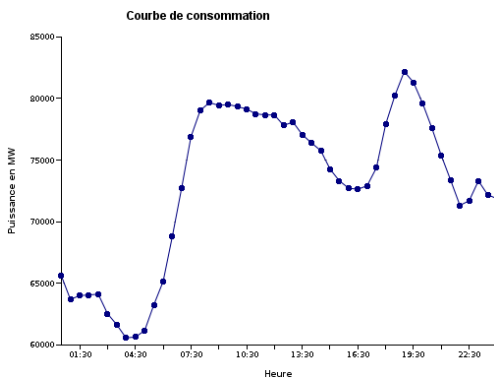


FIGURE 2.3 – Jour ouvré en hiver

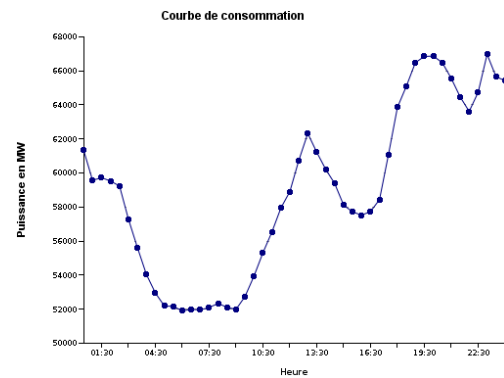


FIGURE 2.4 – Jour férié en hiver

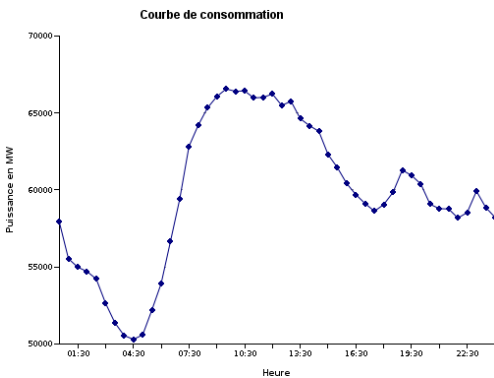


FIGURE 2.5 – Jour ouvré de printemps

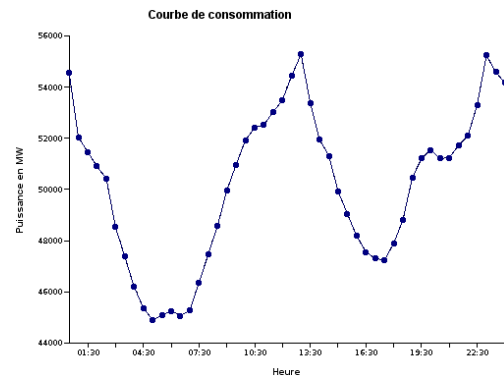


FIGURE 2.6 – Jour férié de printemps

FIGURE 2.7 – Données de RTE de la consommation d'électricité en France

On observe une variabilité importante à l'échelle journalière. La demande est basse de 22h30 à 5h30, un premier pic apparaît de 9h à midi, la demande décroît jusqu'à 18h où un second pic apparaît, avant une décroissance régulière jusqu'à la fin de la journée. Cette forme de courbe est prévisible, suivant l'activité humaine. Une différence apparaît entre jours ouvrés et fériés : le premier pic apparaît plus tôt sur un jour ouvré. La saisonnalité influence la valeur et l'amplitude des pics de la courbe de charge, encore plus marqués en hiver. Cela pose des difficultés comme de nombreux moyens de production ne peuvent pas facilement moduler la production.

2.1.2 Parc de production national

Cette section présente les différentes unités de production avec leurs caractéristiques de fonctionnement.

Parc hydraulique Le parc hydraulique est constitué d'unités dans les régions montagneuses, qui transforment l'énergie mécanique produite en énergie électrique, à partir des apports hydrauliques. Ces apports proviennent de la pluie ou stockés suite à la fonte des neiges, ce qui introduit de la stochasticité dans la gestion de production à moyen terme. Le parc hydraulique français comporte plus de 100 usines organisées dans 20 vallées nationales, et contribue à 10% de la production d'EDF. L'eau turbinée par une usine arrive dans son réservoir aval avec un certain délai de parcours. Certaines usines permettent d'effectuer du pompage d'eau du réservoir aval vers le réservoir amont, en consommant de l'électricité sur le réseau. La production hydraulique est un moyen de production T1 et réactif, mais disponible en quantité limitée selon les niveaux d'eau dans les réservoirs. Il est primordial de l'utiliser de façon optimisée pour réajuster la production à la demande quand les coûts marginaux du système sont les plus élevés.

Parc thermique classique Le parc thermique classique est constitué de 50 unités thermiques représentant environ 4% de la production nationale. Les combustibles sont le charbon, la houille, le fioul et le gaz. Les unités thermiques fonctionnent indépendamment les unes des autres. Leur structure de coût comprend des coûts de démarrage, des coûts fixes et des coûts proportionnels à la production. Le coût de démarrage dépend de la durée d'arrêt qui précède, un démarrage à chaud étant moins coûteux. La production est comprise entre une puissance minimale P_{min} et une puissance maximale P_{max} où aucune participation à la réserve n'est possible. La plage de production où la participation à la réserve est possible se situe entre $PCoMin > P_{Min}$ et $PCoMax < P_{Max}$. Les contraintes dynamiques de production comprennent des durées minimales de fonctionnement, d'arrêt, des durées minimales sur les paliers de production, une courbe de démarrage.

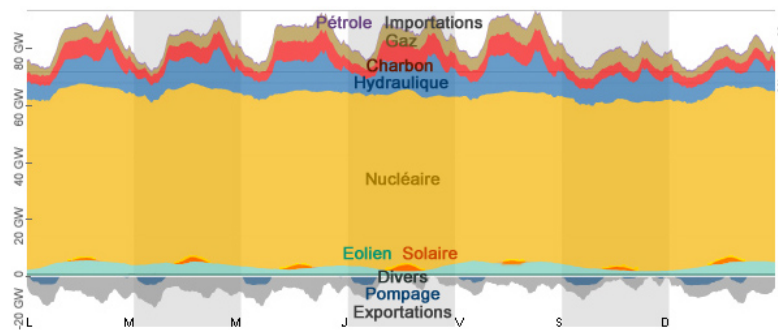


FIGURE 2.8 – Répartition des différents types de production à l'échelle de la semaine

Parc nucléaire Le parc français compte ainsi 58 réacteurs nucléaires (63 100 MW), répartis sur 19 centrales nucléaires en exploitation, dont 34 réacteurs d'une puissance de 900 MW, 20 réacteurs de 1 300 MW et 4 réacteurs de 1 450 MW. La production d'origine nucléaire est prédominante dans la production française d'EDF, environ 80% de la production totale. C'est le mode de production le plus rentable économiquement (sans compter les coûts d'investissement) permettant de produire à grande échelle.

Les réacteurs nucléaires sont soumis aux mêmes contraintes que les autres unités thermiques. Des contraintes additionnelles sont à considérer pour les réacteurs nucléaires. La production nucléaire se fait majoritairement à puissance maximale, les possibilités de modulations sont plus limitées. Ainsi, un nombre de baisses par jour est considéré, ainsi qu'une durée maximale de modulation. La spécificité majeure du nucléaire en terme de production vient du taux de disponibilité du parc nucléaire, de l'ordre de 80%. Lors

d'opérations de maintenance ou de rechargement, la production est complètement arrêtée sur des périodes conséquentes, ce qui est dimensionnant sur la gestion de production globale d'EDF.

Il existe trois types différents d'arrêts de tranche. Les arrêts pour simple rechargement (ASR), tous les 12 à 18 mois durent 30 jours environ, servent à renouveler un tiers du combustible du réacteur. Les visites partielles (VP), durant 60 jours environ, sont effectuées en alternance avec les ASR. Il s'agit non seulement de renouveler du combustible mais aussi d'effectuer des opérations de maintenance plus avancées. Les visites décennales (VD) durent une centaine de jours, réglementaires tous les dix ans pour un contrôle technique plus approfondi. Elles permettent à l'Autorité de Sûreté Nucléaire (ASN) de valider une prolongation de dix ans supplémentaires l'exploitation des réacteurs ou d'arrêter définitivement un réacteur.

Énergies renouvelables Du fait de la prépondérance du nucléaire, la France a pris du retard en matière d'énergies renouvelables. Le bilan RTE 2012 indique une production en France de 757 MW éoliens et 1012 MW photovoltaïques. Un parc renouvelable introduit des aléas sur la production, via l'ensoleillement ou la force des vents, qui se combinent aux aléas sur la demande, ce qui nécessite des réserves plus importantes.

Actifs financiers EDF utilise la tarification et des contrats spécifiques comme levier pour répartir la consommation dans le temps. Ceci se traduit des contrats présentant deux types de tarification : une tarification plus faible que la moyenne sur des périodes avantageuses pour EDF et une tarification bien plus élevée le reste du temps, où les clients concernés seront amenés à diminuer leur consommation électrique. Pour des gros clients comme des entreprises, des contrats tarifaires EJP (Effacements Jours de Pointe) permettent une tarification faible la grande majorité de l'année, et un préavis de quelques heures pour les journées à tarification élevée, choisis unes à unes par EDF entre novembre et mars inclus. Cette quantité d'énergie peut être vue comme un stock d'énergie disponible, que l'on peut estimer à 5000 MW. D'autres actifs financiers viennent des marchés d'électricité, comme le marché spot.

2.1.3 Processus décisionnels et problèmes d'optimisation

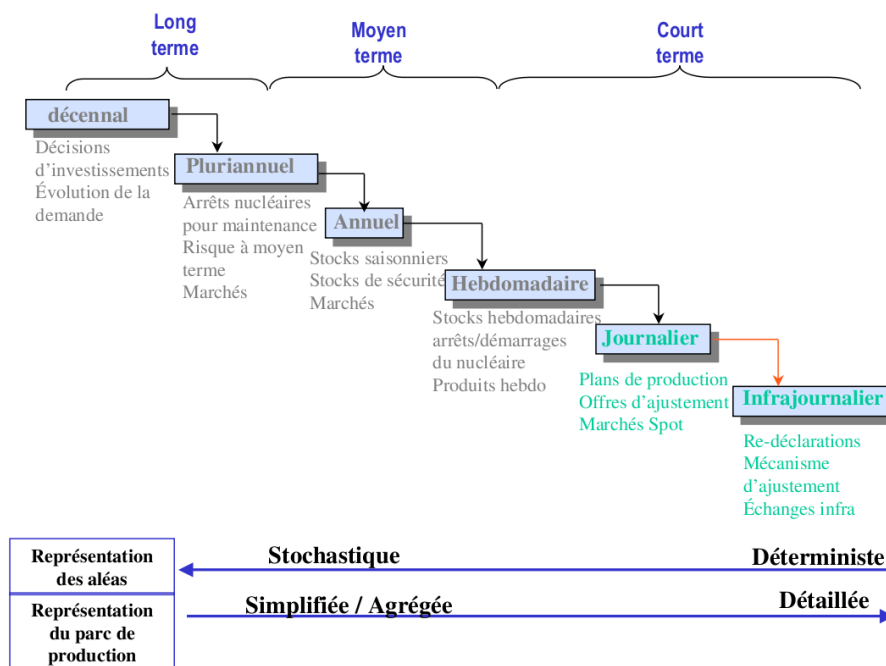


FIGURE 2.9 – Echelle des problèmes d'optimisation à EDF

Pour utiliser au mieux les grands stocks et assurer un coût de production global minimal, la gestion de

production d'EDF segmente les décisions sur divers horizons de temps. À *long terme* (10 à 20 ans), cela concerne les décisions d'investissements d'infrastructures (renouvellement du parc, nouvelles technologies ...). Sur un horizon *pluriannuel* de 3 à 5 ans, on planifie les arrêts des tranches nucléaires pour les maintenances et les rechargements du combustible. Sur l'horizon *annuel*, les valeurs d'usage des stocks hydraulique et nucléaire sont calculées, pour définir la stratégie de passage de l'hiver prochain, et négocier si nécessaire des contrats d'approvisionnement complémentaires. À l'horizon *hebdomadaire* (7 à 12 jours), des simulations sont effectuées, en vue de déclencher des options tarifaires comme les EJP. L'horizon *journalier* (1 à 2 jours) détermine le programme de production court terme pour RTE, tandis que l'horizon *infra-journalier* sert à adapter la production presque en temps continu suivant les évolutions réelles.

Plus l'horizon de temps se rapproche du temps réel, plus les pas de temps des problèmes d'optimisation se sont fins, induisant des modélisations de contraintes de plus en plus fines. À plus long terme, on peut s'autoriser des agrégations pour calculer des indicateurs globaux. Sur les problèmes journaliers, les pas de temps sont de 30 minutes, alors qu'ils sont de l'ordre de la journée pour les problèmes des horizons de temps plus lointains. Les contraintes dynamiques de production sur le parc thermique doivent figurer sur des pas de temps de 30 minutes, là où elles ne seraient pas visibles sur un pas de temps d'une journée. Inversement, plus l'horizon de temps est lointain, plus les aléas à considérer sont importants. Le problème court terme est déterministe, grâce à des estimations de demandes fiables, et par le placement de réserves.

2.2 Unit Commitment Problem

Le Unit Commitment Problem (UCP), problème académique de référence, se définit comme le problème décisionnel de sélection d'unités de production, et de placement de la production des centrales électriques pour satisfaire une demande prévisionnelle. UCP présente naturellement une structure à deux niveaux, exploitée dans [28] et [117]. Les décisions de premier niveau sont les décisions d'arrêt et de fonctionnement, reliées aux contraintes d'inertie de démarrage et d'arrêt des centrales. Le second niveau plus réactif, place la production une fois que les unités actives ont été sélectionnées.

2.2.1 Modélisation en PLNE de l'Unit Commitment Problem

On fournit ici la modélisation PLNE de l'UCP académique. On considère des coûts de démarrages dans la formulation éventuellement nuls si ce n'est pas ainsi modélisé. On introduit tout d'abord les variables binaires $x_{u,t}$ (dites de "set up") pour décrire le fonctionnement d'une centrale u au pas de temps t . Ces variables sont nécessaires pour écrire les coûts fixes dans la fonction objectif, mais aussi dans les contraintes de bornes de puissance et imposer une puissance produite nulle si $x_{u,t} = 0$. Pour les coûts de démarrage, on introduit également des variables binaires de démarrages $y_{u,t}$, dites de "start up", pour indiquer si la centrale u est démarrée à l'instant t . Si ces variables dépendent uniquement des variables $x_{u,t}$, avec la formule $y_{u,t} = \max(x_{u,t} - x_{u,t-1}, 0)$, le procédé de linéarisation revient en fait exactement à introduire ces variables. Pour les puissances et les réserves de puissance, on introduit pour toute centrale à chaque pas de temps des variables continues $P_{u,t}$ et $R_{u,t}$. Avec une telle modélisation, nous avons la formulation PLNE ci dessous.

Le critère à minimiser (2.1) comprend les coûts fixes de fonctionnement C_u^{fix} , les coûts proportionnels à la puissance produite C_u^{ppp} , et les coûts de démarrages C_u^{dem} (les coûts d'arrêt sont nuls). Les contraintes (2.2) couplent les variables binaires, définissant les variables de démarrages sur une variation des variables x . Les contraintes (2.3) et (2.4), contraintes de durées minimales de fonctionnement et d'arrêt seront expliquées à la section 2.2.3. Les contraintes (2.5) codent les conditions initiales en puissance. Les conditions initiales de durées fonctionnement ou d'arrêt à l'instant 0 sont reportées par extension de (2.3) et (2.4). Les

contraintes (2.6) et (2.7) expriment les contraintes de demandes en puissances et en réserves à tout instant. Les contraintes (2.8) et (2.9) expriment les bornes de puissance produites et le couplage entre variables binaires et variables continues, la puissance produite est nulle pour une unité arrêtée.

$$\min_{x,y,P \geq 0} \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}} C_u^{prp} P_{u,t} + C_u^{fix} x_{u,t} + C_u^{dem} y_{u,t} \quad (2.1)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad x_{u,t} - x_{t-1}^u \leq y_{u,t} \quad (2.2)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-L^u+1}^t y_{u,t'} \leq x_{u,t} \quad (2.3)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-l^u+1}^t y_{u,t'} \leq 1 - x_{t-l^u}^u \quad (2.4)$$

$$\forall u \in \mathcal{U}, \quad P_0^u = P_u^{init} \quad (2.5)$$

$$\forall t \in \mathcal{T}, \quad \sum_{u \in \mathcal{U}} P_{u,t} = D_t^P \quad (2.6)$$

$$\forall t \in \mathcal{T}, \quad \sum_{u \in \mathcal{U}} R_{u,t} = D_t^R \quad (2.7)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad x_{u,t} \cdot P_u^{min} + R_{u,t} \leq P_{u,t} \quad (2.8)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad P_{u,t} \leq x_{u,t} \cdot P_u^{max} - R_{u,t} \quad (2.9)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad 0 \leq R_{u,t} \leq R_u^{max} \quad (2.10)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad x_{u,t}, y_{u,t} \in \{0, 1\} \quad (2.11)$$

La portée de ce modèle est limitée en pratique. Les unités ont ici un fonctionnement indépendant les unes des autres, ce qui n'est pas valide pour des unités hydrauliques, où les couplages entre unités d'une vallée sont très dimensionnants. Cette modélisation est plus utilisée comme aide à la décision pour des modèles macroscopiques, à assez long terme, où l'hydraulique doit être agrégé ou défalqué de la demande.

2.2.2 Résolution de problèmes de type UCP, état de l'art

La famille des problèmes UCP est vaste, les modélisations différant suivant le contexte opérationnel. On résume ici quelques approches utilisées avec succès sur des variantes d'UCP.

Approches basées sur des méthodes exactes L'article [11] fournit une modélisation académique de l'UCP en programmation mixte en nombres entiers, où les courbes de démarrages sont calculées finement, ce qui est dimensionnant pour le fonctionnement réel, là où la modélisation précédente néglige l'inertie des démarrages, et les différentes courbes de démarrages possibles. L'approche est frontale en PLNE, tirant partie d'une diminution judicieuse du nombre de variables binaires. En effet, le fait que les variables $x_{u,t}$ soient entières et que la minimisation en $y_{u,t}$ soit avec des coefficients strictement positifs implique que le problème avec la relaxation continue des $y_{u,t}$ a la même valeur que le problème avec des variables de démarrage entières.

Sur le problème de production journalière défini avec les contraintes mentionnées à la section 2.1.3, cela nécessite d'introduire des variables binaires pour modéliser le fonctionnement des centrales thermiques et les unités hydrauliques. L'approche industrialisée est présentée dans [107], [49] et [87]. La méthode est basée sur une méthode exacte, par relaxation lagrangienne des contraintes de demandes. Des sous blocs indépendants sont formés par les unités nucléaires, ou par une vallée hydraulique comprenant de nombreuses usines. C'est en fait la relaxation linéaire qui est calculée par une méthode de faisceaux (pour permettre une résolution en 15 minutes, temps imparti dans le processus opérationnel). Les variables duales sont

calculées à cette étape et sont interprétées économiquement comme des coûts marginaux qui permettent de calculer des solutions entières dans un second temps, sur des problèmes découplés.

Approches heuristiques Pour résoudre des problèmes d'UCP sur de grandes tailles de données, de nombreuses approches ont été développées. De nombreuses publications ont utilisé des approches évolutionnistes et plus particulièrement génétiques. Nous mentionnons [12] pour une approche parallélisable, [42] pour une hybridation avec un algorithme génétique guidée par une heuristique lagrangienne, et [92] pour une hybridation avec un algorithme de recherche tabou. Récemment, le formalisme VNS a donné d'excellents résultats sur un problème de type Unit Commitment dans [121].

2.2.3 Contraintes dynamiques

Les articles [105] et [85] développent une modélisation efficace du polyèdre formé par les variables de set up x_t^u avec les contraintes de durée minimale d'arrêt l^u , et de durée minimale de fonctionnement L^u . La formulation de [85] se base uniquement sur les variables de set up x_t^u .

Théorème 2.2.1 *Les contraintes de durées minimales d'arrêt et de fonctionnement peuvent s'écrire respectivement de la manière suivante avec les variables x_t^u :*

$$\forall u \in \mathcal{U}, \forall 1 \leq t < \tau \leq \min(T, t + L^u), \quad x_t^u - x_{t-1}^u \leq x_\tau^u \quad (2.12)$$

$$\forall u \in \mathcal{U}, \forall 1 \leq t < \tau \leq \min(T, t + l^u), \quad x_{t-1}^u - x_t^u \leq 1 - x_\tau^u \quad (2.13)$$

Pour toute tranche fixée, ces contraintes ne suffisent pas à décrire le polyèdre entier correspondant $P_T(l, L)$. Pour améliorer la description et le relâché continu, leur article expose des coupes valides, qui de plus permettent de décrire le polyèdre entier, pour une application dans un algorithme de type Branch&Cut :

Théorème 2.2.2 *Soit $k \in \mathbb{N}^*$ un entier non nul. On considère un ensemble d'entiers disjoints de \mathcal{T} : $\varphi(1) < \psi(1) < \varphi(2) < \psi(2) < \dots < \varphi(k) < \psi(k) < \varphi(k+1)$ tels que $\varphi(k+1) - \varphi(1) \leq L$ (resp $\leq l$). On a les coupes suivantes valides pour le polyèdre $P_T(l, L)$:*

$$-\sum_{j=1}^{k+1} x_{\varphi(j)} + \sum_{j=1}^k x_{\psi(j)} \leq 0 \quad \left(\text{resp } \sum_{j=1}^{k+1} x_{\varphi(j)} - \sum_{j=1}^k x_{\psi(j)} \leq 1 \right) \quad (2.14)$$

Ces coupes génèrent exactement une description linéaire de $P_T(l, L)$.

Dans l'article [105], on rajoute les variables y_t^u de démarrage, utiles à la modélisation des coûts de démarrage. L'article [105] prouve tout d'abord que les inégalités de couplage écrites entre $x_{u,t}$ et $y_{u,t}$ forment une facette du polyèdre correspondant. De plus, il est remarqué qu'il existe une relation linéaire entre $z_{u,t}$, variables binaires indiquant si l'unité u est arrêtée à l'instant t , avec $x_{u,t}$ et $y_{u,t}$: $z_{u,t} = y_{u,t} + x_{u,t-1} - x_{u,t}$. Les contraintes de démarrage et d'arrêt qui couplent les x_t^u et y_t^u sont écrites comme suit :

$$\forall u \in \mathcal{U}, \forall t \in \llbracket L^u + 1, T \rrbracket, \quad \sum_{t'=t-L^u+1}^t y_{u,t'} \leq x_{u,t} \quad (2.15)$$

$$\forall u \in \mathcal{U}, \forall t \in \llbracket l^u + 1, T \rrbracket, \quad \sum_{t'=t-l^u+1}^t y_{u,t'} \leq 1 - x_{t-l^u}^u \quad (2.16)$$

On peut obtenir la formulation [105] par le processus de coupes de cliques (cf [113]). Les contraintes peuvent s'écrire avec des $y_{u,t'} \leq x_{u,t}$, correspondant à l'implication qu'un démarrage entraîne sur les variables de fonctionnement suivantes. De plus, deux démarrages sont espacés d'au moins $l^u + L^u$ pas de temps,

par les contraintes de durées minimales d'arrêt. En considérant la contrainte implicite $\sum_{t=t_0}^{t_0+l^u+L^u} y_t^u \leq 1$, comprise dans les équations précédentes, on obtient alors la formulation de [105] par regroupement sur des cliques. On note $C_T(l, L)$ l'enveloppe convexe des points réalisables entiers à tranche u fixée. La force de cette formulation réside dans la description du polyèdre :

Théorème 2.2.3 *Les familles d'inégalités (2.15) et (2.16) dominent les coupes (2.14). Enfin, les familles d'inégalités (2.15) et (2.16) associée aux inégalités triviales $y_t \geq 0$ et $x_{u,t} - x_{u,t-1} \geq y_{u,t}$ fournissent une description de $C_T(l, L)$. De plus, la projection sur l'espace des x_t de $C_T(l, L)$ est exactement $P_T(l, L)$.*

Il est également prouvé dans l'article [105] que cette formulation contient les coupes de l'article [85], par projection sur l'espace des x_t^u . Avec la positivité des variables et la contrainte (0.1), on a une description exacte de l'enveloppe convexe du polytope min up-min down avec les variables de démarrages.

2.3 UCPd un UCP discret pour un parc thermique

Un premier problème d'application de la thèse poursuit mes travaux de master de [51], sur une problématique de production journalière, avec les contraintes discrètes du parc thermique à flamme d'EDF. Nous rappelons ici la définition du problème, les formulations PLNE obtenues et les jeux de données.

2.3.1 Description du problème UCPd

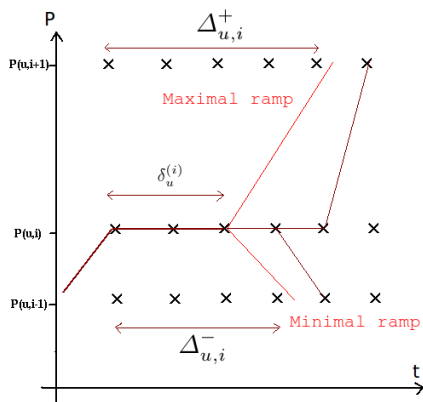


FIGURE 2.10 – Durées de paliers

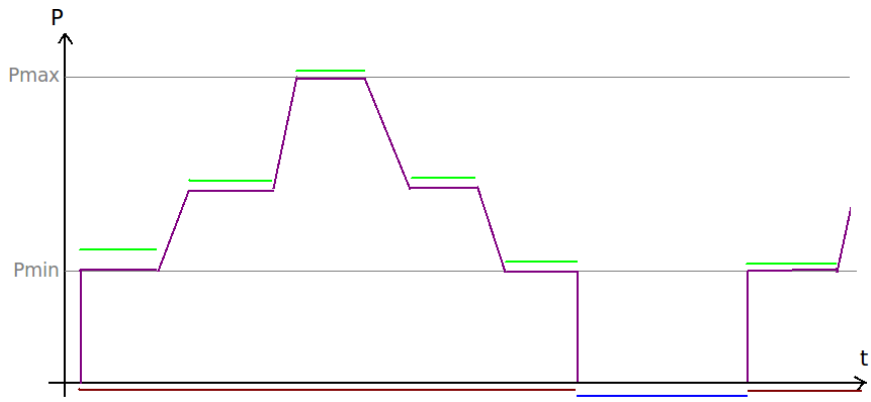


FIGURE 2.11 – Contraintes de paliers d'UCPd

Pour modéliser le fonctionnement des centrales thermiques, nous incorporons des contraintes de paliers sur des *points de fonctionnement*, où la puissance doit être stationnaire sur une *durée minimale de palier* comme illustré sur la figure 2.3.1. L'UCP académique ne comprend pas de telles contraintes. Cela donne naissance au problème UCPd, un UCP discret où la production des centrales est modélisée uniquement sur des points de fonctionnement, pour un fonctionnement discret. Les contraintes techniques d'une centrale thermique sont discrétisées en conséquence. En particulier, les transitions possibles entre deux pas de temps se font uniquement vers un point voisin. Les contraintes de gradients limités et de paliers sur des points de fonctionnement s'agrègent comme des durées de paliers minimales, qui diffèrent suivant que le palier soit suivi par une hausse ou une baisse de puissance, comme illustré sur la figure 2.10.

2.3.2 Formulations PLNE

Définition des variables Les puissances de production étant discrètes, UCPd comprend uniquement des variables binaires. Deux variantes de variables ont été envisagées, comparées sur les figure 2.12 et 2.3.2.

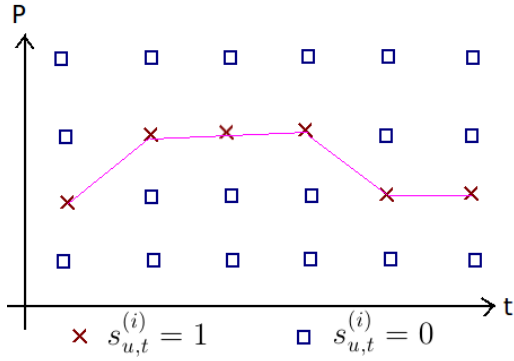


FIGURE 2.12 – Variables d'états

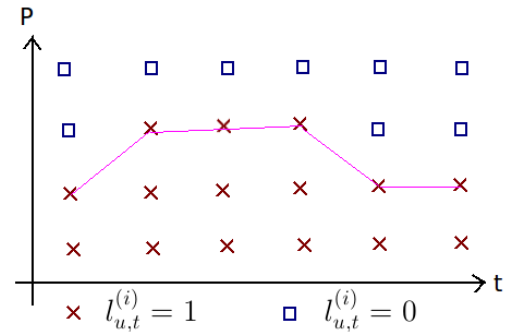


FIGURE 2.13 – Variables de paliers

Les *variables d'état* sont notées $s_{u,t}^{(i)} \in \{0, 1\}$ avec $s_{u,t}^{(i)} = 1$ si la centrale u produit exactement au point i à l'instant t . L'autre formulation utilise des *variables de niveaux*, $x_{u,t}^{(i)} \in \{0, 1\}$ où $x_{u,t}^{(i)} = 1$ si la centrale u produit au moins au point i à l'instant t . Dans les deux cas, ces variables désagrégées contiennent les variables de set up $x_{u,t}$ de l'état de l'art par $x_{u,t} = x_{u,t}^{(1)} = \sum_i s_{u,t}^{(i)}$. On utilisera les extensions $x_{u,t}^{(0)} = 1$, $x_{u,t}^{(N(u)+1)} = 0$, $s_{u,t}^{(N(u)+1)} = 0$, $s_{u,t}^{(0)} = 1 - \sum_i s_{u,t}^{(i)}$. Dans les deux cas, on introduit des variables binaires de démarrages pour chaque point de fonctionnement, $y_{u,t}^{(i)+}$ et $y_{u,t}^{(i)-}$, qui correspondent au démarrage du palier i après une montée (et respectivement une baisse) de puissance. On utilisera par commodité les extensions $y_{u,t}^{(0)+} = y_{u,t}^{(N(u)-)} = 0$.

Formulation par états Les variables d'état permettent de modéliser naturellement UCPd en PLNE :

$$\min_{s,y} \sum_{u,t} +C_u^{strt} y_{u,t}^{(1)+} + \sum_i \left(C_u^{Prop} P_{u,i} + C_u^{fix} \right) s_{u,t}^{(i)} \quad (2.17)$$

$$\forall u, i \quad s_{u,0}^{(i)} = \mathbb{1}_{i_0(u)=i} \quad (2.18)$$

$$\forall u, t, , \quad \sum_i s_{u,t}^{(i)} \leq 1 \quad (2.19)$$

$$\forall u, t, i, \quad s_{u,t}^{(i)} + s_{u,t-1}^{(i-1)} - 1 \leq y_{u,t}^{(i)+} \quad (2.20)$$

$$\forall u, t, i, \quad s_{u,t}^{(i)} + s_{u,t-1}^{(i+1)} - 1 \leq y_{u,t}^{(i)-} \quad (2.21)$$

$$\forall u, t, \quad -1 \leq \sum_i i (s_{u,t}^{(i)} - s_{u,t-1}^{(i)}) \leq 1 \quad (2.22)$$

$$\forall u, t, i, \quad \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} \leq s_{u,t}^{(i)} \quad (2.23)$$

$$\forall u, t, i, \quad \sum_{t'=t+1}^{t+\Delta_{u,i}^-} y_{u,t'}^{(i-1)-} \leq s_{u,t}^{(i)} \quad (2.24)$$

$$\forall u, t, \quad \sum_{t'=t-L_u+1}^t y_{u,t'}^{(1)+} \leq \sum_i s_{u,t}^{(i)} \quad (2.25)$$

$$\forall u, t, \quad \sum_{t'=t-l_u+1}^t y_{u,t'}^{(1)+} \leq 1 - \sum_i s_{u,t-l_u}^{(i)} \quad (2.26)$$

$$\forall t, \quad \sum_{u,i} P_{u,i} s_{u,t}^{(i)} \geq D_t^P \quad (2.27)$$

$$\forall t, \quad \sum_{u,i} R_{u,i}^1 s_{u,t}^{(i)} \geq D_t^{R1} \quad (2.28)$$

$$\forall t, \quad \sum_{u,i} R_{u,i}^2 s_{u,t}^{(i)} \geq D_t^{R2} \quad (2.29)$$

$$\forall u, t, i, \quad y_{u,t}^{(i)+}, y_{u,t}^{(i)-}, s_{u,t}^{(i)} \in \{0, 1\} \quad (2.30)$$

Le critère à minimiser (2.17) est analogue à celui de l'UCP. Les contraintes (2.18) expriment les conditions initiales en puissance. Les conditions initiales de durées fonctionnement ou d'arrêt à l'instant 0 sont également nécessaires, elles se trouvent reportées par extension de (2.25) et (2.26). Les contraintes (2.19) sont nécessaires à la définition des variables s , contraintes GUB pour imposer le fonctionnement sur un

unique point de fonctionnement. Les contraintes (2.22) définissent les sauts de transitions entre deux pas de temps, un changement de point de fonctionnement ne peut se faire uniquement vers un point voisin. Les contraintes (2.20) et (2.21) couplent les variables binaires, définissant les variables de démarrages sur une variation des variables x . Les contraintes (2.25) et (2.26), sont les contraintes "min-up, min-down" de la section 2.2.3 écrites avec les variables s . Les contraintes (2.23) et (2.24), sont les contraintes de durées minimales de paliers sur les points de fonctionnement, adaptées de [105]. Les contraintes (2.27), (2.28) et (2.29) expriment les contraintes de satisfaction des demandes en puissances et des réserves primaires et secondaires à tout instant, il s'agit de contraintes d'inégalités même pour les demandes dans ce modèle.

Formulation par niveaux La formulation par niveaux s'écrit de manière analogue en associant à chaque variable de niveau l'écart induit par rapport au point inférieur :

$$\min_{x,y} \sum_{u,t} \left(C_u^{fix} x_{u,t}^{(1)} + C_u^{strt} y_{u,t}^{(1)+} + \sum_i C_u^{Prop} (P_{u,i} - P_{u,i-1}) x_{u,t}^{(i)} \right) \quad (2.31)$$

$$\forall u, i, \quad x_{u,0}^{(i)} = \mathbb{1}_{i_0(u) \geq i} \quad (2.32)$$

$$\forall u, t, i, \quad x_{u,t}^{(i)} \leq x_{u,t}^{(i-1)} \quad (2.33)$$

$$\forall u, t, i, \quad x_{u,t}^{(i)} \leq x_{u,t+1}^{(i-1)} \quad (2.34)$$

$$\forall u, t, i, \quad x_{u,t}^{(i)} \geq x_{u,t+1}^{(i+1)} \quad (2.35)$$

$$\forall u, t, i, \quad x_{u,t}^{(i)} - x_{u,t-1}^{(i)} \leq y_{u,t}^{(i)+} \quad (2.36)$$

$$\forall u, t, i, \quad x_{u,t-1}^{(i+1)} - x_{u,t}^{(i+1)} \leq y_{u,t}^{(i)-} \quad (2.37)$$

$$\forall u, t, \quad \sum_{t'=t-L_u+1}^t y_{u,t'}^{(1)+} \leq x_{u,t}^{(1)} \quad (2.38)$$

$$\forall u, t, \quad \sum_{t'=t-l_u+1}^t y_{u,t'}^{(1)+} \leq 1 - x_{u,t-l_u}^{(1)} \quad (2.39)$$

$$\forall u, t, i, \quad \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} \leq x_{u,t}^{(i)} \quad (2.40)$$

$$\forall u, t, i, \quad \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} \leq 1 - x_{u,t}^{(i+1)} \quad (2.41)$$

$$\forall u, t, i, \quad \sum_{t'=t+1}^{t+\Delta_{u,i}^-} y_{u,t'}^{(i-1)-} \leq x_{u,t}^{(i)} \quad (2.42)$$

$$\forall u, t, i, \quad \sum_{t'=t+1}^{t+\Delta_{u,i}^-} y_{u,t'}^{(i-1)-} \leq 1 - x_{u,t}^{(i+1)} \quad (2.43)$$

$$\forall t, \quad \sum_{u,i} (P_{u,i} - P_{u,i-1}) x_{u,t}^{(i)} \geq D_t^P \quad (2.44)$$

$$\forall t, \quad \sum_{u,i} (R_{u,i}^1 - R_{u,i-1}^1) x_{u,t}^{(i)} \geq D_t^{R1} \quad (2.45)$$

$$\forall t, \quad \sum_{u,i} (R_{u,i}^2 - R_{u,i-1}^2) x_{u,t}^{(i)} \geq D_t^{R2} \quad (2.46)$$

$$\forall u, t, i, \quad y_{u,t}^{(i)+}, y_{u,t}^{(i)-}, x_{u,t}^{(i)} \in \{0, 1\} \quad (2.47)$$

Les contraintes (2.32) expriment les conditions initiales en puissance. Les contraintes (2.36) et (2.37) couplent les variables binaires, définissant les variables de démarrage sur une variation des variables x . Les contraintes (2.33) sont liées à la définition des variables x , pour qu'un point soit actif, cela impose que les points inférieurs soient aussi activés. Les contraintes (2.34) et (2.35) définissent les sauts de transitions entre deux pas de temps illustrées figure 3.1. Les contraintes (2.38) et (2.39) sont les contraintes "min-up, min-down" de la section 2.2.3 écrites avec les variables de niveaux x . Les contraintes (2.40),(2.41),(2.42) et (2.43) sont les contraintes de durées minimales de paliers, adaptées sur les points de fonctionnement. Les contraintes (2.44), (2.45) et (2.46) expriment les contraintes de demandes en puissances et en réserves primaires et secondaires.

2.3.3 Jeux de données

Les jeux de données ont été construits pour être représentatifs du monde réel, avec un horizon de 2 jours discrétisés en 96 pas de temps, et un parc de production allant jusqu'à 80 centrales. L'influence de la saisonnalité et des jours fériés observée sur la figure 2.7, incite à effectuer des tests sur quatre configurations : D80a sur un jour travaillé de printemps D80b sur un jour non travaillé de printemps, D82a sur un jour travaillé d'hiver, D82b sur un jour non travaillé d'hiver. En fait, ces jeux de données s'avèrent plus difficiles que les données réelles, du fait d'imposition/pannes . . . Des données plus représentatives du parc thermique à flamme ont été obtenues avec un préfixe `dat`. Des données tronquées sur un sous ensemble de tranches issues du jeu de données D80a, le suffixe indiquant alors le nombre de tranches. L'instances D67 correspond au jeu partiel en considérant le parc Fuel/Gaz. Le tableau 2.1 fournit le nombre de centrales totales et initialement actives, et les nombres de variables binaires avant preprocessing pour indiquer la combinatoire des instances.

Data	Units	Init	var bin
Dat6	6	4	4 050
Dat16	16	5	10 800
Dat35	35	10	23 630
Dat57	57	16	38480
Dat67	67	12	45 230
D80a	80	21	54 000
D80b	80	12	54 000
D82a	82	31	56 000
D82b	82	29	56 000

TABLE 2.1 – Caractéristiques des données d'UCPd

2.3.4 Etat de l'art de la résolution PLNE

Il n'y avait pas de résolution spécifique à UCPd à EDF avant [51], le problème étant traité globalement, avec les unités hydrauliques qui concentrent les difficultés de la résolution. L'approche en exploitation [49] est applicable à la restriction au parc thermique à flamme, avec les mêmes hypothèses que le problème UCPd, il s'agit dans ce cas d'une décomposition lagrangienne avec dualisation des contraintes de demandes, où les sous-problèmes découplés des productions pour chaque unité thermique sont résolus par programmation dynamique. Une motivation de l'étude de [51] était d'investiguer si une résolution frontale est envisageable sur le périmètre du parc thermique à flammes.

Le travail mené dans [51] a modélisé et formulé le problème en PLNE, en comparant empiriquement les diverses formulations envisagées. Les résultats de la convergence ont été très satisfaisants dans cette optique, des heuristiques de fixation de variable ont encore accéléré la résolution. La formulation par niveaux est plus efficace à ce stade, avec une meilleure utilisation des structures de coupes de cliques offertes par Cplex, ce que les contraintes (2.33), (2.34) et (2.35) permettent d'activer. Les autres coupes cruciales dans la résolution sont les coupes de couvertures, activées par les contraintes de demandes. En travaillant contraintes par contraintes, les contraintes de transitions (2.34) et (2.35) semblent empiriquement plus efficaces que les contraintes (2.22), en plus grands nombres mais fournissant de meilleurs relâchés continus. Cela compense la taille mémoire supérieure et les temps de calculs supérieurs pour les algorithmes tels que le simplexe. La modélisation des contraintes de durées minimales de paliers a donné lieu à plusieurs types de formulations. Si on pouvait se passer de toutes les variables de démarrages y , en étendant [85],

la qualité de la relaxation continue s'avérait meilleure avec toutes les variables y , pour un avantage décisif sur la convergence Branch&Bound, avec des résultats empiriques similaires à [105]. Dans le cas courant où les durées de paliers $\Delta_{u,i}^-$ et $\Delta_{u,i}^+$ sont identiques, une formulation particulière pouvait être envisagée, pour une meilleure efficacité empirique.

2.4 Planification des arrêts des réacteurs nucléaires

Notre second problème d'application est la planification des arrêts des centrales nucléaires pour maintenance et rechargement. Il s'agit de planifier les dates d'arrêts et les quantités de combustibles à recharger, sur un horizon de 3 à 5 ans. Sur un tel horizon et de tels pas de temps, la modélisation se doit de considérer de l'incertitude dans l'optimisation, en s'autorisant des simplifications des modélisations des contraintes de fonctionnement des centrales. Ce problème a fait l'objet du challenge EURO/ROADEF 2010 proposé par EDF R&D. Le principe du challenge est qu'EDF fournit une spécification du problème et des jeux de données publics, sur lesquels des équipes participantes découvrant le sujet confrontent leurs approches, élaborées en 9 mois environ. Cela a fourni un état de l'art de grande qualité, que nous présentons ici. Les temps de calcul impartis pour le challenge sur des instances de taille réelle étaient de 1h, avec des contraintes d'espace mémoire limités, ce ne sont pas des contraintes opérationnelles.

2.4.1 Définition du problème du challenge ROADEF

Pour le challenge ROADEF, le problème a été modélisé dans [103] comme un problème d'optimisation stochastique avec recours. Le challenge modélise deux types de centrales. Le type 1 (T1) correspond aux unités qui ne nécessitent pas d'arrêts pour leur maintenances, indicées par $j \in \mathcal{J}$, modélisant le parc thermique à flamme mais aussi les marchés. Le type 2 (T2) correspond aux réacteurs nucléaires, indicés par $i \in \mathcal{I}$, nécessitant de longues phases de maintenances où la production est arrêtée. Un ensemble discret de scénarios modélise l'incertitude sur les demandes, les capacités de production et les coûts de productions des unités T1. Le premier niveau est défini avec les décisions principales communes à tous les scénarios, les dates d'arrêts et les niveaux de rechargement des unités T2, avec des contraintes propres d'ordonnancement et de couplage entre les dates d'arrêts et les bornes de rechargements. Le second niveau correspond aux plans de productions réalisés sur tous les scénarios des unités T1 et T2, pour remonter des contraintes techniques et calculer le coût de productions associés à chaque scénario. Les notations sont réintroduites au fur et à mesure de cette section. Nous insistons sur la convention qu'un *cycle de production* (i, k) débute avec une période d'*arrêt* de longueur $\mathbf{D}\mathbf{a}_{i,k}$ en semaines, où les opérations de maintenance et de rechargement sont effectuées à production arrêtée, et se poursuit alors avec la *campagne de production* (i, k) , où le réacteur nucléaire produit selon les contraintes techniques de fonctionnement, et se termine avec le début de l'arrêt $(i, k + 1)$. Les conditions initiales sont incorporées dans un cycle fictif $k = 0$, démarrant à la première semaine avec le stock de combustible résiduel et une durée d'arrêt nulle si le cycle de production était en cours.

Fonction objectif La fonction objectif considérée pour le challenge ROADEF comprend les coûts de rechargement, $\mathbf{C}_{i,k}^{rd}$ proportionnels pour chaque cycle (i, k) à la quantité de combustible rechargée, l'espérance sur les scénarios des coûts de production des unités T1 à tout instant, $\mathbf{C}_{j,s,t}^{prd}$, proportionnels à la puissance de production, ainsi qu'une valorisation $\mathbf{C}_{i,s}^{val}$ du stock final de combustible nucléaire pour tout réacteur i (qui se calcule également en espérance sur les différents scénarios). Sans la valorisation du stock final, cela tendrait à prévoir un maximum d'arrêts juste après la fin de l'horizon de temps.

Contrainte	Description
CT1	Contraintes de demandes en puissance
CT2	Bornes de production des unités T1
CT3	Production nulles des T2 sur les arrêts
CT4-5	Bornes de production des unités T2
CT6	Contraintes de stretch
CT12	Contraintes de modulation
CT7	Bornes de rechargement
CT8	Niveaux de combustibles initiaux
CT9	Lien production/consommation de combustible
CT10	pertes de combustibles au rechargement
CT11	Bornes de combustibles
CT13	Fenêtre de temps des dates d'arrêts
CT14-18	Contraintes d'espacements entre arrêts
CT19	Contraintes d'ordonnancement des arrêts liés à une ressource limitée
CT20	Nombre maximal d'arrêts simultanés
CT21	Puissance maximale arrêtée.

TABLE 2.2 – Description des contraintes ROADEF

Contraintes de demande La contrainte CT1 du challenge ROADEF impose que la production totale des unités T1 et T2 satisfait exactement la demande $\mathbf{Dem}^{t,s}$ à tout instant pour tous les scénarios considérés. Dans les données, la production hydraulique estimée est retirée de la demande $\mathbf{Dem}^{t,s}$. Dans la modélisation, un groupe de défaillance est ajouté, comme un moyen de production T1 au coût de production très élevé et à la capacité de production infinie, garantissant la faisabilité de ces contraintes.

Contraintes de production des unités T1 Les unités T1 ont des bornes de production définies entre $\mathbf{Pmin}_{j,t}^s$ et $\mathbf{Pmax}_{j,t}^s$ par les contraintes CT2. Ces données sont stochastiques, ce qui permet d'introduire des scénarios de pannes sur des unités thermiques, $\mathbf{Pmax}_{j,t}^s = 0$ pour une indisponibilité totale.

Contraintes de stocks des unités T2 Les contraintes CT7 bornent les valeurs des rechargements des arrêts effectués entre $\mathbf{Rmin}_{i,k}$ et $\mathbf{Rmax}_{i,k}$. Les contraintes CT11 bornent les stocks résiduels à tout instant. Le stock nucléaire est une grandeur positive dans l'unité du challenge (différente du système d'unités historique à EDF), et doit toujours être inférieur à $\mathbf{Smax}_{i,k}$. La contrainte d'anticipation maximale donne une borne supérieure $\mathbf{Amax}_{i,k+1}$ au stock résiduel en fin de cycle i, k pour pouvoir entamer le cycle suivant $i, k + 1$. La quantité initiale de combustible est fixée et connue \mathbf{Xi}_i par les contraintes CT8. L'évolution temporelle du stock est déterminée par les puissances et rechargement passés : dans un cycle de production, les contraintes CT9 donnent le lien entre la puissance de production et la décroissance du stock de combustible résultant, tandis que les contraintes CT10 donnent le lien entre le stock après rechargement et le stock avant rechargement, en tenant compte des pertes au rechargement, modélisés par un facteur multiplicatif $\mathbf{Q}_{i,k}$.

Contraintes de production pour les réacteurs nucléaires Pour les réacteurs nucléaires, les contraintes de bornes de productions sont tout d'abord que la production est nulle pendant les arrêts. La puissance maximale de production d'une unité i est \mathbf{Pmax}_i^t , dépendant du temps, mais pas des scénarios stochastiques. Il n'y a pas de puissance minimale autre que 0 dans le modèle, il s'agit d'une simplification.

Instances	I	J	K	S	T	W	varBin	Troncat.	varBin
A1	10	11	6	10	1750	250	3892	A1_3_120	245
A2	18	21	6	20	1750	250	7889	A2_3_120	663
A3	18	21	6	20	1750	250	8162	A3_3_120	568
A4	30	31	6	30	1750	250	17465	A4_3_120	1305
A5	28	31	6	30	1750	250	15357	A5_3_120	1868
B6	50	25	6	50	5817	277	24563	B6_3_120	1519
B7	48	27	6	50	5565	265	35768	B7_3_120	4658
B8	56	19	6	121	5817	277	69653	B8_3_120	11057
B9	56	19	6	121	5817	277	69306	B9_3_120	11146
B10	56	19	6	121	5565	265	29948	B10_3_120	1816
X11	50	25	6	50	5817	277	20081	X11_3_120	1470
X12	48	27	6	50	5523	263	27111	X12_3_120	1927
X13	56	19	6	121	5817	277	30154	X13_3_120	2838
X14	56	19	6	121	5817	277	30691	X14_3_120	2844
X15	56	19	6	121	5523	263	27233	X15_3_120	1784

TABLE 2.3 – Caractéristiques des instances ROADEF et leur troncature

Quand le stock de combustible est en deçà du niveau de bore nul, $\mathbf{Bo}_{i,k}$, la production est imposée par les contraintes CT6 suivant un profil décroissant, "stretch". $m \in \mathcal{M}_{i,k} = \llbracket 1, \mathbf{Np}_{i,k} \rrbracket$ désigne l'ensemble des "modes" de production du réacteur i pour le cycle (i, k) du profil décroissant en fin de cycle. Le mode de production nominal, $m = 1$, correspond à un stock de combustible supérieur à $\mathbf{Bo}_{i,k}$. Une fois le niveau de bore nul $\mathbf{Bo}_{i,k}$ atteint, la production n'est plus modulable et suit un profit décroissant, modélisé par une fonction linéaire par morceaux, dont les points anguleux sont notés $(\mathbf{f}_{i,k,m}, \mathbf{c}_{i,k,m})$.

La capacité des unités T2 à "moduler", ie ne pas produire à la puissance maximale, est limitée par la contrainte CT12, avec un volume maximal de modulation par cycle $\mathbf{Mmax}_{i,k}$. Les contraintes réelles sont des durées maximales de modulation, et des durées minimales aux puissances maximales après une modulation.

Contraintes d'ordonnement des arrêts Tout d'abord, les contraintes de fenêtres de temps CT13 imposent que l'arrêt (i, k) commence au plus tôt à la semaine $\mathbf{To}_{i,k}$ et au plus tard à $\mathbf{Ta}_{i,k}$. Ces contraintes peuvent provenir de contraintes de sécurité, dates au plus tard de visites décennales par exemple, ou à des exigences contractuelles, les opérations contractualisées imposant de ne pas trop modifier l'organisation du sous traitant. Les contraintes CT14 définissent pour un ensemble donné d'unités T2 concernées $\mathbf{A14}$, un nombre de semaines $\mathbf{Se14}$. Si $\mathbf{Se14} \geq 0$, deux arrêts (i, k) et (i, k) doivent être espacés de $\mathbf{Se14}$ semaines entre la fin du premier arrêt et le début de l'arrêt suivant. Si $\mathbf{Se14} < 0$, les plages d'arrêts ne peuvent se recouvrir que sur $-\mathbf{Se14}$ semaines. les contraintes CT15 sont identiques à CT14, s'exerçant sur une fenêtre de temps fixée. Les contraintes CT16 du challenge ROADEF imposent une durée minimale $\mathbf{Se16} > 0$ entre les dates de début d'arrêts des centrales dans un ensemble $\mathbf{A16}$. Les contraintes CT17 imposent une durée minimale $\mathbf{Se17} > 0$ entre les dates de début des cycles de production des centrales dans un ensemble $\mathbf{A17}$. Les contraintes, CT18 imposent que les unités d'un ensemble $\mathbf{A18}$ doivent être espacés d'au moins $\mathbf{Se18}$ semaines entre la fin d'arrêt et le début d'un autre arrêt de cet ensemble de centrales. Des contraintes d'utilisation de ressources pendant un arrêt sont modélisées dans CT19, portant sur l'utilisation d'une ressource (équipe de maintenance, outil, ...), en quantité limitée $\mathbf{Q19}$. Cette ressource est utilisée pour un arrêt (i, k) entre la semaine $\mathbf{L19}_{i,k}$ et $\mathbf{L19}_{i,k} + \mathbf{Tu19}_{i,k}$, de l'arrêt où $\mathbf{Tu19}_{i,k}$ est la durée d'utilisation de la ressource en semaine. Enfin, les contraintes CT20 et CT21 limitent le nombre d'arrêts en parallèle, CT20 par un nombre maximal d'arrêts en parallèle, CT21 avec une puissance maximale d'indisposition. La contrainte CT20 impose pour toute semaine w un nombre maximal d'arrêt en parallèle $\mathbf{N20}_w$ sur un ensemble de réacteurs $\mathbf{A20}_w$.

2.4.2 Jeux de données du challenge ROADEF

Les jeux de données du challenge ROADEF sont non confidentiels et publics, représentatifs du problème réel à EDF. Les jeux de données A1 – A5 correspondent à des instances réduites pour la phase qualificative.

Equipe	S22	S24	J06	J08	S23	S17	S10	S21	Brnd	J16	S04	S08	S11	S16
Type	rls	rls	rls	aco	ext	ext	ext	hrc	hrc	hrc	dcp	dcp	dcp	dcp
B&Bnd					X	X	X	X				X	X	
Coupes							X	X						X
PL					X		X			X	X	X	X	
PPC			X						X	X				X
Pr Dyn					X							X		
Hiérar.					X	X	X	X	X	X				
Repar.			X				X		X	X		X		
Greedy	X	X				X		X	X	X	X			X
R. loc.	X	X	X	X							X	X	X	X
Popul.				X										
B6	0,59	0,00	2,50	2,64	1,54	19,4	7,39	4,14	5,37	11,1	8,79	11,4	9,02	4,55
B7	0,00	0,20	0,95	1,80	0,88	25,2	32,2	3,15	4,20	13,7	6,85	15,5	7,24	3,46
B8	0,05	0,00	NS	9,16	1,31	39,7	NS	6,30	5,35	NS	15,6	81,6	NS	NS
B9	0,13	1,06	2,12	11,2	2,28	56,7	NS	6,3	6,53	NS	27,3	24,6	NS	68,7
B10	0,30	0,00	3,11	2,25	1,24	34,5	61,2	3,70	4,91	12,3	8,51	51,8	9,62	7,22
Moyen	0,21	0,25	NS	5,44	1,45	35,1	NS	4,73	5,3	NS	13,5	36,8	NS	NS

TABLE 2.4 – Caractéristiques des approches des équipes du challenge ROADEF

Après qualification, les équipes recevaient les instances $B6 - B10$, correspondant à la taille réelle des problèmes traités par EDF. Les instances $X11 - X15$ étaient des données secrètes, utilisées pour l'évaluation des approches du challenge sur des instances non connues a priori. Les pas de temps dans les instances étaient de $24h$ pour les jeux d'instances A, et de $8h$ pour les jeux d'instances B et X. Si les instances X devaient avoir les mêmes caractéristiques que les instances B, on s'aperçoit que cela n'est vérifié que sur les caractéristiques I, J, K, T, S . Cela n'est pas le cas sur l'amplitude globale des fenêtres de temps, paramètre majeur la combinatoire du problème, cf tableau 2.3 . Ainsi, les instances $B8 - B9$ ont de loin les plus fortes combinatoires, ne considérant pas de contraintes de fenêtres de temps au-delà du second cycle. Ces instances semblent plus représentatives du problème complet d'EDF, les fenêtres de temps pour les cycles lointains étant imposées pour restreindre la combinatoire. Pour traiter des instances de taille plus petites, on a tronqué les instances précédentes en ne considérant au plus que 3 arrêts et un horizon de temps de 120 semaines. Les nombres de contraintes d'ordonnement de chaque type sont en annexe dans le tableau 8.2. Les contraintes d'ordonnement principales sont les contraintes d'espacement/recouvrement maximaux CT14 et CT15, et les nombres maximaux d'arrêts en parallèle par CT20. Il est à noter que les contraintes CT18, CT19 et CT16 sont peu utilisées sur les jeux de données B et X.

2.4.3 État de l'art du challenge ROADEF

Les équipes du challenge ROADEF, bien que travaillant indépendamment, ont eu des idées communes que nous précisons ici, liées à la structure du problème,. Beaucoup d'approches utilisaient un preprocessing pour réduire les fenêtres de temps. Un preprocessing exact des fenêtres de temps peut être obtenu en actualisant les contraintes de début de fenêtre de temps, en propageant les contraintes de stocks initiaux, rechargements minimaux et de productions et d'anticipations maximales. Le stock initial conditionne le stock minimal à tout instant, obtenu à production maximale sans modulation et à rechargement minimal, ce qui impose des dates au plus tôt pour démarrer les arrêts. Par un tel preprocessing, des arrêts peuvent être retirés lorsque leur date de réalisation au plus tôt dépasse l'horizon de temps considéré.

Du fait de l'absence de longueur maximale d'un cycle de production avec des phases admissibles de production nulle après les phases de stretch, il n'existe pas de propagation de dates au plus tard. Cependant, de telles situations ne sont pas économiquement rentables, comme le coût de production nucléaire est inférieur au coût de production thermique, et ces situations sont très marginales. En interdisant de telles situations, ou en bornant un nombre de semaine maximal à production nulle, propager les contraintes permet également de définir de nouvelles dates au plus tard de fenêtres de temps. Ce preprocessing est en toute rigueur heuristique, et élimine des solutions de coûts économique a priori peu rentable, comme la production nucléaire est plus rentable que la production T1/ En contrepartie, pour une résolution facilitée

de manière significative.

Une des grandes difficultés du challenge est la grande taille des données. Les agrégations naturelles des pas de temps de production à la semaine ont ainsi souvent été utilisées. De même, l'agrégation des scénarios stochastiques sur le scénario moyen a été souvent utilisée. Pour réduire le nombre de scénarios, l'approche [58] s'est intéressée à la réduction à 3 scénarios comprenant le scénario moyen, et les scénarios ayant les demandes cumulées minimales et maximales. Leurs tests concluent qu'il s'agit d'un bon compromis de réduction des scénarios pour ne pas dégrader la qualité des solutions.

En l'absence de contraintes dynamiques sur la production T1, le placement de la production T1 à production T2 fixée est un programme linéaire. Mieux, à chaque pas de temps, on a des problèmes de sac à dos indépendants. La résolution exacte est alors un algorithme glouton en triant sur les coûts marginaux de production étant optimale. Les unités T1 sont équivalentes à considérer une unité T1 agrégée, de capacité de production égale à la somme des capacités de production et dont le coût de production suit une courbe convexe linéaire par morceaux dont les pentes sont les coûts de production de chaque unité triés par ordre croissant, et l'amplitude de l'intervalle correspond aux capacités de production des unités correspondantes.

On distingue trois grands types d'approches pour le challenge ROADEF comme analysé dans le tableau 2.4 : les approches exactes (**ext**) utilisant des résolutions exactes sur des modèles simplifiés, les approches de recherche locale simple (**rls**), sans simplification ni décomposition, et les approches heuristiques de décomposition (**dcp**) qui itèrent suivant la structure à deux niveaux du problème. Les résultats globaux du challenge montrent que les approches heuristiques **rls** sont les plus efficaces dans la recherche de solutions primales, les autres approches étaient limitées par les temps de résolution des sous-problèmes.

Approche TU Denmark, décomposition de Benders L'approche développée dans [91] est la seule approche exacte du challenge qui n'agrègeait pas les scénarios stochastiques, pour une résolution par décomposition de Benders. Pour rentrer dans le cadre d'hypothèse de la décomposition de Benders, et ne pas avoir de variables binaires dans les sous-problèmes, les contraintes de stretch et de modulations maximales (CT12 et CT6) ont été omises. Le problème maître de la décomposition comprend comme variables liantes les dates d'arrêts et les quantités de rechargement, tandis que les sous-problèmes indépendants pour chaque scénario stochastique comprennent les variables dépendant des scénarios, les productions et les niveaux de combustible. L'approche s'est avérée inefficace sur les instances de tailles réelles B et X, pour des raisons de temps de calcul, mais aussi de consommation d'espace mémoire. Ce travail présente un preprocessing exact sur les débuts de fenêtres de temps, et un preprocessing heuristique pour une accélération conséquente des temps de calcul pour peu de perte de performances. Pour des raisons de taille de calcul et d'espace mémoire, les pas de temps de production ont été agrégés à la semaine. La résolution est une heuristique dérivée de la décomposition de Benders : seule la relaxation continue est calculée de manière exacte. Les coupes générées servent alors à calculer le problème maître sans rajouter de coupes par la suite. Une population de solutions est récupérée, pour évaluation et réparation avec CT6, CT12 et les pas de temps en désagrégés.

Approche LIPN, génération de colonnes L'approche [109] est une approche hiérarchique dont les décisions majeures sont prises sur une formulation étendue du problème déterministe en agrégeant les scénarios sur un scénario moyen. Cette approche a été la mieux classée des approches exactes dans le classement du challenge, à 2% des meilleures solutions. La première phase déterministe, sur pas de temps et scénarios stochastiques agrégés fixe pour toute la suite les dates d'arrêts et les semaines d'entrée au bore nul. La formulation étendue pour une résolution par génération de colonne se base sur la décomposition de Dantzig-Wolfe naturelle en dualisant les contraintes couplantes CT1 et CT14-CT21. Les sous-problèmes ont alors pour but de générer un planning complet d'un réacteur donné. Les contraintes de stretch sont modélisées exactement, et le sous-problème se calcule par programmation dynamique comme un problème

de plus court chemin dans un graphe défini par des état discrétisés de production. Pour des raisons de limitation de temps de calcul, l'approche usuelle de Branch&Price ne pouvait pas être déployée. L'approche de génération de colonne n'est déployée que pour calculer la relaxation continue sans les contraintes couplantes d'ordonnement. Une fois la relaxation linéaire de la formulation étendue calculée, la résolution entière se base sur le PMR avec les contraintes d'ordonnement et les colonnes générées, sans génération ultérieure de nouvelles colonnes. Les dates d'arrêts et les semaines d'entrée dans le bore nul obtenues ne seront plus remises en question, ce qui semble de nature à dégrader la qualité de solutions trouvées. Cela aboutit à des problèmes découplés scénarios par scénarios, où les profils décroissants sont fixés, la résolution étant alors par PL.

Approches heuristiques constructives Les approches heuristiques nécessitaient au moins une phase constructive pour obtenir une solution initiale, non fournie par le challenge. Les heuristique constructive étaient basées sur des stratégies gloutonnes, ou sur une décomposition hiérarchique du problème. Les approches heuristiques purement constructives (J16 et S21), sans recherche locale, ont eu des résultats bien moins bons que les approches complétant avec une recherche locale. S21 a néanmoins été l'approche gagnante du challenge, dû à sa robustesse là où les approches plus performantes ont chuté sur des instances du jeu secret X. Cette approche avait l'avantage de donner rapidement des solutions. La première étape calcule les dates d'arrêts par PLNE avec en considérant les contraintes d'ordonnement CT13-21, incitant par la fonction objectif à avoir un nombre maximal de réacteurs actifs. Afin de ne pas avoir d'infaisabilités, des itérations sont effectuées pour remonter des infaisabilités causées par les contraintes de stock CT11, en ajoutant des coupes violées. Les dates d'arrêts calculées ne sont plus remises en question. Les recharges sont affectées à leur plus hautes valeurs permettant de vérifier les contraintes de stocks et les besoins en modulation. La production est enfin placée par un algorithme glouton, où des modulations ne sont effectuées que quand la puissance nucléaire disponible est supérieure à la demande.

Approches de recherche locale simple Les deux approches qui se démarquaient en donnant les meilleurs résultats étaient des approches de recherche locale simple, S24, sur une approche de recuit simulé et S22 selon [57] et l'approche de recherche locale du Local Solver. Les points cruciaux de ce type d'approche sont les choix des voisinages, et la possibilité de calculer rapidement le coût de la solution modifiée localement. De telles approches peuvent se déplacer dans des ensembles de solutions non réalisables, en pénalisant la violation de contraintes dans la fonction objectif. Sur la première phase, S24 obtenait des résultats médiocres, en considérant comme voisinage un unique déplacement d'un arrêt, de 1 semaine. De tels voisinages donnaient des minimas locaux de mauvaise qualité. Pour expliquer cela, de tels voisinages ne remettent pas en question les ordres des arrêts lorsque les contraintes d'ordonnement CT14-CT21 s'appliquent. Des voisinages multiples apparaissaient, pour une approche VNS utilisant des types de voisinages permettant d'ajouter ou d'enlever un dernier arrêt du planning, de déplacer un seul arrêt choisi aléatoirement, et de pouvoir échanger l'ordre de début de 2 arrêts. Pour l'approche finale de [57], les voisinages étaient les suivants, pour permettre de calculer rapidement les coûts associés à de tels déplacements élémentaires :

- "k-moveOutagesRandom" : k arrêts sont sélectionnés aléatoirement et bougés aléatoirement dans des domaines réalisables vis à vis des contraintes de stock CT11 et des fenêtres de temps CT13.
- "k-moveOutagesConstrained" : k arrêts sont sélectionnés et bougés aléatoirement parmi des unités T2 impliquées dans les sous ensembles des contraintes d'ordonnement CT14-CT21.
- "k-moveOutagesConsecutive" : Pour une unité T2 sélectionnée aléatoirement, k arrêts consécutifs sont sélectionnés aléatoirement et déplacés aléatoirement à des dates d'arrêt réalisables.

Approches heuristiques de décomposition De nombreuses heuristiques sont basées sur la décomposition naturelle du problème à deux niveaux. Ce type d’approche, effectuée des itérations entre un problème d’ordonnement des arrêts, résolu par PLNE, Programmation par Contraintes ([35]) ou recherche locale pour fournir des dates d’arrêts, avec des problèmes de production à planning d’arrêt fixé, souvent résolu par un algorithme glouton ou à l’aide de Programmation Linéaire. On peut remarquer que ce processus de décomposition est présente des analogies avec la décomposition de Benders. Du problème de production, des coupes de Benders ”no good cut” peuvent être remontées dans le problème d’ordonnement, ce qui a été implémenté par l’équipe S16. De telles approches de décomposition étaient handicapées par les temps de calcul nécessités à chaque itération. Dans le temps imparti pour le challenge ROADEF, les approches de recherche locale simples, permettant des itérations plus rapides et d’explorer plus de solutions s’avéraient alors bien plus efficaces.

Approche heuristique de population Une seule approche du challenge a utilisé une approche heuristique avec des populations de solution, il s’agissait de J08, construisant une première solution réalisable avec l’algorithme de colonies de fourmis. Cette approche fournissait des résultats en utilisant des capacités mémoires bien plus élevées que celle allouée pour le challenge ROADEF. Le caractère limitant de l’espace mémoire pour le challenge ont ainsi peut être mis de côté les approches heuristiques de populations.

Instances	Primal	Brandt1	% Gap	Brandt2	% Gap
A1	169538	160847	5,13	165560	2,35
A2	146048	130148	10,89	139991	4,15
A3	154429	137744	10,80	148454	3,87
A4	111591	82428	26,13	102326	8,30
A5	125822	94379	24,99	112467	10,61
Total A	707428	605546	14,40 %	668798	5,46%
B6	83424	36338	56,44	69592	16,58
B7	81174	38263	52,86	68528	15,58
B8	81926	28410	65,32	62594	23,60
B9	81750	30000	63,30	63991	21,72
B10	77767	30389	60,92	63747	18,03
X11	79116	33377	57,81	66931	15,40
X12	77589	36740	52,65	66558	14,22
X13	76449	25789	66,27	62155	18,70
X14	76172	26903	64,68	63045	17,23
X15	75101	28444	62,13	61866	17,62
Total B et X	790468	314653	60,19%	649007	17,90%
Total	1497896	920199	38,57%	1317805	12,02%

TABLE 2.5 – Meilleures solutions primales et bornes duales du challenge ROADEF

Meilleures bornes primales et duales Le tableau 2.5 recense les meilleures bornes primales améliorées après le challenge. Il se posait également les questions du calcul de bornes duales. Seul [35] a fourni des bornes duales. La colonne Brandt1 fournit des bornes duales obtenues rapidement, tandis qu’en s’autorisant de longs temps de calculs, leur approche fournissait les valeurs de la colonne Brandt2. Les approches basées sur les approches exactes n’ont pas fourni de bornes duales. Garantir des bornes inférieures du challenge ROADEF était incompatible avec certaines hypothèses restrictives du problème. Ainsi, les approches de

type génération de colonne en discrétisant la production dans le calcul des sous-problèmes ne permettent pas d'avoir des bornes inférieures valables pour le problème ROADEF, en restreignant le domaine réalisable. L'approche de [91] sans preprocessing heuristique pourrait théoriquement fournir des bornes duales, l'approche se heurtait alors à des limites de temps de calcul et d'espace mémoire.

2.4.4 Etat de l'art de la résolution à EDF

Cette section retrace l'état de l'art existant à EDF R&D sur le problème, l'outil déployé à l'opérationnel ORION, et les travaux de recherches menés en interne ou en partenariat.

Le prétraitement FUSCOU Pour des raisons de taille et de temps de calcul, un prétraitement est utilisé à l'opérationnel pour aboutir à des instances tractables. Le prétraitement FUSCOU se ramène à une situation déterministe, agrégeant les scénarios stochastiques de demandes, pannes et prix de marchés. De plus, l'optimisation est réalisée sur une granularité hebdomadaire pour les niveaux de production, les données de demandes sont ainsi agrégées à la semaine. Un des enjeux de la formulation du challenge ROADEF était de pouvoir chiffrer l'approximation engendrée par l'agrégation de FUSCOU.

ORION, l'algorithme en exploitation ORION, l'approche en exploitation à EDF est une approche analogue aux heuristiques de décompositions du challenge ROADEF. PLAFIGE résout le problème de placement de la production nucléaire à planning d'arrêt figé, selon une approche exacte de type décomposition lagrangienne. ORION LOCAL modifie localement le planning d'arrêts autour de la solution courante. ORION faisant l'objet de modifications continues en internes, et ayant évolué pendant la thèse, nous n'en donnons que les lignes générales. ORION nécessite une solution initiale, qui est fournie par le processus de réoptimisation dynamique du processus opérationnel à EDF.

Cette approche présentait de nombreux axes d'améliorations. Pendant longtemps, ORION ne comprenait pas de gestion des contraintes d'ordonnancement (de CT14 à CT21), ce qui a motivé les travaux de [80] pour incorporer de la faisabilité de certaines contraintes d'ordonnancement, à l'aide de la PPC. Les contraintes de Stretch n'étaient également pas modélisées initialement, et il se posait la question de savoir si cette simplification est impactante sur les solutions trouvées. Depuis, une modélisation analogue à celle de la thèse a été incorporée, pour des sous contraintes intéressantes pour les contraintes de stretch, fournissant uniquement la borne supérieure. Enfin, dans la modélisation d'ORION, la possibilité de faire varier les recharges n'était pas prise en compte.

L'inconvénient majeur de la recherche locale d'ORION est qu'elle considère de petits voisinages, ce qui se traduit par l'existence de beaucoup de minimums locaux, ce qui nuit à la qualité de la solution finale, la configuration finale est assez proche de la configuration initiale. Derrière cela, se cache aussi une volonté de peu désorganiser le planning et de rechercher des plannings stables. Pendant longtemps, les voisinages d'ORION étaient de ne considérer que des déplacements élémentaires d'un seul arrêt, déplacement de 1 semaine. Ce type de voisinage, lorsqu'il était utilisé par les approches du challenge est connu pour donner beaucoup de minima locaux de piètre qualité. Une problématique qui intéressait à EDF R&D était d'autoriser des situations d'interversions d'arrêts ou des "sauts d'hiver", changements radicaux de planification d'un arrêt permettant de le déplacer de l'automne au printemps sans passer par l'hiver où les coûts de substitution du nucléaire sont les plus élevés.

TANNAT, approche frontale PLNE Une approche frontale en PLNE a déjà été étudiée à EDF R&D, par le projet TANNAT ([90]). La résolution se basait également sur les sorties du preprocessing de FUSCOU, traitant ainsi un problème de nature déterministe après agrégation. La modélisation fournie permettait de prendre en compte toutes les contraintes d'ordonnancement des arrêts, de modéliser des contraintes

de stretch, en se passant des approximations hiérarchiques de l'approche ORION. L'inconvénient d'une telle approche était les temps de calculs et la combinatoire limitée que pouvait gérer l'approche. Ainsi, l'utilisation préconisée était de limiter la recherche autour d'un planning initial, et d'itérer des itérations d'améliorations locales avec des fenêtres de temps très restreintes. De tels voisinages locaux ne permettaient pas les "sauts d'hiver", ou limitaient les possibilités d'interventions d'ordres d'arrêts. Ce projet n'a pas été poursuivi et incorporé dans la résolution opérationnelle. Avec les progrès faits depuis sur l'algorithmie et les capacités de calculs, de forts progrès s'en suivraient sur une telle approche.

Approche INRIA Bordeaux L'INRIA Bordeaux a démarré le challenge et participé à la phase qualificative avant de s'en écarter, comme un contrat était en cours d'être signé avec EDF ce qui ne permettait pas de participer au challenge. Des travaux préparatoires se retrouvent publiés en annexe de la thèse [74]. Ce travail fournit d'abord une modélisation compacte du problème avec toutes les contraintes, au prix de nombreuses variables binaires qui rendent la formulation intractable pour une résolution frontale. L'approche envisagée pour le challenge et poursuivie par la suite était à l'instar de l'approche du LIPN, de placer les dates d'arrêts sur une formulation déterministe sur les scénarios agrégés à la moyenne, et en agrégeant les pas de temps, suivant une formulation étendue par génération de colonne permettant de modéliser exactement le stretch. Les considérations d'espace mémoire s'avéraient critiques même sur un modèle agrégé, déjà pour le calcul de la relaxation continue, avec des graphes de grande taille pour les sous-problèmes, et une génération de grand nombres de colonnes pour le calcul de la relaxation continue. Des travaux en cours en parallèle de la thèse et en partenariat avec EDF R&D consistaient à incorporer de la robustesse dans la modélisation vis à vis de prolongations d'arrêt. Contrairement à notre modélisation de la robustesse du chapitre 8, leur modélisation approchée ne considérait que des questions de coûts liés aux prolongations d'arrêt, en négligeant les questions de faisabilités, ce que notre approche robuste a respecté, cela correspondant au besoin opérationnel.

Autres travaux de recherche menés par EDF R&D Le problème de placement des arrêts nucléaires, au vu de ses enjeux financiers pour EDF, concentre beaucoup d'efforts de recherche. Plusieurs thèses CIFRE ont été motivées par la problématique, pour explorer plusieurs types d'approches. La Programmation Par Contraintes a été investigué dans [80] dans le but d'introduire dans ORION la faisabilité de contraintes d'ordonnancement. Le problème traité était assez analogue au problème d'ordonnancement traité dans [77], contenant les contraintes d'ordonnancement, et essayant de remonter des contraintes induites par la production. La programmation conique, semi-définie positive (SDP), a été investiguée dans [64]. L'idée était que le formalisme SDP pouvait bien modéliser des contraintes disjonctives (l'arrêt A est avant l'arrêt B de s semaines ou réciproquement). Une telle approche n'est pas mature pour une résolution d'un si gros problème industriel, la SDP ne disposant pas de solveurs industriels comme pour la PLNE.

Deuxième partie

Problème de planification journalière de la production

Chapitre 3

Résolution exacte d'UCPd en PLNE

Résumé : Ce chapitre améliore les formulations compactes du problème UCPd obtenues à l'issue de [51]. Les contraintes de durées minimales de paliers sont renforcées avec des considérations de coupes de cliques. Une comparaison polyédrale est réalisée à l'aide d'un isomorphisme permettant de plonger et comparer les formulations de contraintes, et enfin de renforcer des contraintes par projection. Les implémentations comparent la convergence pratique des formulations polyédralement équivalentes par l'algorithme de Branch&Bound suivant la détection de structures spécifiques par Cplex, pour aboutir au choix de formulation compacte le plus adapté. La reformulation étendue de Dantzig-Wolfe par dualisation des contraintes de demandes est également investiguée avec sa résolution par génération de colonnes. Ce travail a fait l'objet de communications aux congrès COPI 2010, JPOC 2010 et d'une soumission à la revue EJCO (EURO Journal of Combinatorial Optimization).

On a défini à la section 2.3 deux formulations PLNE compactes du problème UCPd suivant la définition des variables, par niveaux ou par états. À l'issue de [51], la comparaison des formulations était empirique, et il a été observé que la formulation par niveaux qui donnait les convergences les plus rapides. De plus, les reformulations étendues de génération de colonnes n'avaient pas été étudiées, reportées aux travaux de cette thèse. Un premier travail a été d'investiguer le choix de formulation PLNE adéquat pour la résolution exacte d'UCPd. Dans un premier temps, la formulation par niveaux a pu être encore renforcée. Dans un second temps, on a pu obtenir des résultats théoriques sur la comparaison des formulations par niveaux et par états, et renforcer la formulation par états. Dans un troisième temps, la reformulation de Dantzig-Wolfe a été implémentée. Ce chapitre fournit des résultats à la fois pratiques et théoriques.

3.1 Renforcements de la formulation par niveaux

La formulation par niveaux a pu être encore renforcée par rapport à ce qui avait été obtenu dans [51], avec la formulation rappelée à la section 2.3. Les contraintes de durées minimales sur les points de fonctionnement ont pu être renforcées, et un couplage a été exhibé entre les variables de démarrage.

3.1.1 Formulation renforcée par regroupement sur des cliques

Avec les variables de niveaux, quatre jeux de contraintes exprimaient les contraintes de durées minimales sur les points de fonctionnement, (2.40),(2.41),(2.42) et (2.43). On arrivera à exhiber un jeu de contraintes renforcées comprenant toutes ces contraintes.

Lemme 3.1.1 *Les démarrages des membres de gauche de (2.40) et (2.41) sont incompatibles :*

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \quad \sum_{t'=t+1}^{t+\Delta_u^{(i)+}} y_{u,t'}^{(i+1)+} + \sum_{t'=t+1}^{t+\Delta_u^{(i)-}} y_{u,t'}^{(i-1)-} \leq 1 \quad (3.1)$$

PREUVE : Soit $u \in \mathcal{U}, t \in \mathcal{T}, i \in \mathcal{I}_u$. Soit $t' \in \llbracket t+1, t+\Delta_u^{(i)+} \rrbracket$, soit $t'' \in \llbracket t+1, t+\Delta_u^{(i)-} \rrbracket$. Il s'agit de montrer que $y_{u,t'}^{(i+1)+} + y_{u,t''}^{(i-1)-} \leq 1$. Dans le cas où $t' = t''$, le résultat est acquis, il ne peut y avoir simultanément démarrage du palier $i-1$ et du palier $i+1$. On supposera $t' < t''$, l'autre cas étant analogue. On suppose, par l'absurde que $y_{u,t'}^{(i+1)+} + y_{u,t''}^{(i-1)-} = 2$. Le palier $i-1$ est atteint après une baisse de puissance à partir du palier i . La puissance a ainsi dû être stabilisée sur le palier i pendant au moins $\Delta_u(i, i-1)$ pas de temps. Il en résulte que $t' + \Delta_u^{(i)-} \leq t''$. On a alors $t+1 + \Delta_u^{(i)-} \leq t''$ ce qui est exclu. On a ainsi montré pour tout $u \in \mathcal{U}, t \in \mathcal{T}, i \in \mathcal{I}_u : \forall t' \in \llbracket t+1, t+\Delta_u^{(i)+} \rrbracket, \forall t'' \in \llbracket t+1, t+\Delta_u^{(i)-} \rrbracket, y_{u,t'}^{(i+1)+} + y_{u,t''}^{(i-1)-} \leq 1$. Ainsi, dans le graphe de dépendance, toutes les variables de démarrages considérées sont dans une clique. Il s'en déduit

$$\text{du regroupement sur cette clique : } \forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t''=t+1}^{t+\Delta_{u,i}^-} y_{u,t''}^{(i-1)-} \leq 1 \quad \blacksquare$$

Proposition 3.1.1 *Les contraintes suivantes fournissent une formulation renforcée des contraintes de paliers sur les points de fonctionnement. Pour la formulation avec les variables d'état :*

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t''=t+1}^{t+\Delta_{u,i}^-} y_{u,t''}^{(i-1)-} \leq s_{u,t}^{(i)} \quad (3.2)$$

Avec la formulation de variables de niveaux, on a les contraintes suivantes, qui comprennent également les contraintes de précédence (2.33) :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t''=t+1}^{t+\Delta_{u,i}^-} y_{u,t''}^{(i-1)-} + x_{u,t}^{(i+1)} \leq x_{u,t}^{(i)} \quad (3.3)$$

PREUVE : À partir des inégalités (2.40), (2.42) et (3.1), et respectivement (2.41), (2.43) et (3.1), l'argument de regroupement de ces trois cliques déjà utilisé permet d'écrire des contraintes plus fortes :

$$\begin{aligned} \forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t''=t+1}^{t+\Delta_{u,i}^-} y_{u,t''}^{(i-1)-} &\leq x_{u,t}^{(i)} \\ \forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t''=t+1}^{t+\Delta_{u,i}^-} y_{u,t''}^{(i-1)-} &\leq 1 - x_{u,t}^{(i+1)} \end{aligned}$$

Il reste à présent à remarquer que les variables $x_{u,t}^{(i)}$ et $1 - x_{u,t}^{(i+1)}$ sont incompatibles par la contrainte de précédence (2.33). Les deux cliques précédentes sont alors en fait incluses dans une même clique, et on a alors l'écriture suivante obtenue par renforcement :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t''=t+1}^{t+\Delta_{u,i}^-} y_{u,t''}^{(i-1)-} + x_{u,t}^{(i+1)} \leq x_{u,t}^{(i)} \quad (3.4)$$

L'équation (3.4) donne alors une caractérisation renforcée qui remplace les contraintes de paliers (2.40), (2.41), (2.42) et (2.43) et des contraintes d'ordres (2.33).

3.1.2 Couplages entre variables binaires et démarrages

En fait de manière analogue à la relation d'égalité observée dans [105] entre variables de fonctionnement, de démarrage et d'arrêts d'une centrale, $z_t^u = y_t^u + x_{t-1}^u - x_{u,t}$, on a avec la structure de paliers superposés une relation analogue :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, y_{u,t}^{(i-1)-} = y_{u,t}^{(i)+} + x_{u,t-1}^{(i)} - x_{u,t}^{(i)} \quad (3.5)$$

La contrainte (3.5) peut alors remplacer les contraintes (2.36) et (2.37) pour un couplage renforcé. On remarque alors que la famille des $y_{u,t}^{(i)-}$ est inutile, les $y_{u,t}^{(i)-}$ peuvent être remplacés par leur expression (3.5) dans toutes les contraintes. La contrainte (2.37) est alors inutile et peut être enlevée.

3.1.3 Formulation renforcée

A l'issue du travail de cette section, les formulations avec les variables de paliers ont été renforcées. La formulation renforcée avec les variables de paliers s'écrit à présent :

$$\begin{aligned}
\min_{x,y} \quad & \sum_{u,t} \left(C_u^{fix} x_{u,t}^{(1)} + C_u^{dem} y_{u,t}^{(1)+} + \sum_i C_u^{ppp} (P_{u,i} - P_{u,i-1}) x_{u,t}^{(i)} \right) \\
\forall u, i, \quad & x_{u,t}^{(i)} = \mathbb{1}_{i_u^{(0)} \geq i} \\
\forall u, t, i, \quad & x_{u,t}^{(i)} \leq x_{u,t+1}^{(i-1)} \\
\forall u, t, i, \quad & x_{u,t}^{(i)} \geq x_{u,t+1}^{(i+1)} \\
\forall u, t, i, \quad & y_{u,t}^{(i-1)-} = y_{u,t}^{(i)+} + x_{u,t-1}^{(i)} - x_{u,t}^{(i)} \\
\forall u, t, i, \quad & \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t'=t+1}^{t+\Delta_{u,i}^-} y_{u,t'}^{(i-1)-} + x_{u,t}^{(i+1)} \leq x_{u,t}^{(i)} \\
\forall u, t, \quad & \sum_{t'=t-L_u+1}^t y_{u,t'}^{(1)+} \leq x_{u,t}^{(1)} \\
\forall u, t, \quad & \sum_{t'=t-l_u+1}^t y_{u,t'}^{(1)+} \leq 1 - x_{u,t-l_u}^{(1)} \\
\forall t, \quad & \sum_{u,i} (P_{u,i} - P_{u,i-1}) x_{u,t}^{(i)} \geq D_P(t) \\
\forall t, \quad & \sum_{u,i} (R_{u,i}^1 - R_{u,i-1}^1) x_{u,t}^{(i)} \geq D_t^{R1} \\
\forall t, \quad & \sum_{u,i} (R_{u,i}^2 - R_{u,i-1}^2) x_{u,t}^{(i)} \geq D_t^{R2} \\
\forall u, t, i, \quad & y_{u,t}^{(i)+}, y_{u,t}^{(i)-}, x_{u,t}^{(i)} \in \{0, 1\}
\end{aligned} \tag{3.6}$$

3.2 Comparaison polyédrale et renforcements

Cette section va permettre de comparer de manière théorique les formulations de contraintes dans les modélisations avec les variables d'état et de niveaux, en passant d'une formulation à l'autre par isomorphisme en comparant les descriptions polyédrales engendrées en projetant dans un même espace.

3.2.1 Transformation des variables d'états en variables de paliers

Soit $n \in \mathbb{N}$, et soit $\varphi : (x_1, \dots, x_j, \dots, x_n) \mapsto \left(\sum_{i=1}^n x_i, \dots, \sum_{i=j}^n x_i, \dots, x_n \right)$. φ est une application linéaire de \mathbb{R}^n dans \mathbb{R}^n qui permet de transformer les variables d'états en variables de paliers. φ a des propriétés intéressantes pour la suite :

Proposition 3.2.1 φ est bijective, c'est donc un automorphisme \mathbb{R}^n .

Sa réciproque est $\psi : (x_1, \dots, x_j, \dots, x_n) \mapsto (x_1 - x_2, \dots, x_j - x_{j+1}, \dots, x_n)$.

PREUVE : En regardant φ comme une matrice de passage, c'est une matrice triangulaire supérieure, dont les éléments diagonaux sont tous des 1. 1 est donc la seule valeur propre de φ , qui est aussitôt inversible. Il est aisé de vérifier que $\varphi \circ \psi = \psi \circ \varphi = Id$. ■

Proposition 3.2.2 Soit $\mathcal{H} = \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^n a_i x_i \leq d \right\}$ un hyperplan. $\varphi(\mathcal{H})$ est hyperplan, dont la description linéaire est : $\varphi(\mathcal{H}) = \left\{ x \in \mathbb{R}^n \mid a_1 x_1 + \sum_{i=2}^n (a_i - a_{i-1}) x_i \leq d \right\}$

PREUVE : φ étant un isomorphisme, $\varphi(\mathcal{H})$ est un hyperplan. Soit $y \in \varphi(\mathcal{H})$. $y = \varphi(x)$ et $\sum_{i=1}^n a_i x_i \leq d$. Nous avons les relations $x_i = y_i - y_{i+1}$ pour $i < n$ et $x_n = y_n$. En réécrivant l'équation de l'hyperplan avec y_i , et en réindexant la sommation, nous avons :

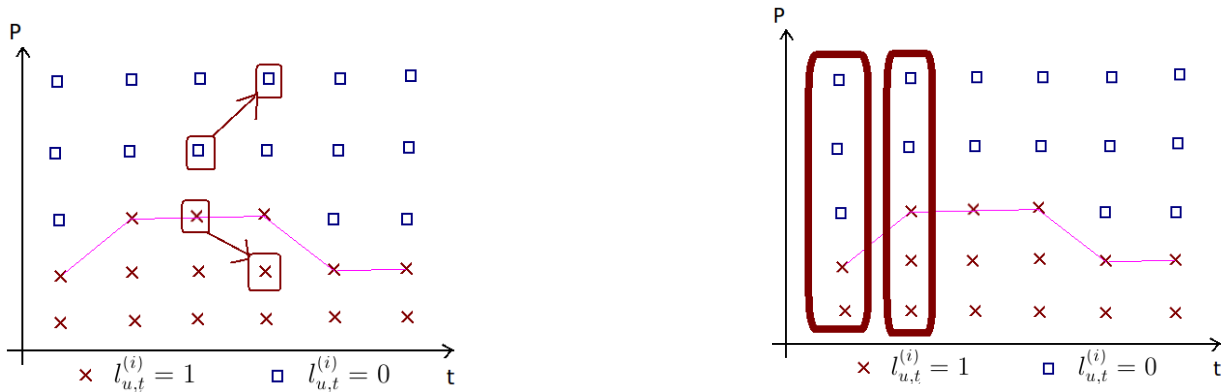
$$a_n x_n + \sum_{i=1}^{n-1} a_i (y_i - y_{i+1}) = a_1 y_1 + \sum_{i=2}^n (a_i - a_{i-1}) y_i \leq d$$

Cela prouve $\varphi(\mathcal{H}) \subset \left\{ y \in \mathbb{R}^n \mid a_1 y_1 + \sum_{i=2}^n (a_i - a_{i-1}) y_i \leq d \right\}$. Comme φ est un isomorphisme, $\varphi(\mathcal{H})$ est un hyperplan. L'inclusion réciproque est alors vérifiée par l'égalité des dimensions.

Proposition 3.2.3 Soit $\mathcal{S} = \left\{ s \in [0, 1]^n \mid \sum_{i=1}^n s_i \leq 1 \right\}$, $\mathcal{L} = \left\{ l \in [0, 1]^n \mid \forall i, l_{i+1} \leq l_i \right\}$. φ se restreint à une bijection de \mathcal{S} dans \mathcal{L} , $\varphi(\mathcal{S}) = \mathcal{L}$. les variables d'état sont transformées par φ en variables de paliers.

PREUVE : \mathcal{S} est définie comme une intersection d'hyperplans. On calcule l'image de ces hyperplans à l'aide de la proposition précédente. L'hyperplan $\mathcal{H} = \sum_{i=1}^n s_i \leq 1$, avec $(a_i) = 1$ a pour image l'hyperplan $l_1 \leq 1$. Les hyperplans $s_i \geq 0$ avec $i < n$ et $(a_j) = \mathbb{1}_{j=i}$, ont pour image les hyperplans $l_{i+1} \leq l_i$. L'hyperplan $s_n \geq 0$ avec $(a_j) = \mathbb{1}_{j=n}$, a pour image l'hyperplan $0 \leq l_n$. L'image d'une intersection d'hyperplans étant l'intersection des hyperplans images, le résultat est prouvé.

3.2.2 Comparaison polyédrale des formulations



Contraintes de la formulation par niveaux :

$$x_{u,t}^{(i)} \geq x_{u,t+1}^{(i+1)} \text{ et } x_{u,t}^{(i)} \leq x_{u,t+1}^{(i-1)}$$

Contraintes de la formulation par états :

$$-1 \leq \sum_i (s_{u,t}^{(i)} - s_{u,t+1}^{(i)}) \leq 1$$

FIGURE 3.1 – Contraintes de transitions réalisables

On définit les polyèdres suivants, selon les descriptions de contraintes de la section 2.3 sur les deux formulations de variables après renforcement des contraintes de durées minimales de paliers. L'objectif de cette section est de comparer ces descriptions polyédrales.

$$\mathcal{P}^{stt} = \left\{ \left(s_{u,t}^{(i)}, y_{u,t}^{(i)+}, y_{u,t}^{(i)-} \right)_{u,t,i} \in [0, 1] \mid (2.19 - 2.29 - 2.26), (2.22), (3.2) \right\}$$

$$\mathcal{P}^{lvl} = \left\{ \left(l_{u,t}^{(i)}, y_{u,t}^{(i)+}, y_{u,t}^{(i)-} \right)_{u,t,i} \in [0, 1] \mid (2.32), (8.3), (2.44 - 2.35), (3.3) \right\}$$

$$\text{Soit } \Phi : \left(x_{u,t}^{(i)}, y_{u,t}^{(i)+}, y_{u,t}^{(i)-} \right) \mapsto \left(\varphi(x_{u,t}^{(i)}), y_{u,t}^{(i)+}, y_{u,t}^{(i)-} \right).$$

Proposition 3.2.4 Φ hérite des propriétés de φ :

(1) Φ est bijective, et réalise ainsi un automorphisme de \mathbb{R}^n . Sa réciproque est :

$$\Phi^{-1} : \left(x_{u,t}^{(i)}, y_{u,t}^{(i)+}, y_{u,t}^{(i)-} \right) \mapsto \left(\varphi^{-1}(x_{u,t}^{(i)}), y_{u,t}^{(i)+}, y_{u,t}^{(i)-} \right)$$

(2) L'image par Φ de l'hyperplan $\sum_u \sum_t \sum_i \left(a_{u,t}^{(i)} x_{u,t}^{(i)} + b_{u,t}^{(i)} y_{u,t}^{(i)+} + c_{u,t}^{(i)} y_{u,t}^{(i)-} \right) \leq d$ est l'hyperplan

$$\sum_u \sum_t \sum_i \left((a_{u,t}^{(i)} - a_{u,t}^{(i-1)}) x_{u,t}^{(i)} + b_{u,t}^{(i)} y_{u,t}^{(i)+} + c_{u,t}^{(i)} y_{u,t}^{(i)-} \right) \leq d \text{ avec l'extension } a_{u,t}^{(0)} = 0.$$

PREUVE : (1) découle directement du fait que φ est un automorphisme de \mathbb{R}^n . (2) se prouve de manière similaire que pour la proposition 3.2.2.

Théorème 3.2.1 La formulation avec les variables de niveaux est plus forte : $\mathcal{P}^{lvl} \subset \Phi(\mathcal{P}^{stt})$.

En projetant les hyperplans des contraintes de la formulation par niveaux en hyperplans dans l'espace des variables d'état à l'aide de Φ^{-1} , on obtient des contraintes plus fortes dans la formulation par niveaux. Pour lier les variables de démarrages aux variables de fonctionnement, on a les contraintes suivantes :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \quad y_{u,t}^{(i-1)-} + \sum_{j \geq i} \left(s_{u,t}^{(j)} - s_{u,t-1}^{(j)} \right) = y_{u,t}^{(i)+} \quad (3.7)$$

Les contraintes suivantes fournissent des contraintes plus fortes pour les transitions réalisables :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \quad \sum_{j \geq i} s_{u,t}^{(j)} \leq \sum_{j \geq i-1} s_{u,t+1}^{(j)} \quad (3.8)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall i \in \mathcal{I}_u, \quad \sum_{j \geq i} s_{u,t}^{(j)} \geq \sum_{j \geq i+1} s_{u,t+1}^{(j)} \quad (3.9)$$

Avec ces contraintes dans la formulation par état, les deux formulations sont polyédralement équivalentes.

PREUVE : En appliquant le point(2) de la proposition précédente à des $(a_{u,t}^{(j)}, b_{u,t}^{(j)}, c_{u,t+1}^{(j)})$ bien choisis, on peut prouver que les hyperplans (2.32),(2.33),(2.44), (2.45), (2.46),(2.38),(2.39),(3.3) (3.5),(2.34),(2.35) sont respectivement les images par Φ des hyperplans (2.18),(2.19),(2.27), (2.28), (2.29), (2.25),(2.26) et (3.2), (3.7),(3.8),(3.9)

On définit $\mathcal{P}_2^{stt} = \left\{ (s, y^+, y^-) \in [0, 1]^N \mid (2.18), (2.19), (2.25), (2.26), (2.27 - 2.29), (3.2), (3.7 - 3.9) \right\}$.

On écrit $\mathcal{P}_2^{stt} = \bigcap_j \mathcal{H}_j$, comme une intersection de demi-espaces. $\Phi(\mathcal{P}_2^{stt}) = \Phi\left(\bigcap_j \mathcal{H}_j\right) = \bigcap_j \Phi(\mathcal{H}_j)$. Cela prouve ainsi $\Phi(\mathcal{P}_2^{stt}) = \mathcal{P}^{lev}$. Dans la suite, nous montrons donc $\mathcal{P}_2^{stt} \subset \mathcal{P}^{stt}$, ce qui induit $\Phi(\mathcal{P}^{stt}) \subset \Phi(\mathcal{P}_2^{stt}) = \mathcal{P}^{lev}$. Soit $(s, y^+, y^-) \in \mathcal{P}_2^{stt}$, nous devons prouver que (s, y^+, y^-) vérifie (2.20),(2.21),(2.22).

$y_{u,t}^{(i-1)-} + \sum_{j \geq i} \left(s_{u,t}^{(j)} - s_{u,t-1}^{(j)} \right) = y_{u,t}^{(i)+}$, donc $\sum_{j \geq i} \left(s_{u,t}^{(j)} - s_{u,t-1}^{(j)} \right) \leq y_{u,t}^{(i)+}$ ie $s_{u,t}^{(i)} + \sum_{j \geq i+1} s_{u,t}^{(j)} \leq y_{u,t}^{(i)+} + \sum_{j \geq i} s_{u,t-1}^{(j)}$ $s_{u,t-1}^{(i-1)} + s_{u,t}^{(i)} \leq s_{u,t-1}^{(i-1)} + \sum_{j \geq i} s_{u,t-1}^{(j)} - \sum_{j \geq i+1} s_{u,t-1}^{(j)} + y_{u,t}^{(i)+}$. Avec (2.19), $s_{u,t-1}^{(i-1)} + \sum_{j \geq i} s_{u,t-1}^{(j)} = \sum_{j \geq i-1} s_{u,t-1}^{(j)} \leq 1$, so $s_{u,t-1}^{(i-1)} + s_{u,t}^{(i)} \leq 1 - \sum_{j \geq i+1} s_{u,t-1}^{(j)} + y_{u,t}^{(i)+} \leq 1 + y_{u,t}^{(i)+}$ ie (2.20) est vérifié. On prouve de manière analogue que (s, y^+, y^-) vérifie (2.21).

Prouvons à présent que s vérifie (2.22). $s_{u,t+1}^{(i-1)} + s_{u,t+1}^{(i)} + s_{u,t+1}^{(i+1)} = \sum_{j \geq i-1} s_{u,t+1}^{(j)} - \sum_{j \geq i+2} s_{u,t+1}^{(j)}$. Avec (3.8),(3.9) $\sum_{j \geq i-1} s_{u,t+1}^{(j)} \geq \sum_{j \geq i} s_{u,t}^{(j)}$ et $\sum_{j \geq i+2} s_{u,t+1}^{(j)} \leq \sum_{j \geq i+1} s_{u,t}^{(j)}$. Donc $s_{u,t+1}^{(i-1)} + s_{u,t+1}^{(i)} + s_{u,t+1}^{(i+1)} \geq \sum_{j \geq i} s_{u,t}^{(j)} - \sum_{j \geq i+1} s_{u,t}^{(j)} = s_{u,t}^{(i)}$, s vérifie (2.22).

A l'issue de ce travail, on a prouvé que la formulation de la section 3.1.3 est équivalente ou domine les autres formulations. La projection cette formulation sur l'espace des variables d'état permet d'obtenir la formulation équivalente :

$$\begin{aligned}
\min_{x,y} \quad & \sum_{u,t} C_u^{dem} y_{u,t}^{(1)+} + \sum_i \left(C_u^{ppr} P(u,i) + C_u^{fix} \right) s_{u,t}^{(i)} \\
\forall u, i \quad & s_{u,0}^{(i)} = \mathbb{1}_{i^{(0)}=i} \\
\forall u, t, \quad & \sum_i s_{u,t}^{(i)} \leq 1 \\
\forall u, t, i, \quad & y_{u,t}^{(i-1)-} + \sum_{j \geq i} \left(s_{u,t}^{(j)} - s_{u,t-1}^{(j)} \right) = y_{u,t}^{(i)+} \\
\forall u, t, \quad & \sum_{j \geq i} s_{u,t}^{(j)} \leq \sum_{j \geq i-1} s_{u,t+1}^{(j)} \\
\forall u, t, \quad & \sum_{j \geq i} s_{u,t}^{(j)} \geq \sum_{j \geq i+1} s_{u,t+1}^{(j)} \\
\forall u, t, i, \quad & \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t'=t+1}^{t+\Delta_{u,i}^-} y_{u,t'}^{(i-1)-} \leq s_{u,t}^{(i)} \\
\forall u, t, \quad & \sum_{t'=t-L_u+1}^t y_{u,t'}^{(1)+} \leq \sum_i s_{u,t}^{(i)} \\
\forall u, t, \quad & \sum_{t'=t-l_u+1}^t y_{u,t'}^{(1)+} \leq 1 - \sum_i s_{u,t-l_u}^{(i)} \\
\forall t, \quad & \sum_{u,i} P(u,i) s_{u,t}^{(i)} \geq D_P(t) \\
\forall t, \quad & \sum_{u,i} R_1(u,i) s_{u,t}^{(i)} \geq D_t^{R1} \\
\forall t, \quad & \sum_{u,i} R_2(u,i) s_{u,t}^{(i)} \geq D_t^{R2} \\
\forall u, t, i, \quad & y_{u,t}^{(i)+}, y_{u,t}^{(i)-}, s_{u,t}^{(i)} \in \{0, 1\}
\end{aligned} \tag{3.10}$$

3.3 Reformulation étendue de Dantzig Wolfe

La structure des contraintes du problème se prêterait à une décomposition de Dantzig-Wolfe : en dualisant les contraintes couplantes de demandes en puissance et en réserves, le problème est alors découpé avec des blocs indépendants unité par unité. On investigate ici pour UCPd la reformulation étendue de Dantzig-Wolfe et le schéma de résolution par génération de colonne correspondant comme décrits page 36.

3.3.1 Formulation étendue pour UCPd

Pour une formulation étendue, on définit \mathcal{P} comme étant l'ensemble des plans de productions réalisables pour chaque unité, \mathcal{P}^u l'ensemble des plans de production de l'unité u , $\mathcal{P} = \bigcup_u \mathcal{P}^u$. Les variables décisionnelles sont alors $\lambda_p \in \{0, 1\}$ indiquant si le plan $p \in \mathcal{P}$ est choisi dans le plan de production global. Pour tout plan $p \in \mathcal{P}$, on définit $P(p, t)$, $R_1(p, t)$, $R_2(p, t)$ les productions et réserves primaires et secondaires générées par p à tout instant t . On a alors la formulation étendue suivante pour UCPd :

$$\min_{\lambda} \quad \sum_{p \in \mathcal{P}} c_p \lambda_p \tag{3.11}$$

$$\forall t \in \mathcal{T}, \quad \sum_{p \in \mathcal{P}} P(p, t) \lambda_p \geq D_P(t) \tag{3.12}$$

$$\forall t \in \mathcal{T}, \quad \sum_{p \in \mathcal{P}} R_1(p, t) \lambda_p \geq D_t^{R1} \tag{3.13}$$

$$\forall t \in \mathcal{T}, \quad \sum_{p \in \mathcal{P}} R_2(p, t) \lambda_p \geq D_t^{R2} \tag{3.14}$$

$$\forall u_0 \in \mathcal{U}, \quad \sum_{p \in \mathcal{P}^{u_0}} \lambda_p = 1 \tag{3.15}$$

$$\forall p \in \mathcal{P}, \quad \lambda_p \in \{0, 1\} \tag{3.16}$$

3.3.2 Problème maître restreint

On calcule la relaxation continue de la formulation étendue par génération de colonne. On écrit ici la formulation du PMR, en considérant éventuellement des branchements précédents. Le branchement choisi est le branchement sur les variables de niveaux de la formulation compacte (justifié par les résultats sur la formulation compacte). Un branchement $b \in \mathcal{N}_{und}^{brch} \cup \mathcal{N}_{ovr}^{brch}$ se caractérise alors par le choix d'un pas de temps t_b , d'une unité u et d'un point de fonctionnement représenté par sa puissance P_b . Pour \mathcal{N}_{und}^{brch} (resp \mathcal{N}_{ovr}^{brch}), le branchement se caractérise par une production de l'unité u inférieure (resp supérieure) à P_b .

Du fait des contraintes (3.15), la relaxation linéaire écrit uniquement $\lambda_p \geq 0$, les contraintes de type $\lambda_p \leq 1$ étant redondantes avec (3.15), et donc ne sont pas écrites de manière à ne pas ajouter de variables duales. Pour démarrer l'algorithme de génération de colonnes, il est nécessaire d'initialiser avec un PMR réalisable lorsque des solutions existent. Pour l'initialisation sans branchements, définir un ensemble initial avec les productions minimales et maximales pour chaque tranche avec l'inertie des conditions initiales. Avec les branchements, les heuristiques du chapitre suivant peuvent aider à fournir des solutions vérifiant des branchements quand cela est réalisable, une résolution PLNE frontale pourra détecter très rapidement si les branchements mènent à une infaisabilité.

$$\begin{aligned}
\min_{\lambda_p \geq 0} \quad & \sum_{p \in \mathcal{P}} c_p \lambda_p \\
\forall t \in \mathcal{T}, \quad & \sum_{p \in \mathcal{P}} P(p, t) \lambda_p \geq D_P(t) \quad (\pi_P) \\
\forall t \in \mathcal{T}, \quad & \sum_{p \in \mathcal{P}} R1(p, t) \lambda_p \geq D_t^{R1} \quad (\pi_{R1}) \\
\forall t \in \mathcal{T}, \quad & \sum_{p \in \mathcal{P}} R2(p, t) \lambda_p \geq D_t^{R2} \quad (\pi_{R2}) \\
\forall b \in \mathcal{N}_{und}^{brch}, \quad & \sum_{p \in \mathcal{P}^{u(b)}, P(p, t_b) \geq P_b} \lambda_p \geq 1 \quad (\pi_{und}^{brch}) \\
\forall b \in \mathcal{N}_{ovr}^{brch}, \quad & \sum_{p \in \mathcal{P}^{u(b)}, P(p, t_b) \leq P_b} \lambda_p \geq 1 \quad (\pi_{ovr}^{brch}) \\
\forall u_0 \in \mathcal{U}, \quad & \sum_{p \in \mathcal{P}^{u_0}} \lambda_p = 1 \quad (\sigma)
\end{aligned} \tag{3.17}$$

3.3.3 Génération de colonne

Les sous-problèmes se décomposent unité par unité et s'écrivent pour tout u , $CR(u)$.

$$\begin{aligned}
CR(u) = \min_{x, y} \quad & f_u^{cout}(x, y) + f_u^{dual}(x) + f_u^{brch} \\
\forall t, i, \quad & x_{u,t}^{(i)} \leq x_{u,t+1}^{(i-1)} \\
\forall t, i, \quad & x_{u,t}^{(i)} \geq x_{u,t+1}^{(i+1)} \\
\forall t, \quad & \sum_{t'=t-L^u+1}^t y_{u,t'}^{(1)+} \leq x_{u,t}^{(1)} \\
\forall t, \quad & \sum_{t'=t-l^u+1}^t y_{u,t'}^{(1)+} \leq 1 - x_{u,t-l^u}^{(1)} \\
\forall t, i, \quad & \sum_{t'=t+1}^{t+\Delta_{u,i}^+} y_{u,t'}^{(i+1)+} + \sum_{t'=t+1}^{t+\Delta_{u,i}^-} y_{u,t'}^{(i-1)-} + x_{u,t}^{(i+1)} \leq x_{u,t}^{(i)} \\
\forall b \in \mathcal{N}_{und}^{brch}, \forall i, P(u_b, i) > P_b \quad & x_{u,t}^{(i)} \leq 0 \\
\forall b \in \mathcal{N}_{ovr}^{brch}, \forall i, P(u_b, i) \leq P_b \quad & x_{u,t}^{(i)} \geq 1 \\
\forall t, i, \quad & y_t^{(i)+}, y_t^{(i)-}, x_t^{(i)} \in \{0, 1\}
\end{aligned} \tag{3.18}$$

$$f_u^{dual}(x) = \sigma_u - \sum_{u,i,t} (\pi_t^P (P_{u,i} - P_{u,i-1}) + \pi_t^{R1} (R_{u,i}^1 - R_{u,i-1}^1) + \pi_t^{R2} (R_{u,i}^2 - R_{u,i-1}^2)) x_t^{(i)}$$

$$f_u^{brch}(x) = - \sum_{b \in \mathcal{N}_{und}^{brch}} \pi_{und}^{brch} x_{t_b}^{(i_b)} - \sum_{b \in \mathcal{N}_{ovr}^{brch}} \pi_{ovr}^{brch} x_{t_b}^{(i_b)}$$

$$f_u^{cout}(x, y) = \sum_t \left(C_u^{fix} x_t^{(1)} + C_u^{dem} y_t^{(1)+} + \sum_i C_u^{prp} (P_{u,i} - P_{u,i-1}) x_t^{(i)} \right)$$

Pour la résolution, de tels sous-problèmes, nous avons utilisé une résolution frontale par Branch&Bound. Une résolution par programmation dynamique aurait également été envisageable.

3.4 Résultats d'implémentation

Les implémentations ont été réalisées avec Cplex, à l'aide du modeleur OPL dans leur version 12.2. Les résultats présentés ici ont été calculés avec un processeur Intel Core2 Duo de 2.80GHz. Les formulations compactes suivantes ont été implémentées et comparées : La formulation **Etat0** est la formulation avec les variables d'états de la section 2.3.2. La formulation **Niv0** est la formulation avec les variables de niveaux de la section 2.3.2. La formulation **EtatRenf** est directement la formulation avec les variables d'états de

la section 3.1.3. La formulation **NivRenf** est directement la formulation avec les variables de niveaux de la section 3.1.3. La formulation **NivRenf2** est la formulation de la section 2.3.2 où les variables $y_{u,t}^{(i)-}$ ont été remplacées par leur expression linéaire des autres variables dans (8.3). La formulation **Mixte** comprend les variables de niveaux et de paliers, ainsi que toutes les variables $y_{u,t}^{(i)-}$ et $y_{u,t}^{(i)+}$. Les contraintes sont celles de 3.1.3 avec les variables de niveaux. Les contraintes (u, t) , $\sum_{i=1}^j s_{u,t}^{(i)} = l_{u,t}^{(j)}$ sont ajoutées pour relier les variables de niveaux aux variables d'état. Des contraintes sur les variables d'état sont ajoutées, non nécessaires à la description du problème, en vue de la génération de coupes dûes à la reconnaissance par Cplex de structures spécifiques. Il s'agit des contraintes de demandes 2.27, 2.28, 2.29, de flots (2.22) et les contraintes GUB (2.19).

3.4.1 Relaxation linéaire des formulations compactes

Le tableau 3.1 compare les qualités des relaxations continues des formulations compactes précédemment détaillées. Comme la théorie le prévoyait, les formulations **EtatRenf**, **NivRenf**, **NivRenf2** et **Mixte** ont des bornes duales de relaxation continue identiques, renseignées dans la colonne "Renf". Un gain très significatif a ainsi été obtenu sur les qualités de relaxation continue, ce qui aura de fortes implications sur les qualités de résolution exacte par Branch&Bound.

Instance	Valeur	Etat0	Niv0	Renf
Dat6	9 209 926	5,96%	3,56%	3,53%
Dat16	6 895 604	5,29%	4,50%	4,46%
Dat35	17 007 286	3,79%	1,43%	1,40%
Dat57	≤48 940 734	4,81%	2,68%	1,84%
Dat67	≤ 55 911 670	4,03%	2,00%	0,88%
D80a	≤51 087 946	4,94%	2,33%	1,77%
D80b	≤24 427 355	8,41%	2,27%	0,39%
D82a	≤124 194 907	5,74%	1,46%	0,72%
D82b	≤124 481 919	6,54%	1,71%	0,70%

TABLE 3.1 – Bornes duales de relaxation continue

3.4.2 Convergence Branch&Bound des formulations compactes

On a comparé la convergence des formulations sur les tableaux 3.2, 3.3 et 3.4. Les critères comparés sont la valeur du relâché continu, les écarts des bornes primales et duales à l'optimum avant les branchements, le nombre de lignes et de colonnes après les presolve de Cplex, ainsi que le nombre de nœuds nécessaires à la convergence et le temps de calcul.

La structure de convergence est assez globale pour ces formulations, suivant les convergences des figures 3.2 et 3.3. L'algorithme est très efficace sur les premiers stades, que ce soit dans le traitement du nœud racine ou aux premiers branchements. D'excellentes solutions sont trouvées rapidement, et cela est suivi par une longue phase où la borne primale est peu améliorée pour enfin prouver l'optimalité de la solution. En paramétrant Cplex pour favoriser la croissance des bornes duales (à l'aide du paramètre `mipemphasis`), on améliore l'efficacité de la formulation **NivRenf2** pour le calcul exact, ce qui n'est pas le cas pour **NivRenf**. Le grand nombre de branchements sur les grandes instances s'expliquent par la structure sous jacente de sac à dos entiers, et aux fortes symétries entre solutions. Pour une application industrielle, cette structure de convergence s'avère très utile pour obtenir de très bonnes solutions avec du calcul arrêté en PLNE.

Malgré le faible écart de relaxation continue entre les formulations, on observe de nettes différences en temps de calcul. L'avantage des variables de niveaux sur les variables de paliers se situe dans une

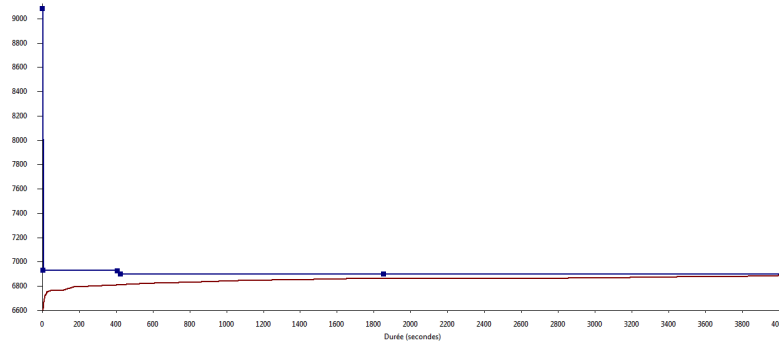


FIGURE 3.2 – Convergence Branch&Bound sur Dat16 avec la formulation EtatRenf

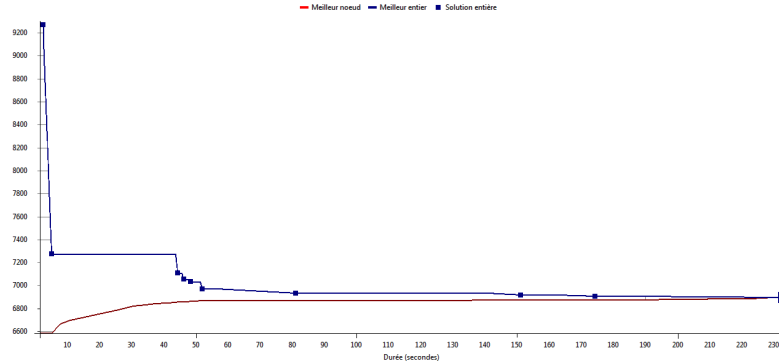


FIGURE 3.3 – Convergence Branch&Bound sur Dat16 avec la formulation Mixte

meilleure efficacité des branchements. Brancher sur une variable de niveau revient en effet à la séparation efficace des contraintes GUB sur les variables d'état. Comme OPL ne permettait pas de déclaration Special Ordered Sets, les branchements avec les variables d'états étaient connus peu efficaces car menant à des arbres de Branch&Bound déséquilibrés. Cependant, activer la séparation SOS (réalisable avec Concert) est incompatible avec la génération de coupes de couverture GUB, ce qui est pourtant utilisé abondamment avec les variables d'état et est très utile, ce qui n'est donc pas une solution d'implémentation adaptée. La formulation **EtatRenf** peut en effet surpasser la résolution par **NivRenf** (cf data16) malgré une mauvaise stratégie de branchements, grâce à des bornes duales de meilleure qualité au noeuds racine. Cela s'explique par la génération de coupes de couvertures GUB renforcées, activée par les contraintes (2.19), (2.27), (2.28) ou (2.29).

Sur la question de la suppression de variables par la relation (8.3), on observe que le ratio de $\frac{2}{3}$ sur le nombre de variables est conservé par le presolve de Cplex. Cplex ne se sert donc pas de la relation (8.3) pour supprimer des variables, comme pour **NivRenf2**. Sur les plus petites instances, il s'avère plus efficace de supprimer manuellement des variables, pour gérer des matrices de contraintes plus petites et accélérer les algorithmes matriciels tels que le simplexe ou les générations de coupes. Sur les instances où de nombreux branchements sont nécessaires, la tendance s'inverse. En effet, les variables supplémentaires **NivRenf** offrent plus de choix de branchements, pour une meilleure efficacité des branchements de Cplex. Cela s'explique par les règles de branchement implémentées dans Cplex. Garder toutes les variables permet de brancher sur les variables $y_{u,t}^{(i)-}$, alors que Cplex ne détecte pas que l'expression linéaire $y_{u,t}^{(i)+} + l_{u,t-1}^{(i)} - l_{u,t}^{(i)}$ est binaire, et donc que des branchements peuvent être réalisés sur de tels hyperplans.

Les constats précédents ont incité à expérimenter la formulation **Mixte**. Nous avons obtenu une convergence comme sur la figure 3.2, pouvant combiner les avantages de deux formulations de variables. Un fait paradoxal, malgré une formulation lourde en variables, cela pouvait permettre d'accélérer nettement les temps de calcul. Même des contraintes dominées polyédralement pouvaient avoir un apport non négligeable, ainsi rajouter les contraintes de flots (2.22), bien que dominées par les contraintes (2.34) et (2.35) présentes

Form.	Relax	Inf0	Sup0	Col.	Lign.	Noeuds	Temps
Etat0	5,96%	0,67%	1,29%	4.439	8.601	2385	366s
Niv0	3,56%	0,35%	0,21%	4.449	11.392	328	65 s
NivRenf2	3,53%	0,42%	0,12%	2.942	6.084	192	16 s
NivRenf	3,53%	0,31%	0,13%	4.426	6.101	307	19 s
EtatRenf	3,53%	0,56%	0,30%	4.435	6.869	1191	143 s
Mixte	3,53%	0,27%	0,30%	5.756	8.647	270	47 s

TABLE 3.2 – Caractéristiques de convergence Branch&Bound sur Dat6

Form.	Relax	Inf0	Sup0	Col.	Lign.	Noeuds	Temps
Etat0	5,29%	0,78%	5,40%	9.392	17.555	13.665	2692s
Niv0	4,50%	1,93%	0,37%	10.855	25.228	11.978	5306s
NivRenf2	4,46%	2,33%	0,50%	7.576	14.160	199.372	50099s
NivRenf	4,46%	1,83%	0,50%	11.165	14.984	12.810	1907 s
EtatRenf	4,46%	0,78%	3,30%	16.704	11.179	5.241	591 s
Mixte	4,46%	0,26%	0,89%	14.783	21.190	524	256 s

TABLE 3.3 – Caractéristiques de convergence Branch&Bound sur Dat16

Form.	Relax	Inf0	Sup0	Col.	Lign.	Noeuds	Temps
Etat0	3,79%	0,35%	0,02%	22.123	41.046	2008	488 s
Niv0	1,43%	0,57%	0,01%	24.066	58.955	865	580 s
NivRenf2	1,40%	0,71%	0,0%	16.509	31.071	850	381 s
NivRenf	1,40%	0,50%	0,0%	23.887	31.445	870	330 s
EtatRenf	1,40%	0,28%	0,02%	23.911	35.625	2434	271 s
Mixte	1,40%	0,53%	0,13%	31.445	44.345	753	533 s

TABLE 3.4 – Caractéristiques de convergence Branch&Bound sur Dat35

dans le modèle , permettaient de générer des coupes spécifiques liés à une structure de flot, pour une amélioration des bornes duales après les passes de génération de coupes au nœud racine.

Comme nous ne sommes pas sortis du cadre d'OPL, nous n'avons pas pu implémenter une meilleure formulation mixte, pour mieux combiner les avantages de différentes formulations, sans alourdir les calculs avec un nombre trop important de variables et de contraintes tout le long de la résolution. Avec une implémentation de plus bas niveau, avec Concert, l'implémentation que les constats précédents nous suggère serait de faire une génération de coupes au nœud racine avec la formulation *Mixte*, pour des coupes duales de meilleure qualité. L'idée est de récupérer ces coupes et finir la résolution avec la phase de branchements sur la formulation *NivRenf*, qui permet d'avoir les branchements et les calculs de borne duales à chaque nœud les plus efficaces. Pour ce faire, il faut projeter les coupes obtenues à la racine avec les variables s et x en coupes polyédralement équivalentes sur les variables x ce que permet la transformation de la proposition 3.2.2.

3.4.3 Résultats de la formulation étendue

La reformulation de Dantzig-Wolfe, n'a pas permis d'améliorer les bornes duales, avec une résolution facile des sous-problème par PLNE. Cela laisse présager que la description polyédrale des contraintes de fonctionnement thermique présentée a des points extrêmes entiers, étendant les résultats de [105]. Il n'y aurait donc dans ce cas aucun intérêt à considérer une résolution par Branch&Price du problème UCPd,

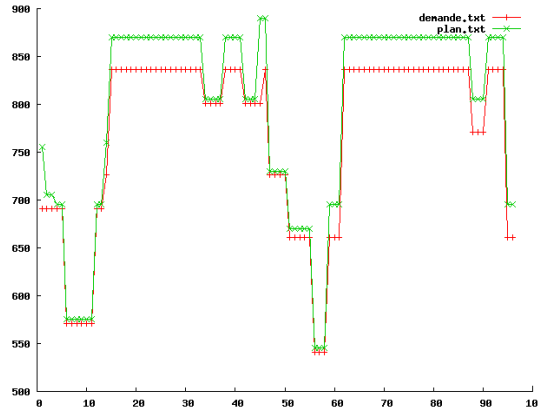


FIGURE 3.4 – Production globale/ Demande de solutions du problème UCPd

comme la description des contraintes des sous-problèmes présente de bonnes propriétés polyédrales. De plus, on observait une convergence très lente de l’algorithme de génération de colonnes, avec un dual très instable. On observait des vecteurs duaux avec très peu de valeurs non nulles, et un comportement très ératique, qui avait alors tendance à générer comme colonnes des plannings de production correspondant à un démarrage et un arrêt autour de ces pics de dualités, qui sont économiquement de très mauvaises solutions.

Les tentatives de stabilisation de la convergence avec l’ajout initial d’excellentes solutions primales (dont l’optimum entier) ne résolvait pas ce problème, les solutions finales de la relaxation continue comportaient ce type de solutions assemblés par combinaison linéaires. Cela est dû au fait que les plannings de production réalisables ne prennent que des valeurs discrètes avec de forts écarts, et que les contraintes de demandes par assemblage de blocs de puissances discrets sont des inégalités cruciales que sur quelques points, et les contraintes d’inertie de production liées à chaque unité rendent les contraintes de demandes moins prégnantes sur ces points par rapport des solutions entières s’avérait plus rentable, pour mieux satisfaire la contrainte de demande en continue et ne pas avoir une production coûteuse non demandée. Ce dernier point enlevait également les velléités de dériver la formulation en mathématique pour générer des plannings de solutions intéressants pour améliorer une bonne solution existante.

3.5 Conclusions et perspectives

Dans ce chapitre, plusieurs formulations compactes en PLNE ont permis de modéliser le problème UCPd. Des considérations polyédrales et de coupes de clique ont permis d’améliorer et de comparer les formulations. Cela a impliqué des progrès conséquents sur la résolution en nombres entiers.

Si à l’issue du travail polyédral, trois formulations étaient équivalentes en terme de description polyédrale et ainsi de relaxation continue, la détection de structure spécifique par Cplex s’avérait cruciale dans la résolution en nombres entiers. On a pu exhiber l’implémentation adéquate pour tirer au mieux parti des différentes structures spécifiques : une génération de coupes au nœud racine de Branch&Bound selon la formulation *Mixte*, pour projeter ces coupes dans l’espace des variables de niveaux uniquement, et terminer la résolution avec la phase de branchements selon la formulation *NivRenf*.

La reformulation de Dantzig-Wolfe, n’a pas permis d’améliorer les bornes duales, avec une résolution facile des sous-problème par PLNE. Cela laisse présager que la description polyédrale des contraintes de fonctionnement thermique présentée a des points extrêmes entiers, étendant les résultats de [105]. Une perspective serait de le démontrer, suivant le schéma de preuve de [105]. Il n’y a donc aucun intérêt à considérer une résolution par Branch&Price du problème UCPd.

Une autre perspective de ce travail est d'accélérer la convergence des bornes primales dans un but opérationnel. Le temps de calcul est principalement dédié à la preuve d'optimalité, pour une application industrielle, seules les bornes primales comptent. Le chapitre suivant suit ces perspectives, en dérivant des heuristiques du travail précédent.

Chapitre 4

Matheuristiques dérivées pour UCPd

Résumé : Ce chapitre dérive du travail de résolution PLNE du chapitre 3 des matheuristiques pour la résolution de problèmes UCPd en temps contraints. On élabore des heuristiques constructives efficaces, par fixation judicieuse de variables ou suivant des processus hiérarchiques de fixation de variables. Les solutions construites sont améliorées par recherche locale, avec un algorithme de VND où de multiples voisinages sont définis en réoptimisant autour de la meilleure solution courante. De telles approches ont amélioré significativement les solutions obtenues en temps contraints.

Le travail du chapitre précédent a fourni une formulation PLNE pour la résolution efficace de problèmes UCPd. En pratique, les temps de calculs pour la résolution opérationnelle sont très limités, 15 minutes pour la résolution du problème de production journalière à EDF pour le parc complet. Même sur la restriction du parc aux unités thermiques, de tels temps de calculs ne permettent pas la convergence Branch&Bound à l'optimalité. La problématique est alors de trouver les meilleures solutions primales en temps restreint. L'approche exacte par Branch&Bound permet d'obtenir des solutions primales en tronquant le temps de résolution. Ce chapitre étudie la possibilité d'améliorer la convergence des bornes primales de l'approche exacte, grâce à l'intuition et la connaissance que l'on peut avoir du problème. On se basera sur le travail de modélisation PLNE pour en dériver des matheuristiques efficaces. Une première partie est consacrée aux heuristiques constructives basées sur la formulation exacte du chapitre précédent, tandis que la seconde partie cherchera à améliorer rapidement des solutions existantes en adaptant l'algorithme VNS à des voisinages définis par des fixations de variables dans le PLNE. Dans tout ce chapitre, on utilisera la formulation d'UCPd avec les variables de paliers $x_{u,t}^{(i)}$, pour ses propriétés aux branchements, intéressantes lorsqu'il s'agira de fixer des variables.

4.1 Heuristiques constructives

Dans cette section, on cherche à construire rapidement des bonnes solutions du le problème d'UCPd, en utilisant le travail de formulation PLNE du chapitre 3. Des PLNE plus petits sont calculés en fixant des variables. L'accélération escomptée de la convergence des bornes primales doit alors contrebalancer la dégradation de la fonction objectif pour une résolution efficace en temps contraint.

4.1.1 Heuristiques par fixation de variables

Une première idée d'heuristique constructive est de résoudre frontalement un plus petit PLNE après avoir fixé des variables, en espérant que les décisions figées correspondent à de bonnes solutions. Pour cela, l'intuition et la connaissance du problème peuvent donner une idée de variables à fixer. L'excellente qualité de la relaxation continue d'UCPd précédemment mise en évidence peut être utilisée, en espérant que la solution continue est proche de bonnes solutions entières et ou contient des informations caractérisant

de bonnes solutions entières. On notera $\tilde{x}_{u,t}^{(i)}$ la solution de la relaxation continue. Fixer toutes les variables entières dans la relaxation continue permettait d'avoir une résolution très rapide, mais les solutions s'avéraient assez mauvaises. On a ainsi implémenté des stratégies plus flexibles. Les stratégies les plus efficaces étaient les suivantes :

- stratégie **hSelect** : on identifie les centrales u à production systématiquement nulles dans le relâché continu, ie telles que pour tout t , $\tilde{x}_{u,t}^{(i)} = 0$ et on les retire du PLNE final. Intuitivement, ces centrales ont des coûts marginaux de production prohibitifs, et ne sont pas activées au vue de la demande, des centrales économiquement plus rentables étant préférées.
- stratégie **hTube** : on supprime d'abord des centrales selon **hSelect**. Pour les autres centrales, on fixe les variables autour d'un voisinage en tube de la trajectoire de la solution continue : si $\tilde{x}_{u,t}^{(i+1)} = \tilde{x}_{u,t-1}^{(i)} = \tilde{x}_{u,t}^{(i)} = \tilde{x}_{u,t+1}^{(i)} = 1$ (resp 0), on utilisera la fixation de variable $x_{u,t}^{(i)} = 1$ (resp 0).
- stratégie **hTube3** : on considère trois scénarios de demandes : la demande réelle, une demande majorée et une demande minorée (ce que l'on a implémenté à l'aide d'un facteur multiplicatif). On calcule le relâché continu obtenu pour ces trois scénarios, et on fixe les variables communes de la stratégie **hTube** sur ces trois scénarios.

4.1.2 Approches de construction hiérarchique

On peut décomposer les décisions par étapes hiérarchiques, par résolution séquentielle de PLNE moins combinatoires. Les calculs successifs sont arrêtés suivant des critères d'arrêts (temps, écarts relatifs) à paramétrer. L'algorithme de résolution Branch&Bound étant exponentiel, résoudre deux problèmes comprenant 50% des variables est plus rapide que résoudre le problème complet, ce qui justifie la méthode.

Pour résoudre un PLNE, on opère une partition des variables, considérant un ensemble de variables binaires sur lequel on figera la décision pour la suite, tandis que les autres variables calculées ultérieurement seront relâchées continûment, en espérant que cette simplification ne dégrade ni la qualité des solutions, ni la faisabilité. Après fixation des variables, Il reste alors un PLNE restreint à traiter, sur lequel on itère le processus, ou on opère un calcul final. Les stratégies testées rentrant dans ce cadre sont les suivantes :

- La stratégie **dcpJour** fixe d'abord le planning sur la première journée, en estimant l'impact des décisions prises avec les variables continues de la seconde journée. avant de résoudre le PLNE sur la seconde journée en fixant la première journée. Une telle séparation du PLNE divise la combinatoire par deux.
- La stratégie **dcpPal** procède paliers par paliers, calculant à chaque itération i' , les variables $x_{u,t}^{(i')}$ une fois que les variables $x_{u,t}^{(i)}$ pour $i < i'$ ont été fixées, les variables correspondant à $i > i'$ étant alors relâchées. Le premier problème fixe les les décisions de fonctionnement, id les variables $x_{u,t}^{(1)}$. Les derniers problèmes se retrouvaient alors avec une combinatoire très amoindrie du fait des contraintes dynamiques, qui imposaient alors peu de flexibilité.
- La stratégie **dcpMid** procède également par partition des variables sur l'indice i , mais en figeant initialement le palier du milieu, ie pour une unité u avec $N_{fct}(u)$ points de fonctionnement, on considère le palier $Mid(u) = \lfloor \frac{N_{fct}(u)}{2} \rfloor$. Une variable fixée à 0 (resp 1) implique que les variables supérieures (resp inférieures) sont aussi fixées et entières, ce qui induit une approche de fixation plus équilibrée que **dcpPal**.

Pour assouplir les décisions prises à chaque étape et gagner en qualité de solutions finales (ou ne pas mettre en péril la faisabilité finale), on peut s'autoriser de petites modifications par rapport aux décisions précédemment prises. Dans le cas de la décomposition temporelle **dcpJour**, cela pourra se traduire par un premier calcul avec des variables binaires sur les pas de temps $\llbracket 1, 54 \rrbracket$, et une fixation des variables sur $\llbracket 1, 42 \rrbracket$. Dans le cas des décompositions par paliers, on pourra fixer les variables $x_{u,t}^{(i)}$ que l'on notera $1-$

stable, telles que $x_{u,t}^{(i)} = x_{u,t-1}^{(i)} = x_{u,t+1}^{(i)}$.

4.1.3 Approche constructive parallélisable de type POPMUSIC

Les stratégies précédentes peuvent être cumulées. La stratégie **hMixte** suivante cumule les stratégies **hSelect**, **dcpJour** et **dcpMid**, pour permettre de gérer des sous-problèmes encore plus petits. Cela donne l'algorithme suivant où les fixations sont hiérarchisées, chaque étape comprenant des calculs indépendants qui peuvent être parallélisés. Un soin particulier a été pris aux interfaces des fixations.

Etape 1 : Après calcul de la relaxation continue, on supprime des unités à l'aide de la stratégie de fixation de variables **hSelect**.

Etape 2 : Deux PLNE indépendants sont calculés, première étape de **dcpMid** sur les deux journées :

1. PLNE 2.1 : Les variables binaires sont les $x_{u,t}^{Mid(u)}$ pour $t \in \llbracket 1, 48 \rrbracket$.
2. PLNE 2.2 : Les variables binaires sont les $x_{u,t}^{Mid(u)}$ pour $t \in \llbracket 49, 96 \rrbracket$.

Etape 3 : Trois PLNE indépendants sont calculés, calculant des plannings complets sur les fenêtres temporelles $t \in \llbracket 1, 32 \rrbracket$ et $t \in \llbracket 65, 96 \rrbracket$, et fusionnant les solutions de PLNE 2.1 et PLNE 2.2 :

1. PLNE 3.1 : Seules les variables $x_{u,t}^{(i)}$ sont binaires pour $t \in \llbracket 1, 32 \rrbracket$, en ayant fixé les solutions "1-stable" du PLNE 2.1 à leur valeur.
2. PLNE 3.2 : Seules les variables $x_{u,t}^{(i)}$ sont binaires pour $t \in \llbracket 65, 96 \rrbracket$, en ayant fixé les solutions "1-stable" du PLNE 2.2 à leur valeur.
3. PLNE 3.3 : Toutes les variables $x_{u,t}^{Mid(u)}$ sont binaires, en ayant fixé les solutions de PLNE 2.1 (resp PLNE 2.2) pour $t \in \llbracket 1, 36 \rrbracket$ (resp $t \in \llbracket 65, 96 \rrbracket$).

Etape 4 : On calcule un PLNE final avec les fixations suivantes :

- Les variables $x_{u,t}^{(i)}$ pour $t \in \llbracket 1, 24 \rrbracket$ (resp $\llbracket 79, 96 \rrbracket$) sont fixées à leur valeur dans PLNE 3.1 (resp 3.2).
- Les variables $x_{u,t}^{(Mid(u))}$ for $t \in \llbracket 25, 32 \rrbracket$ (resp $\llbracket 65, 78 \rrbracket$) sont fixées si leur valeur est 1-stable et commune dans PLNE 3.1 (resp 3.2) et PLNE 3.3.
- Les variables $x_{u,t}^{(Mid(u))}$ pour $t \in \llbracket 33, 64 \rrbracket$ 1-stable dans PLNE 3.3.

4.1.4 Résultats

Les résultats des heuristiques illustrent le compromis à trouver entre temps de calcul et qualité de la solution. La figure 4.3 compare les convergences des bornes primales lors de l'algorithme de Branch& Bound pour la formulation exacte avec les fixations de variables **hSelect**, **hTube** et **hTube3** sur 15 minutes. Sur de tels temps de calculs, les approches de fixation convergent vers de meilleures solutions. Pour expliquer cela, il s'avère que les fixations de variables dégradent peu la qualité des solutions, et ont des valeurs de convergence proches de l'optimum. L'accélération engendrée par la diminution de la combinatoire permet ainsi d'obtenir en temps contraint de meilleures solutions que la résolution exacte.

hSelect permettait en moyenne d'éliminer 50% des variables sur nos instances, pour une approximation quasi nulle. Pour **hTube3**, cela atteignait 70%, et 80% pour **hTube**, pour une très faible approximation. Si **hTube** est la plus grande approximation, pour des temps arrêtés de 15 minutes qui restent courts, c'est l'approche qui donne la meilleure valeur arrêtée alors que la convergence se fait vers une valeur plus élevée. Pour un temps de calcul très longs (plusieurs journées) en vue d'établir d'excellente bornes primales, l'approche exacte était fréquemment surpassée par les approches heuristiques aboutissant à une convergence plus avancée de l'algorithme de Branch&Bound. La stratégie **hTube3** s'avère être un excellent compromis.

Les heuristiques hiérarchiques dégradent plus la fonction objectif que les approches frontales, mais convergent bien plus vite. Les tableaux 4.2 comparent les valeurs obtenues par les différentes heuristiques

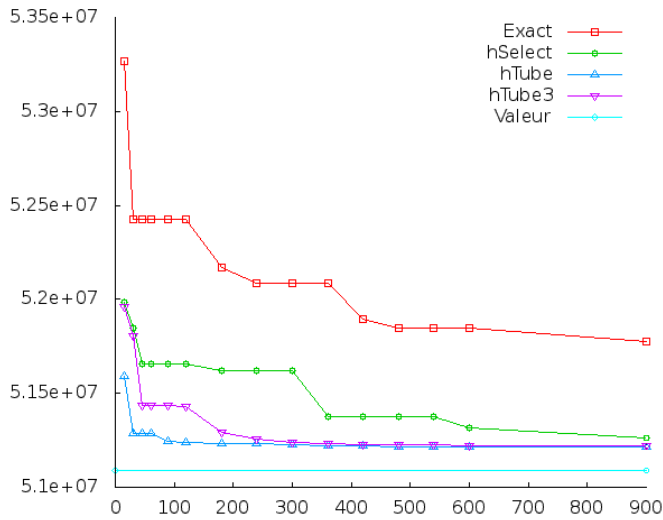


FIGURE 4.1 – Sur l’instance Data80Spring

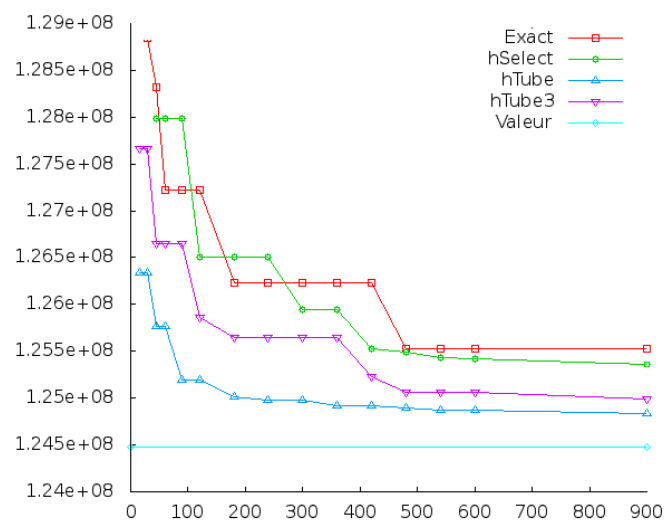


FIGURE 4.2 – Sur l’instance Data80WinterWE

FIGURE 4.3 – Comparaison de la convergence des heuristiques avec la résolution exacte

aux solutions fournies par l’approche exacte tronquée en se donnant des temps de calculs de 1 et 5 minutes. `dcpJour` pouvait amener à des infaisabilités ou des coûts prohibitifs, `dcpMid` et `hMixte` s’avéraient très compétitives en moyenne, `dcpPal` fournissant des résultats de qualité variables. En moyenne, `hMixte` et `hTube` donnent les meilleurs résultats, mais n’étaient pas systématiquement les meilleures approches qui donnaient les meilleurs résultats. Avoir différentes stratégies heuristiques sans hiérarchie nette dans les résultats laisserait à penser que pour une utilisation opérationnelle, considérer indépendamment et parallèlement ces différentes stratégies permettrait d’obtenir des gains conséquents.

Pour `hMixte`, les valeurs à 5 minutes étaient le plus souvent les valeurs de convergence de la méthode. Une troncature de la convergence des sous-problèmes a peu d’incidences. Cela permettrait de traiter des tailles de données encore plus grandes, là où les stratégies de résolution frontale après fixations pourraient avoir des convergences moins bonnes.

4.2 Amélioration de solutions par descente VNS

Dans cette section, on cherchera à améliorer une solution initiale par une recherche locale, avec un algorithme de VND où de multiples voisinages sont définis par fixation de variables sur des PLNE, en réoptimisant autour de la meilleure solution courante. Notre algorithme s’est inspiré du travail effectué dans [84], faisant un pont entre la PLNE et les heuristiques de recherche locale VNS. Dans le cas d’UCPd, la diversité des voisinages, et le travail de formulation et la réflexion sur les fixations de variables rendait l’application de la méthode particulièrement intéressante.

4.2.1 L’algorithme général de descente VNS par PLNE

La méthode nécessite une solution initiale, que les approches précédentes peuvent fournir. Il s’agit de procéder par itérations de recherche locale, où à chaque itération, on réoptimise des morceaux de la solution courante par des petits PLNE, en fixant des variables de la solution courante. La solution courante est solution d’un tel PLNE, et peut être fournie comme démarrage à chaud. Une telle méthode est une méthode de descente pure, sans remise en cause de la meilleure solution, variante VND de VNS. L’algorithme progresse toujours avec des solutions réalisables, définies par les contraintes en PLNE, et dont les coûts sont donnés par la résolution PLNE. Le critère d’arrêt est un nombre d’itérations maximal, ainsi

Instance	Valeur	Exact	hSelect	hTube	hTube3	dcpPal	dcpMid	sMixte
Data57	48 940 734	3,18%	1,76%	1,73%	1,64%	1,60%	1,84%	1,86%
Data67	55 911 670	2,24%	1,81%	1,59%	1,60%	1,87%	1,99%	1,97%
Data80Spring	51 087 946	1,20%	0,57%	0,25%	0,28%	0,16%	0,42%	0,42%
Data80SprgWE	24 427 355	0,70%	0,66%	0,69%	0,52%	0,78%	0,93%	0,97%
Data80WtrWE	124 481 919	1,39%	0,76%	0,33%	0,27%	3,05%	0,64%	0,21%
Data80Winter	124 194 907	0,89%	0,87%	0,16%	0,89%	0,20%	0,38%	0,16%
Total	429 044 531	1,50%	1,01%	0,62%	0,79%	1,43%	0,87%	0,68%

TABLE 4.1 – Comparaison des solutions obtenues en temps restreint à 5 minutes

Instance	Valeur	Exact	hSelect	hTube	hTube3	dcpPal	dcpMid	sMixte
Data57	48 940 734	4,74%	3,01%	1,76%	2,02%	1,55%	1,86%	1,86%
Data67	55 911 670	4,22%	3,20%	1,59%	2,16%	2,08%	1,79%	1,97%
Data80Spring	51 087 946	1,11%	0,38%	0,68%	0,28%	0,32%	0,38%	0,42%
Data80SpgWE	24 427 355	1,70%	0,77%	0,89%	0,77%	0,81%	1,07%	1,05%
Data80WtrWE	124 481 919	2,82%	1,03%	1,75%	3,55%	0,53%	1,07%	0,97%
Data80Winter	124 194 907	1,13%	0,35%	1,39%	0,36%	0,45%	0,54%	0,23%
Total	429 044 531	2,67%	2,13%	0,90%	1,55%	1,65%	1,02%	0,93%

TABLE 4.2 – Comparaison des solutions obtenues en temps restreint à 1 minute

TABLE 4.3 – Comparaison des heuristiques primales constructives

qu'un nombre d'itérations sans améliorations.

A chaque itération, les voisinages sont définis et paramétrés par :

- Un ensemble de variables de la solution courante à fixer.
- Un critère d'arrêt à la résolution PLNE, considérant temps de calcul ou branchements limités, ...
- Une paramétrisation de la résolution Branch&Bound.

Il s'agit de paramétrer le solveur de PLNE pour tirer profit de ses forces. Nous utilisons la résolution PLNE à ses premiers stades, où la convergence est la plus efficace. La paramétrisation privilégie alors fortement les heuristiques primales au détriment de la convergence des bornes duales. Le paramètre générique `mipEmphasis` est utilisé, et peut être combiné à une limitation des générations de coupes à la racine, ou à une augmentation de la fréquence d'appel des heuristiques primales.

Cplex est adapté pour une telle utilisation, le preprocessing et l'agrégateur permettent de propager les fixations de variables pour se ramener à des petits PLNE. Le fait d'avoir une solution initiale à tout calcul de PLNE permet un démarrage à chaud efficace, activant dès le début les heuristiques perturbatives de type RINS, là où la résolution PLNE est souvent handicapée dans la recherche de premières solutions.

Par une telle recherche locale, de grands voisinages peuvent être explorés sur des temps de calculs faibles, en se basant sur les forces de la résolution PLNE, le travail de formulation compacte, et la connaissance du problème du problème par les fixations de variables adéquates. En considérant de grands voisinages, on a alors peu de minimums locaux. La facilité et la généricité d'implémentation permet de considérer un grand nombre de voisinages, rentrant dans le cadre des heuristiques VNS : on compte sortir d'un extremum local pour un type de voisinages en changeant de type de voisinage. Les sections suivantes fourniront des exemples de voisinages spécifiques au problème UCPd. Des voisinages génériques peuvent être envisagés. Similairement aux heuristiques primales Local Branching [54], on peut envisager des voisinages variables de la solution courante où k variables au plus peuvent être modifiées. Si on dispose d'un calcul de relaxation continue, on peut définir un voisinage RINS suivant la fixation de [46] : les variables communes dans la

solution courante et dans une relaxation continue sont fixées.

La méthode est naturellement parallélisable, en considérant à chaque itération plusieurs types de voisinages différents sur une même limite de temps de calcul, et en choisissant la meilleure solution obtenue par les différents calculs parallèles.

4.2.2 Définition des voisinages pour UCPd

Voisinages basés sur la fixation d'un indice

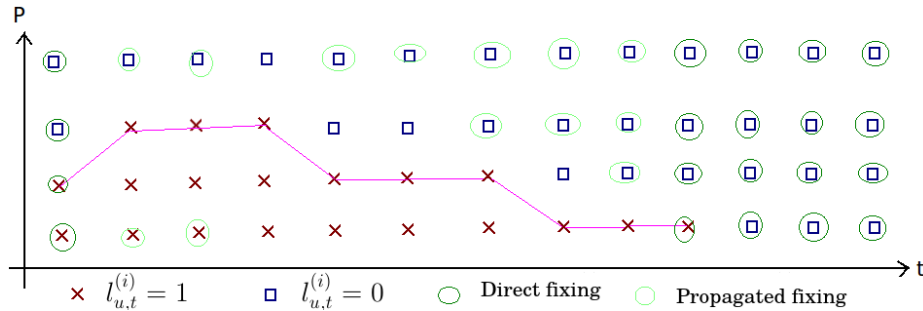


FIGURE 4.4 – Fixation autour de fenêtres de temps

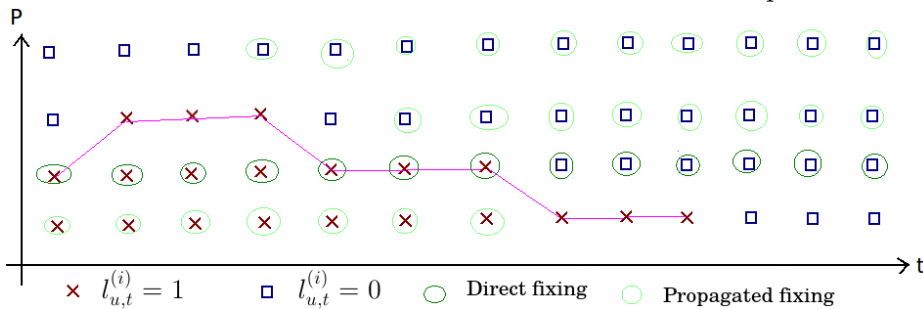


FIGURE 4.5 – Fixation du palier intermédiaire

FIGURE 4.6 – Illustrations de voisinages par fixation d'indices

Une idée naturelle lorsque l'on manipule des variables à plusieurs indices est de fixer les décisions prises sur un indice. On retrouve alors des fixations et des réoptimisations naturelles :

- Sur l'indice u : retirer les unités à la production systématiquement nulle (similaire à `hSelect`).
- Sur l'indice i , figer les variables d'un point de fonctionnement donné. Pour $i = 1$, cela fige les décisions d'arrêts et de démarrages. Sur un palier intermédiaire, comme il a été remarqué précédemment, cela induit un branchement équilibré.
- Sur l'indice t , on peut figer les décisions en dehors d'une fenêtre de temps. Les périodes de pics de demande sont bien connues par exemple, il est intéressant de chercher juste à réoptimiser.

La figure 4.6 illustre de telles fixations, et l'impact en terme de variables fixées par propagation des contraintes dynamiques.

Voisinages basés sur la structure dynamique du problème

Le problème UCPd a une structure dynamique spécifique, liée aux contraintes d'inertie sur les points de fonctionnement. Suivant les indices i et t , il y a peu de changements de valeur. On peut alors définir un voisinage en tube autour de la solution courante, où les décisions sont stables :

- Voisinage par stabilité temporelle : On fixe une variable $x_{u,t}^{(i)}$ si dans la solution courante $x_{u,t}^{(i)} = x_{u,t-1}^{(i)} = x_{u,t+1}^{(i)}$.

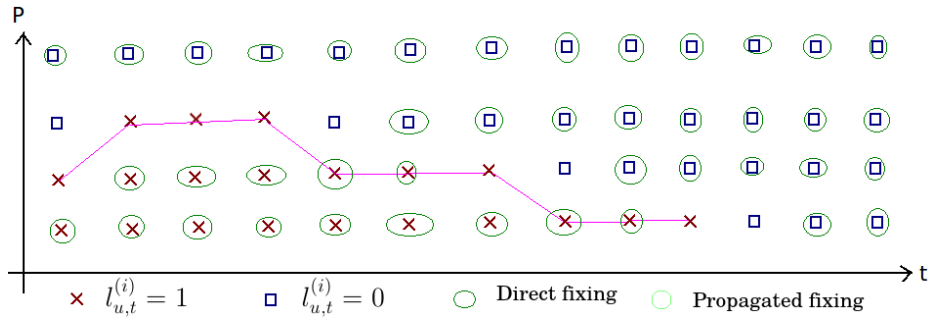


FIGURE 4.7 – Fixation temporelle autour des changements de paliers

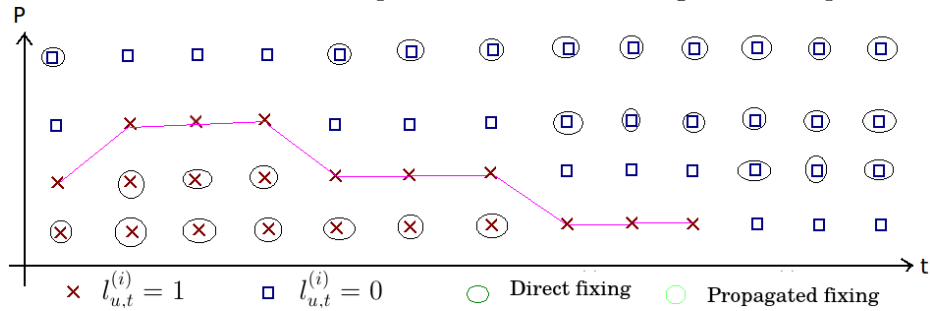


FIGURE 4.8 – Fixation similaire à hTube

FIGURE 4.9 – Illustrations de voisinages utilisant la structure dynamique du problème

- Voisinage par stabilité de niveau de puissance : On fixe une variable $x_{u,t}^{(i)}$ si dans la solution courante $x_{u,t}^{(i)} = x_{u,t}^{(i-1)} = x_{u,t}^{(i+1)}$. Cela revient exactement à la fixation de hTube sur la solution courante.

On remarque que la stabilité des niveaux de puissance implique la stabilité temporelle.

Voisinages mixtes

Pour assouplir les décisions de fixation, on peut combiner les deux types de voisinages précédents. La figure 4.10 illustre cela, en fixant les variables communes à la fixation sur le palier intermédiaire, en assouplissant avec la stabilité temporelle.

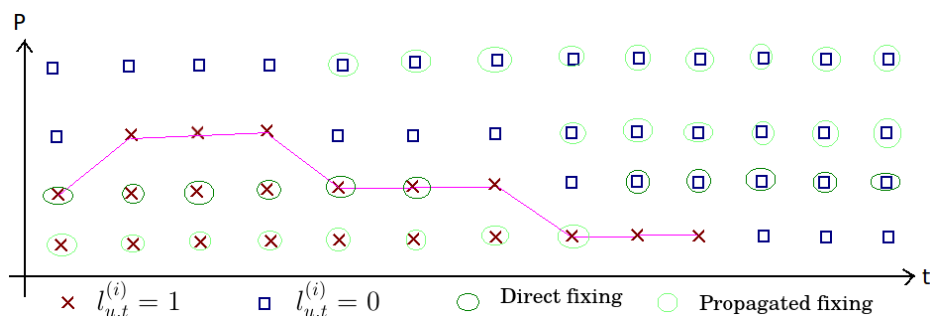


FIGURE 4.10 – Fixation assouplie du palier intermédiaire

Voisinages basés sur la relaxation continue

Un autre critère de fixation de variables se base sur une solution de la relaxation continue du problème, calculée séparément et sans fixation de variables. D'autres solutions relâchées peuvent également être considérées, comme la relaxation continue avec les coupes du noeuds racine de l'algorithme de Branch&Bound.

Les stratégies de fixations de variables de la section 4.1.1 restent toujours valables. On peut ainsi utiliser les fixations hSelect et hTube. Ces dernières fixations se basent alors uniquement sur la solution continue,

sans utiliser la solution courante autrement que dans le démarrage à chaud du PLNE.

En recoupant des informations issues de la solution courante et de la relaxation continue, on obtient un voisinage ν RINS similairement à l'heuristique RINS, où les variables fixées sont les variables entières communes dans la solution courante et dans la relaxation continue. Cela donne alors un voisinage très différents des voisinages classiques, car non intuitif.

4.2.3 Résultats

L'implémentation a été réalisée à partir du modèle compact en OPL, en appelant les différents PLNE à l'aide d'OPL script. L'implémentation était séquentielle, pour valider uniquement les propriétés mathématiques des différents voisinages et leur impact sur la convergence. Le temps de résolution maximal pour les grands voisinages est de 30s, contre moins de 10s pour les voisinages où la résolution était la plus facile. L'ordre de sélection des voisinages implémenté est déterministe, basé sur la connaissance empirique et l'observation des différents types de voisinages.

Dans un premier temps, il est efficace de privilégier les plus petits voisinages pour une convergence rapide, et de n'utiliser les grands voisinages que dans un second temps pour affiner les solutions. La diversité des types de voisinages doit être privilégiée pour ne pas rester dans un minimum local. Dans cette optique, la fixation RINS fournissait des voisinages très complémentaires des autres fixations.

Les voisinages de la section 4.2.2, permettaient de sortir d'excellents minimums locaux pour affiner de bonnes solutions, au prix de temps de calculs plus importants. Lorsque peu de grands voisinages améliorent la solution courante au prix de lourds temps de calculs, une parallélisation serait un atout dans l'implémentation. En partant d'une mauvaise solution obtenue rapidement par la formulation frontale, l'approche est tout à fait compétitive dans un temps de calcul de 5 minutes, comme le montre le tableau 4.11. Dans certains cas, la descente VNS permet d'obtenir des solutions qui améliorent même les meilleures valeurs obtenues après des temps de calculs très longs (plusieurs journées avec les heuristiques de la section 4.1.1). De tels faits montrent que les solutions obtenues par un calcul tronqué sur des temps de calculs courts suivant les heuristiques aveugles d'un solveur de PLNE peuvent être améliorées par des modifications locales basées sur la connaissance du problème.

Instance	Valeur	Frontal	hSelect	hTube	VNS
dataCharbon	16445498	0,93%	0,44%	0,21%	-0,02%
dataFuel	55911670	2,24%	1,81%	1,59%	1,57%
data48	34767116	3,23%	1,64%	1,87%	1,56%
data57	48940734	3,18%	1,76%	1,73%	1,75%
dataSpring	51087946	1,2%	0,57%	0,25	0,24%
dataSpringWE	24427355	0,7%	0,66%	0,69	0,76%
dataWinterWE	124481919	1,39%	0,76%	0,33%	0,29%
dataWinter	124194907	0,89%	0,87%	0,16%	0,19%
datSpring	17407479	0,42%	0,31%	0,06%	0,03%
datWinterWE	38837637	1,02%	1,31%	0,65%	0,21%
datWinter	34852129	0,5%	0,49%	0,24%	-0,06%
Total	520141776	1,36%	0,98%	0,58%	0,52%

FIGURE 4.11 – Comparaison des matheuristiques sur UCPd en temps de résolution de 5 minutes

4.3 Synthèse des résultats et perspectives

La question initiale était d'investiguer si la connaissance et l'intuition du problème spécifique permet d'améliorer la convergence des bornes primales par Branch&Bound, approche générique, sur des grands problèmes où l'approche exacte ne peut converger entièrement. Nous avons répondu par l'affirmative à cette question par trois types approches.

Fixer des variables selon l'intuition et la connaissance du problème pour une résolution frontale ultérieure permet d'accélérer la convergence des bornes primales en temps limité lorsque les fixations sont judicieuses et ne dégradent que très peu la fonction objectif. Dans le cas présent, les fixations `hSelect` et `hTube` sont intéressantes. Sur des temps très courts comme exigé par les contraintes opérationnelles, décomposer heuristiquement le PLNE en sous-problèmes avec une résolution hiérarchique engendre de plus fortes approximations. Les bornes primales de convergence étant alors plus élevée, mais la convergence y est plus rapide que les approches de résolution frontale. De telles approches permettraient de traiter de plus grandes tailles de données avec moins de limitations que les approches de fixation et de résolution frontale précédentes.

Une fois une première solution construite, une recherche locale de descente VND par itérations de PLNE successifs présentait de nombreux avantages : implémentation facile et générique à partir d'une formulation PLNE, où les questions de coûts et de faisabilité sont garantis par la PLNE, avec de nombreuses manières de définir des voisinages de grande taille pour avoir des minimas locaux de très bonne qualité. Les VND ont pu surpasser les autres heuristiques en contexte de temps de calcul restreint, mais aussi en temps plus long.

De tels résultats laissent à penser que ces approches pourraient être utilisées sur des problèmes plus grands ou plus difficiles, où la résolution frontale serait moins performante. Le problème UCPd étant un problème de travail, de telles idées pourraient être reprises pour le problème de production journalier d'EDF, bien plus difficile avec un parc de production plus important. Sur la suite de la thèse, nous nous sommes consacrés au problème de planification des arrêts de maintenances des centrales nucléaires du challenge ROADEF, en fournissant des réponses à ces perspectives méthodologiques d'utilisation de matheuristiques.

Troisième partie

Planification des arrêts nucléaires pour maintenances et rechargements

Chapitre 5

Formulations PLNE compactes pour le challenge ROADEF

Résumé : Ce chapitre pose les modélisations PLNE adaptées pour le problème de planification des maintenances et des rechargements des centrales nucléaires. Par rapport au challenge ROADEF, toutes les contraintes autres que CT6 et CT12 ont été modélisées exactement, avec une formulation PLNE où les seules variables discrètes concernent les dates d'arrêt. Pour les contraintes de stretch, des contraintes moins fortes ont été proposées. En outre, l'extension du modèle du challenge ROADEF pour considérer un planning initial, un nombre maximal de modifications et/ou une pénalisation de la déviation au planning initial. Un effort a été réalisé pour réduire le nombre de variables continues dans le modèle. Ce travail a fait l'objet d'une communication au congrès PGMO 2013.

Le problème d'application est dans la suite le problème de planification des arrêts de maintenances et rechargement des centrales nucléaires, défini à la section 2.4, dont les travaux préparatoires et l'état de l'art majeurs de cette thèse sont issus du Challenge EURO/ROADEF 2010. Ce chapitre fournit la modélisation PLNE compacte utilisée dans la suite. En omettant les contraintes CT6 et CT12 du Challenge ROADEF (stretch et modulation), on a une modélisation compacte en PLNE exacte. Les contraintes de modulations sont omises dans toute la suite. En ce qui concerne le stretch, on assouplit le domaine de production, en autorisant des modulations pendant le stretch. La justification de cette hypothèse est que pour des raisons économiques, il est souvent plus rentable d'avoir une production nucléaire maximale, et les contraintes impactantes sont les contraintes majorant la production. Avec ce domaine de production, on pourra aboutir à une formulation en programmation linéaire mixte où les seules variables discrètes sont les variables décisionnelles du problème, le placement des dates d'arrêt. Par rapport à la modélisation du challenge ROADEF, on permettra des extensions, avec la possibilité d'ajouter et de paramétrer des contraintes de longueur minimale et maximale d'une cycle de production ou une pénalisation par rapport à un planning initial.

5.1 Définition des variables et de la fonction objectif

Cette section définit les variables utilisées dans la modélisation PLNE, et écrit la fonction objectif avec ces variables. On réalisera une première extension du challenge ROADEF en considérant des coûts de stabilité de planning, en pénalisant les écarts à un planning initial, la fonction objectif du challenge ROADEF se retrouvant alors avec des pénalisation nulles à un planning initial.

5.1.1 Définition des variables

Variables de décisions d'arrêt Les variables de décisions d'arrêts sont les seules variables discrètes du problème. $d_{i,k,w} \in \{0, 1\}$ est une famille de variables binaires pour indiquer les dates d'arrêts, $d_{i,k,w} = 1$ si l'arrêt du cycle k du réacteur nucléaire i a été commencé avant la semaine w . Cette définition de variables implique les contraintes couplantes suivantes :

$$\forall i, k, w, \quad d_{i,k,w-1} \leq d_{i,k,w} \quad (5.1)$$

$$\forall i, k, w, \quad d_{i,k,w} \leq d_{i,k-1,w-1} \quad (5.2)$$

On utilisera les conventions suivantes pour écrire plus simplement certaines contraintes, $d_{i,k,w} = 0$ pour $k > K_i, d_{i,-1,w} = 1$ pour $w < 0$, $d_{i,k,w} = 0$ pour $w < 0$ et $k > -1$.

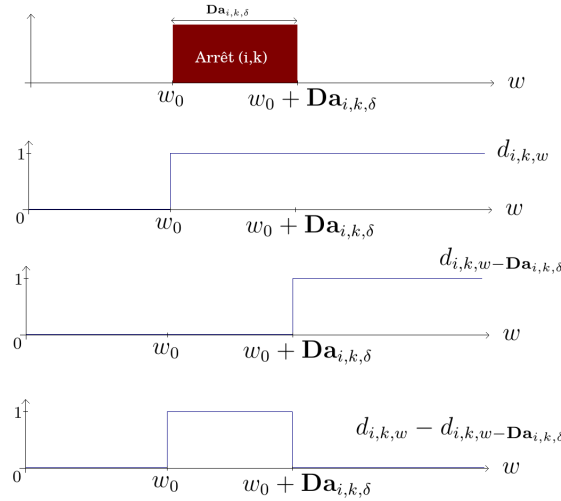


FIGURE 5.1 – Définition des variables d'arrêt

La définition des variables binaires est illustrée à la figure 5.1. On peut exprimer comme combinaison linéaire de variables $d_{i,k,w}$ les grandeurs suivantes :

- $d_{i,k,w-Da_{i,k}}$ indique si le cycle de production (i, k) a déjà débuté.
- $d_{i,k,w} - d_{i,k,w-1} \in \{0, 1\}$ vaut 1 si et seulement si l'arrêt (i, k) débute à la semaine w . De même $d_{i,k,w-Da_{i,k}} - d_{i,k,w-1-Da_{i,k}} \in \{0, 1\}$ vaut 1 si et seulement si la campagne de production (i, k) débute à la semaine w .
- $d_{i,k,w} - d_{i,k,w-Da_{i,k}} \in \{0, 1\}$ vaut 1 uniquement pendant l'arrêt du cycle (i, k) .
- $d_{i,k,w} - d_{i,k+1,w} \in \{0, 1\}$ vaut 1 uniquement sur le cycle k du réacteur i .
- $d_{i,k,w-Da_{i,k}} - d_{i,k+1,w} \in \{0, 1\}$ vaut 1 uniquement sur la campagne de production k du réacteur i .

Variables continues Les autres variables du problème sont des variables continues :

- $r_{i,k}$: quantité de carburant à recharger pour le cycle k du réacteur $i \in \mathcal{I}$.
- $p_{i,k,s,t}$: puissance produite par le réacteur $i \in \mathcal{I}$ sur le cycle k , au pas de temps t sur le scénarios $s \in \mathcal{S}$.
- $p_{j,s,t}$ puissance produite par l'unité $j \in \mathcal{J}$, au pas de temps t sur le scénarios $s \in \mathcal{S}$.
- $x_{i,s}$, niveau de carburant du réacteur $i \in \mathcal{I}$, au dernier pas de temps t , pour le scénario s . Ces variables sont nécessaires comme exprimées dans la fonction objectif, pour la valorisation du stock final, mais ne sont pas des variables décisionnelles du problème : les variables de rechargement et les productions nucléaires déterminent les stocks finaux. Pour la linéarisation du problème, il était nécessaire d'introduire ces variables, la linéarisation est proposée plus bas.

Remarque 1 On remarquera que l'écriture des contraintes du problème a nécessité de distinguer les puissances de productions nucléaires sur chaque cycle pour les réacteurs nucléaires. La combinatoire des variables continues est de ce fait majoritairement portée par les variables de production nucléaire.

Variables redondantes Les variables précédemment définies sont les seules variables décisionnelles du problème. Les autres quantités sont directement reliées à ces notions. Ainsi pour une centrale nucléaire, la donnée du fuel initial et de toutes les décisions de production, d'arrêts et les quantités rechargées impliquent directement la connaissance du stock de combustible à tout instant.

- $x_{i,k,s}^{init}$, les niveaux de carburant du réacteur $i \in \mathcal{I}$, pour le scénario s , après le rechargement au cycle k , sont déterminés par les rechargements $r_{i,k}$ et les production $p_{i,k',s,t}$ pour $k' < k$.
- $x_{i,k,s}^{fin}$, les niveaux de carburant du réacteur $i \in \mathcal{I}$, pour le scénario s , avant le rechargement du cycle $k + 1$. sont déterminés par les rechargements $r_{i,k}$ et les productions $p_{i,k,s,t}$ pour $k' \leq k$.

Dans un modèle initial, et de même que [91], on avait déclaré ces quantités comme des variables. L'écriture de contraintes donnant des contraintes d'égalité permettait alors de supprimer ces variables. Il s'agit des contraintes suivantes :

- stock initial (CT8) : $x_{i,0,s}^{init} = \mathbf{X}i$
- équation de consommation (CT9) $x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \mathbf{D}^t p_{i,k,w}$
- pertes au rechargement (CT10) : $x_{i,k,s}^{init} - \mathbf{B}o_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k} - 1}{\mathbf{Q}_{i,k}} (x_{i,k-1}^{fin} - \mathbf{B}o_{i,k-1})$

On en déduit la formule de récurrence sur les $x_{i,k,s}^{init}$: $x_{i,k,s}^{init} - \mathbf{B}o_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k} - 1}{\mathbf{Q}_{i,k}} (x_{i,k-1,s}^{init} - \sum_w \mathbf{D}^w p_{i,k-1,s,w} - \mathbf{B}o_{i,k-1})$

Où la condition initiale est $x_{i,0,s}^{init} = \mathbf{X}i$. Il s'agit d'une formule de récurrence du type $x_{i,k,s}^{init} = a_{i,k-1,s} x_{i,k-1,s}^{init} + b_{i,k-1,s}$ avec $a_{i,k,s} = \frac{\mathbf{Q}_{i,k+1} - 1}{\mathbf{Q}_{i,k+1}}$ et $b_{i,k,s} = r_{i,k+1} - \frac{\mathbf{Q}_{i,k+1} - 1}{\mathbf{Q}_{i,k+1}} (\sum_w \mathbf{D}^w p_{i,k,s,t} - \mathbf{B}o_{i,k})$

Lemme 5.1.1 Soit $(u_n)_{n \in \mathbb{N}}$, la suite réelle récurrente définie par la formule de récurrence $u_{n+1} = a_n u_n + b_n$, où $(a_n)_{n \in \mathbb{N}}$ et $(b_n)_{n \in \mathbb{N}}$ sont deux familles de réelles. Alors :

$$u_n = \left(\prod_{l=0}^{n-1} a_l \right) u_0 + \sum_{l=0}^{n-1} \left(\prod_{m=l+1}^{n-1} a_m \right) b_l$$

Preuve : Le résultat se prouve par récurrence sur $n \in \mathbb{N}$. Pour $n = 0$ ou $n = 1$, le résultat est acquis, comme la somme (resp le produit) sur un ensemble vide vaut 0 (resp 1).

On suppose $n \geq 1$ et le résultat vérifié au rang n . Montrons que le résultat reste valide au rang $n + 1$.

$u_{n+1} = a_n u_n + b_n$, avec l'hypothèse de récurrence

$$u_{n+1} = a_n \left(\left(\prod_{l=0}^{n-1} a_l \right) u_0 + \sum_{l=0}^{n-1} \left(\prod_{m=l+1}^{n-1} a_m \right) b_l \right) + b_n = a_n \left(\prod_{l=0}^{n-1} a_l \right) u_0 + \sum_{l=0}^{n-1} a_n \left(\prod_{m=l+1}^{n-1} a_m \right) b_l + b_n$$

$$u_{n+1} = \left(\prod_{l=0}^n a_l \right) u_0 + \sum_{l=0}^{n-1} \left(\prod_{m=l+1}^n a_m \right) b_l + b_n \left(\prod_{m=n+1}^n a_m \right) = \left(\prod_{l=0}^n a_l \right) u_0 + \sum_{l=0}^n \left(\prod_{m=l+1}^n a_m \right) b_l$$

Ce qui est bien le résultat annoncé pour $n + 1$, et le lemme est prouvé par récurrence. \blacksquare

En appliquant le lemme, on a la relation suivante, en définissant pour tout $\forall (m, n) \in K$, $\mathbf{q}_{m,n} = \prod_{l=m}^n \frac{\mathbf{Q}_{i,l} - 1}{\mathbf{Q}_{i,l}}$, avec l'extension $\mathbf{q}_{m,n} = 1$ si $m > n$ et $\mathbf{q}_{m,m} = \frac{\mathbf{Q}_{i,m} - 1}{\mathbf{Q}_{i,m}}$;

$$x_{i,k,s}^{init} = \mathbf{q}_{1,k} \mathbf{X}i + \sum_{l=0}^{k-1} \mathbf{q}_{l+2,k} \left(r_{i,l+1} - \mathbf{q}_{l+1,l+1} \left(\sum_t \mathbf{D}^t p_{i,l,s,t} - \mathbf{B}o_{i,l} \right) \right) \quad (5.3)$$

$$x_{i,k,s}^{fin} = \mathbf{q}_{1,k} \mathbf{X} \mathbf{i}_i + \sum_{l=0}^{k-1} \mathbf{q}_{l+2,k} \left(r_{i,l+1} - \mathbf{q}_{l+1,l+1} \left(\sum_t \mathbf{D}^t p_{i,l,s,t} - \mathbf{B} \mathbf{o}_{i,l} \right) \right) - \sum_t \mathbf{D}^t p_{i,k,s,t} \quad (5.4)$$

Pour la lisibilité de la suite, on gardera les expressions $x_{i,k,s}^{init}$ et $x_{i,k,s}^{fin}$, en gardant à l'esprit que ce ne sont pas directement de nouvelles variables, mais des expressions linéaires des variables de rechargement $r_{i,k}$ et des puissances de production nucléaire $p_{i,k,s,w}$.

Linéarisation du stock final On s'attache ici à fournir les contraintes permettant de linéariser et de définir les variables de stock de combustibles résiduels au dernier pas de temps : $x_{i,s}$, niveau de carburant du réacteur $i \in \mathcal{I}$ déterminées par les variables de rechargement et les productions nucléaires. Une relation non linéaire pourrait s'obtenir ainsi, en remarquant que $x_{i,s}$ est le stock final sur le dernier cycle en cours, ie tel que au dernier pas de temps W , $d_{i,k,W} - d_{i,k-1,W} = 1$, $x_{i,s} = \sum_k x_{i,k,s}^{fin} (d_{i,k,W} - d_{i,k-1,W})$.

La linéarisation de ces contraintes quadratiques peut être écrite en introduisant $x_{i,s} \leq M_i$ où M_i est une constante suffisamment grande pour qu'on ait toujours $x_{i,s}^{fin} \leq M_i$. On peut prendre par exemple $M_i = \max_k \mathbf{S} \max_{i,k}$ (CT 11).

$$\forall i, k, s, x_{i,s} \leq x_{i,k,s}^{fin} + M_i (d_{i,k,W} - d_{i,k+1,W}) \quad (5.5)$$

En effet, si $d_{i,k,w_t} - d_{i,k-1,w_t} = 1$, t est bien dans le cycle k , et la contrainte active est $x_{i,s,t} \leq x_{i,k,s}^{init} - \sum_{t' < t} \mathbf{D}^{t'} p_{i,k,s,t'}$, les autres contraintes étant $x_{i,s} \leq x_{i,k,s}^{fin} + M_i$, vérifiées comme $x_{i,s} \leq M_i$ et $x_{i,k,s}^{fin} \geq 0$.

5.1.2 Considérations de planning initial

Cette section fournit des réponses sur les problématiques de réoptimisation dynamique en usage à l'opérationnel, où les planning d'arrêts sont recalculés tous les mois. La solution courante calculée le mois précédent est impactante sur le calcul du nouveau planning. Certains arrêts nécessitent des contractualisations et des organisations lourdes et ne peuvent pas être déplacés facilement. Fixer un arrêt individuellement peut se faire via des contraintes de fenêtre de temps très restreintes voire figées.

Une approche permettant de ne pas trop perturber le planning initial globalement est d'introduire dans la fonction objectif pour chaque arrêt une pénalisation des écarts à la solution initiale. On définit alors $\mathbf{C}_{i,k,w}^{pen}$ correspondant à la pénalisation payée par rapport à la solution initiale si l'arrêt (i, k) est planifié à la semaine w , alors que la solution initiale était $W_{i,k}^0$. La pénalisation s'écrit linéairement :

$$\mathbf{C}^{desorg} = \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) \quad (5.6)$$

N'importe quelle fonction de pénalisation peut être utilisée. On pourra par exemple prendre en compte un coût proportionnel à l'amplitude du déplacement, alors $\mathbf{C}_{i,k,w}^{pen} = |w - W_{i,k}^0|$ ou à son carré $\mathbf{C}_{i,k,w}^{pen} = (w - W_{i,k}^0)^2$, ou un coût constant alloué si il y a un déplacement de la date prévue, indépendamment de l'amplitude de déplacement $\mathbf{C}_{i,k,w}^{pen} =_{i,k} \mathbb{1}_{w \neq W_{i,k}}$.

Nous pouvons également écrire des contraintes pour limiter le nombre de modifications autorisées par rapport au planning de référence. Soit \mathcal{C} un sous ensemble d'arrêts, on définit $\mathbf{Nmax}^{\mathcal{C}}$ le nombre maximal de modifications pour ce sous ensemble \mathcal{C} . Les contraintes à ajouter sont alors les suivantes :

$$\sum_{(i,k) \in \mathcal{C}} (1 + d_{i,k,W^0(i,k)-1} - d_{i,k,W^0(i,k)}) \leq \mathbf{Nmax}^{\mathcal{C}} \quad (5.7)$$

Dans les deux cas, il s'agit d'une adaptation mineure du problème, qui ne change pas le nombre de variables du problème. Dans le cas des jeux de données ROADEF, on n'avait pas de solution initiale, cela revient à définir $\mathbf{C}_{i,k,w}^{pen} = 0$, et $\mathcal{C} = \emptyset$.

5.1.3 Fonction objectif

La fonction objectif comprend les coûts de rechargement, des coûts pénalisant l'écart à une solution initiale, et les coûts de production associés au planning. Cela s'écrit comme l'expression linéaire :

$$\sum_{i,k} C_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} C_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) \sum_{j,s,t} \pi_s C_{j,s,t}^{prd} \mathbf{D}^t p_{j,s,t} - \sum_{i,s} \pi_s C_{i,s}^{val} x_{i,s} \quad (5.8)$$

5.2 Ecriture des contraintes

5.2.1 Contraintes sur les niveaux de production

Comme la contrainte CT1 du challenge ROADEF, la production totale doit satisfaire exactement la demande $\mathbf{Dem}^{t,s}$ à tout instant pour tous les scénarios considérés :

$$\forall s, t, \quad \sum_{i,k} p_{i,k,s,t} + \sum_j p_{j,s,t} = \mathbf{Dem}^{t,s} \quad (5.9)$$

Suivant les contraintes CT2, les unités flexibles ont les bornes de production suivantes :

$$\forall j, s, t, \quad \mathbf{Pmin}_{j,t}^s \leq p_{j,s,t} \leq \mathbf{Pmax}_{j,t}^s \quad (5.10)$$

Pour les réacteurs nucléaires, les contraintes de bornes de productions sont les suivantes, avec une production nulle pendant les arrêts, et en dehors de leur cycle de production propre. Cela modélise les contraintes CT3, CT4 et CT5.

$$\forall i, k, s, t, \quad 0 \leq p_{i,k,s,t} \leq \mathbf{Pmax}_i^t (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \quad (5.11)$$

Les autres contraintes techniques de productions nucléaires (CT6 et CT12) sont relâchées. Des expressions de contraintes moins fortes que les contraintes du challenge seront investiguées plus bas.

5.2.2 Contraintes liées aux quantités de combustible

On écrit à présent les contraintes portant sur les niveaux de combustibles et de recharges. La première contrainte vient des bornes de rechargement, qui s'écrivent conditionnellement à réalisation de l'arrêt k considéré dans l'horizon de temps. Cela fait intervenir alors les variables $d_{i,k,W}$, pour un rechargement nul sur un arrêt non réalisé ie $d_{i,k,W} = 0$, et avec les bornes $\mathbf{Rmin}_{i,k}$ et $\mathbf{Rmax}_{i,k}$ si $d_{i,k,W} = 1$:

$$\forall i, k, \quad \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \quad (5.12)$$

Notre définition des variables a déjà fait porter les équations de stocks initial (CT8), de consommation de combustible liant stocks et productions (CT9) et l'équation de perte de combustible à un arrêt dans la définition des $x_{i,k,s}^{init}$ et $x_{i,k,s}^{fin}$ comme expression linéaire des puissances et des recharges.

Les contraintes restant à écrire sont les contraintes de positivité du stock, de stock maximal, d'anticipation maximale (stock maximal avant un rechargement). Ces contraintes porteront sur les $x_{i,k,s}^{init}$ et $x_{i,k,s}^{fin}$. Par l'équation de consommation, les stocks de combustibles sont décroissants au cours d'un cycle. Les contraintes de stock maximal ne s'écrivent donc uniquement sur les valeurs en début de cycle :

$$\forall i, k, s, \quad x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \quad (5.13)$$

De même, la positivité du stock ne doit être réalisée qu'en fin de cycle, la décroissance du combustible impliquant alors la positivité du stock à tout instant :

$$\forall i, k, s, \quad 0 \leq x_{i,k,s}^{fin} \quad (5.14)$$

On rappelle que $x_{i,k,s}^{init}$ et $x_{i,k,s}^{fin}$ sont des expressions linéaires des autres variables, l'équation (5.14) doit être écrite car non comprise dans la définition et la positivité des variables.

La contrainte d'anticipation maximale n'autorise l'arrêt d'un cycle i, k que si le niveau de combustible est inférieur à $\mathbf{Amax}_{i,k+1}$. Ces contraintes portent sur les stocks en fin de cycle. Toutefois, cette contrainte n'est pas toujours $x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1}$: pour le dernier cycle cours, arrêté à la fin de l'horizon de temps, la contrainte n'est pas active. Cette distinction donne lieu à l'écriture suivante :

$$\forall i, k, s, \quad x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} d_{i,k+1,W} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \quad (5.15)$$

En effet, pour un cycle achevé k , on a $d_{i,k+1,W} = 1$, et la contrainte est l'anticipation maximale $x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1}$. Dans le cas contraire, pour un arrêt inachevé, ie $d_{i,k+1,W} = 0$, on a la contrainte de stock maximal $x_{i,k,s}^{fin} \leq \mathbf{Smax}_{i,k}$ automatiquement satisfaite. Dans la suite, on condense l'écriture en écrivant les contraintes (5.12), (5.13), (5.14), (5.15), dans le système matriciel $T_d^{stockl} d + T_r^{stockr} r + W^{stock} p \geq h^{stock}$.

5.2.3 Contraintes d'ordonnement sur les dates d'arrêts

Il s'agit ici de traiter les contraintes couplantes sur les dates d'arrêts, et donc ne faisant intervenir que les variables $d_{i,k,w}$. Les contraintes de définition (5.1) rentraient déjà dans ce cadre. Les contraintes CT13 de fenêtres de temps rentrent également dans ce cadre. Les autres contraintes concernées sont les contraintes de CT14 à CT19, contraintes d'ordonnements induisant des espacements minimaux et chevauchements maximaux entre arrêts ou les contraintes CT20 et CT21 induisant un nombre maximal d'arrêt en parallèle. En outre, on introduit des contraintes additionnelles non présentes dans le challenge ROADEF, de longueur minimum ou maximale d'un cycle de production, qui se complètent bien de la connaissance opérationnelle du problème, de telles contraintes étaient d'ailleurs modélisées dans TANNAT.

Contraintes de fenêtres de temps Tout d'abord, des contraintes de fenêtres de temps imposées, l'arrêt (i, k) commence au plus tôt à la semaine $\mathbf{To}_{i,k}$ et au plus tard à $\mathbf{Ta}_{i,k}$.

$$\forall i, k, w < \mathbf{To}_{i,k}, \quad d_{i,k,w} \leq 0 \quad (5.16)$$

$$\forall i, k, w \geq \mathbf{Ta}_{i,k}, \quad d_{i,k,w} \geq 1 \quad (5.17)$$

Contraintes d'espacements entre arrêts consécutifs On considère ici des contraintes d'espacements entre arrêts consécutifs, de longueur minimum ou maximale d'un cycle de production, non présentes dans le challenge ROADEF :

$$\forall i, k, w, \quad d_{i,k+1,w} \leq d_{i,k,w} - \mathbf{Lmin}_{i,k} \quad (5.18)$$

$$\forall i, k, w, \quad d_{i,k+1,w} \geq d_{i,k,w} - \mathbf{Lmax}_{i,k} \quad (5.19)$$

On notera que l'on a défini $\mathbf{Lmin}_{i,k}$, $\mathbf{Lmax}_{i,k}$ comme les longueurs d'espacement entre deux arrêts consécutifs. S'il s'agissait de bornes de la longueur du cycle de production, on aurait alors dans ces équations $\mathbf{Lmin}_{i,k} + \mathbf{Da}_{i,k}$ et $\mathbf{Lmax}_{i,k} + \mathbf{Da}_{i,k}$.

Les contraintes de bornes de rechargement et d'anticipation maximales impliquent des bornes sur la longueur minimale des cycles de production. Du fait de la positivité du stock, le stock minimal au début du cycle (i, k) est $\mathbf{Rmin}_{i,k}$. En définissant $P_i = \max_t \mathbf{Pmax}_{i,t}$ la puissance maximale de l'unité i , l'équation

de variation du stock impose au minimum $\mathbf{Lmin}_{i,k} = \left\lceil \frac{\mathbf{Rmin}_{i,k} - \mathbf{Amax}_{i,k}}{\mathbf{D}^w P_i} \right\rceil$ semaines pour attendre le niveau d'anticipation maximale et pouvoir passer à l'arrêt suivant.

Pour les contraintes de longueurs maximale, on se ramène au challenge ROADEF en omettant ces contraintes, ou en posant $\mathbf{Lmax}_{i,k} = W$. L'intérêt de telles contraintes est que pour ces données sont bien connues des praticiens pour les meilleures solutions, ce qui permet de restreindre l'espace de recherche.

Contraintes limitant le nombre d'arrêts en parallèle Les contraintes CT20 et CT21 limitent le nombre d'arrêts en parallèle, CT20 par un nombre maximal d'arrêts en parallèle, CT21 avec une puissance maximale d'indisposition. On rappelle qu'avec notre définition des variables $d_{i,k,w}$, un réacteur i est arrêté par l'arrêt (i, k) à la semaine w si et seulement si $d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}} = 1$. La contrainte CT20 impose pour toute semaine w un nombre maximal d'arrêt en parallèle $\mathbf{N20}_w$ sur un ensemble de réacteurs $\mathbf{A20}_w$. Il s'agit d'une contrainte de ressources sur les expressions binaires $d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}}$:

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A20}_w} (d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}}) \leq \mathbf{N20}_w \quad (5.20)$$

Pour la contrainte CT21, la ressource est la puissance indisponible pour l'arrêt, ce qui est analogue :

$$\forall \mathbf{IT21} \quad \sum_{i \in \mathbf{C21}} \sum_{w \in \mathbf{IT21}} \left(\sum_{t=t_w}^{t_{w+1}-1} \mathbf{Pmax}_i^t \right) \sum_k (d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}}) \leq \mathbf{Imax} \quad (5.21)$$

Contraintes de chevauchements maximal, CT14, CT15 Pour les contraintes CT14, définissent pour un ensemble donné de réacteurs concernés $\mathbf{A14}$, un nombre de semaines $\mathbf{Se14}$. Si $\mathbf{Se14} \geq 0$, deux arrêts (i, k) et (i', k') doivent être espacés de $\mathbf{Se14}$ semaines entre la fin du premier arrêt et le début de l'arrêt suivant. Si $\mathbf{Se14} < 0$, les plages d'arrêts ne peuvent se recouvrir que sur $-\mathbf{Se14}$ semaines. En notant $W_{i,k}$ la semaine où débute l'arrêt (i, k) , cela peut s'écrire avec des contraintes disjonctives, pour tout $(i, k), (i', k')$, on a $W_{i,k} \leq W_{i',k'} + \mathbf{Da}_{i',k'} + \mathbf{Se14}$ ou $W_{i',k'} \leq W_{i,k} + \mathbf{Da}_{i,k} + \mathbf{Se14}$. Il s'agit d'une contrainte conjonctive, qui peut se linéariser en ajoutant des variables binaires. Outre le fait d'ajouter des variables binaires, cette linéarisation fournit des relaxations continues de mauvaise qualité.

La formulation alternative se base sur la situation équivalente où chaque arrêt (i, k) tel que $\mathbf{Da}_{i,k} + \mathbf{Se14} \geq 0$ consomme une ressource fictive entre $W_{i,k}$ et $W_{i,k} + \mathbf{Da}_{i,k} + \mathbf{Se14}$. On formule l'espacement en disant que cette ressource fictive est limitée à 1, et donc aucune autre centrale ne pourra consommer cette ressource sur ces semaines là, ce qui correspond exactement à interdire un début d'arrêt ou un chevauchement sur cette période. En remarquant que la ressource consommée à la semaine w pour le cycle (i, k) est $R_{i,k,w} = d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}+\mathbf{Se14}}$ si $\mathbf{Da}_{i,k} + \mathbf{Se14} \geq 0$. Cela correspond à l'écriture de contraintes, avec la notation $x^+ = \max(0, x)$:

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A14}} (d_{i,k,w} - d_{i,k,w-(\mathbf{Da}_{i,k}+\mathbf{Se14})^+}) \leq 1 \quad (5.22)$$

On remarquera que cela fait indexer les contraintes sur les semaines, donnant plus de contraintes, mais donnant des meilleurs relâchés continus. Les contraintes CT15 sont analogues, sur des plages de définition $[\mathbf{ID15}, \mathbf{IF15}]$. Avec une telle formulation des contraintes CT14, on a la même écriture pour CT15 en se restreignant à $[\mathbf{ID15}, \mathbf{IF15}]$:

$$\forall w \in [\mathbf{ID15}, \mathbf{IF15}], \quad \sum_{(i,k) \in \mathbf{A15}} (d_{i,k,w} - d_{i,k,w-(\mathbf{Da}_{i,k}+\mathbf{Se15})^+}) \leq 1 \quad (5.23)$$

Contraintes d'espacement entre débuts d'arrêts Les contraintes CT 16 du challenge ROADEF imposent une durée minimale $\mathbf{Se16} > 0$ entre les dates de début d'arrêts des centrales dans un ensemble $\mathbf{A16}$. On peut écrire ces contraintes comme précédemment en introduisant une ressource fictive, un arrêt (i, k) démarré à la semaine $W_{i,k}$ consommant la ressource sur $[W_{i,k}, W_{i,k} + \mathbf{Se16}]$. Cela donne l'écriture de contraintes suivantes, identique à [74] :

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A16}} (d_{i,k,w} - d_{i,k,w-\mathbf{Se16}}) \leq 1 \quad (5.24)$$

Contraintes d'espacement entre dates de fin d'arrêt Les contraintes CT17 du challenge ROADEF imposent une durée minimale $\mathbf{Se17} > 0$ entre les dates de début des cycles de production des centrales dans un ensemble $\mathbf{A17}$. Ces contraintes d'espacement sont analogues, en portant sur $d_{i,k,w-\mathbf{Da}_{i,k}}$, qui dépend de s ici. Tout scénario de prolongation d'arrêt, doit respecter ces contraintes, qui s'écrivent :

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A17}} (d_{i,k,w-\mathbf{Da}_{i,k}} - d_{i,k,w-\mathbf{Da}_{i,k}-\mathbf{Se17}}) \leq 1 \quad (5.25)$$

Contraintes d'espacement entre dates de fins et de débuts d'arrêts. Pour ces contraintes, CT18 dans le challenge ROADEF, les unités d'un ensemble $\mathbf{A18}$ doivent être espacés d'au moins $\mathbf{Se18}$ semaines entre la fin d'arrêt et le début d'un autre arrêt de $\mathbf{A18}$. Cela s'écrit de manière analogue :

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A18}} (d_{i,k,w} - d_{i,k,w-\mathbf{Se18}}) + (d_{i,k,w-\mathbf{Da}_{i,k}} - d_{i,k,w-\mathbf{Da}_{i,k}-\mathbf{Se18}}) \leq 1 \quad (5.26)$$

Contraintes d'utilisation de ressources pendant un arrêt Les contraintes CT19 du challenge ROADEF portent sur l'utilisation d'une ressource, en quantité limitée $\mathbf{Q19}$, utilisée pour un arrêt (i, k) entre la semaine $\mathbf{L19}_{i,k}$ et $\mathbf{L19}_{i,k} + \mathbf{Tu19}_{i,k}$, de l'arrêt où $\mathbf{Tu19}_{i,k}$ est la durée d'utilisation de la ressource en semaine. La variable binaire indiquant si la ressource est utilisée à la semaine w est ainsi $d_{i,k,w-\mathbf{L19}_{i,k}} - d_{i,k,w-\mathbf{L19}_{i,k}-\mathbf{Tu19}_{i,k}}$. On a alors l'écriture suivante de contrainte de ressources de même que précédemment :

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A19}} (d_{i,k,w-\mathbf{L19}_{i,k}} - d_{i,k,w-\mathbf{L19}_{i,k}-\mathbf{Tu19}_{i,k}}) \leq \mathbf{Q19} \quad (5.27)$$

5.2.4 Gestion des contraintes de stretch CT6

Formulation continue du stretch En fin de campagne, c'est à dire quand le stock de combustible est en deçà du niveau de bore nul, $\mathbf{Bo}_{i,k}$, la production est dans la réalité imposée suivant un profil décroissant, "stretch", comme représenté sur la figure 5.2. La modélisation exacte du stretch, réalisée dans [74] nécessite l'introduction de variables binaires à chaque pas de temps, chaque cycle et chaque mode de production $m \in \mathcal{M}$ pour indiquer le mode exact de production par rapport à cette courbe de profil décroissant. Pour notre modélisation, les seules variables binaires sont uniquement les variables décisionnelles du problème, les dates d'arrêt. On assouplit la définition du domaine de production, représenté sur la figure 5.3. Au plan métier, cette approximation est justifiée comme faible, l'optimum économique tendant à placer la production nucléaire au maximum. On utilise que la courbe de Stretch est concave, le domaine de production est défini par l'intersection des demis plans inférieurs définis par ces équations. L'équation de droite correspondant au segment de droite entre les modes $m-1$ et m est :

$$Y = \frac{\mathbf{c}_{i,k,m-1} - \mathbf{c}_{i,k,m}}{\mathbf{f}_{i,k,m-1} - \mathbf{f}_{i,k,m}} (X - \mathbf{f}_{i,k,m}) + \mathbf{c}_{i,k,m}$$

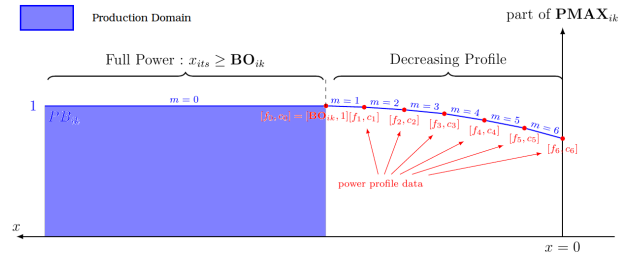


FIGURE 5.2 – Domaine de production pour la modélisation ROADEF

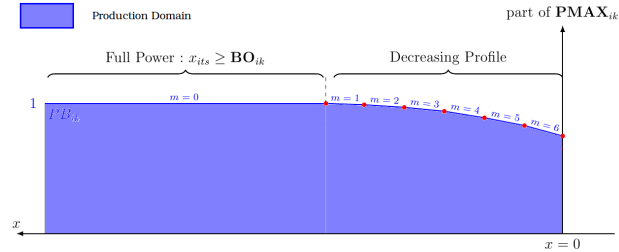


FIGURE 5.3 – Domaine de production pour notre modélisation du stretch

où Y est le rapport de la puissance maximale disponible due au stretch sur la puissance maximale réelle, et X est le stock de carburant. On introduit alors des variables $x_{i,s,t}$ définies par :

$$\forall i, k, s, t, \quad x_{i,s,t} \leq x_{i,k,s}^{init} - \sum_{t' \leq t} \mathbf{D}^{t'} p_{i,k,s,t'} + M_i (1 - d_{i,k,w_t} + d_{i,k-1,w_t}) \quad (5.28)$$

où M_i est une constante suffisamment grande pour qu'on ait toujours $x_{i,s,t} \leq M_i$. On peut prendre par exemple $M_i = \max_k \mathbf{Smax}_{i,k}$. En effet, si $d_{i,k,w_t} - d_{i,k-1,w_t} = 1$, t est bien dans le cycle k , et la contrainte active est $x_{i,s,t} \leq x_{i,k,s}^{init} - \sum_{t' < t} \mathbf{D}^{t'} p_{i,k,s,t'}$, les autres contraintes étant $x_{i,s,t} \leq M_i$, vérifiées par définition de M_i . La linéarisation est alors :

$$\forall i, k, s, t, x_{i,s,t} \leq x_{i,k,s}^{init} - \sum_{t' \leq t} \mathbf{D}^{t'} p_{i,k,s,t'} + \mathbf{Smax}_{i,k} (1 - d_{i,k,w_t} + d_{i,k-1,w_t}) \quad (5.29)$$

On notera qu'il s'agit d'une contrainte d'inégalité dans la linéarisation, l'égalité étant assurée à l'optimum. Les contraintes de stretch s'écrivent ainsi :

$$\forall i, k, s, t, m > 0, \quad \frac{p_{i,k,s,t}}{\mathbf{Pmax}_i^t} \leq \frac{\mathbf{c}_{i,k,m-1} - \mathbf{c}_{i,k,m}}{\mathbf{f}_{i,k,m-1} - \mathbf{f}_{i,k,m}} (x_{i,s,t} - \mathbf{f}_{i,k,m}) + \mathbf{c}_{i,k,m} \quad (5.30)$$

Dans la formulation précédente, on aboutit à une formulation des contraintes de stretch indexée sur 5 indices, ce qui donne un nombre de contraintes bien plus important que pour les autres types de contraintes. Une première manière de réduire le nombre de contraintes est d'opérer un preprocessing, en remarquant que les contraintes de stretch ne peuvent être actives que sur un ensemble de pas de temps plus restreint, en fin de cycle où le stock est inférieur à $\mathbf{Bo}_{i,k}$. Les bornes de production, de dates d'arrêts et de rechargements induisent de telles restrictions. On peut escompter que le preprocessing des solveurs de PLNE modernes comme Cplex réalisent de telles opérations. Une seconde manière de diminuer le nombre de contraintes est de les écrire en considérant moins de points dans la courbe de stretch. L'intérêt de ces contraintes étant avant tout d'estimer les pertes engendrées des cycles de productions longs qui nécessitent du stretch. Dans le cas le plus extrême, écrire un seul mode de stretch permet cela. Une dernière relaxation raisonnable est de considérer de telles contraintes sur un horizon de temps ou sur un nombre de cycle restreint.

Cas particulier Dans la formulation précédente du stretch, on peut supprimer l'indice k quand les coefficients $\mathbf{c}_{i,k,m}, \mathbf{f}_{i,k,m-1}$ ne dépendent pas du cycle k , caractéristique propre à un réacteur nucléaire, et invariant dans le temps. Une telle hypothèse n'est pas toujours vérifiée dans les données ROADEF, quelques variations apparaissant dans des cycles lointains pour quelques rares unités. En toute rigueur, on peut utiliser la formulation suivante pour les unités où cette hypothèse d'invariance est réalisée. Pour une application opérationnelle, étendre une telle hypothèse à tout le parc nucléaire semble raisonnable opérationnellement, là encore, il s'agit d'estimer les impacts du stretch sur le planning. L'écriture de telles contraintes porte alors sur la puissance globale $\sum_k p_{i,k,s,t}$, distinction de cycles :

$$\forall i, s, t, m > 0 \quad \sum_k \frac{p_{i,k,s,t}}{\mathbf{Pmax}_i^t} \leq \frac{\mathbf{c}_{i,m-1} - \mathbf{c}_{i,m}}{\mathbf{f}_{i,m-1} - \mathbf{f}_{i,m}} (x_{i,s,t} - \mathbf{f}_{i,m}) + \mathbf{c}_{i,m} \quad (5.31)$$

5.3 Bilan des formulations

A l'issue de ce chapitre, nous avons modélisé de manière exacte toutes contraintes écrites du challenge ROADEF sauf les contraintes CT6 et CT12 (stretch et modulation). Cette restriction venait du choix de restreindre le nombre de variables binaires pour une résolution efficace, les variables seules binaires étant liées au dates d'arrêt, et amenait à se focaliser sur ces variables décisionnelles du problème.

Avec une telle modélisation, nous avons pu fournir des extensions intéressantes du point de vue opérationnel : contraintes de longueurs minimales ou maximales d'un cycle de production, et des considérations de planning initial, en pénalisant les écarts à une solution initiale (non fournie pour le challenge ROADEF, mais utilisée et importantes pour des considérations opérationnelles), ou en écrivant des contraintes indiquant un nombre maximal de modifications par rapport à la solution courante. Pour les contraintes CT6 de stretch, des formulations moins fortes ont été proposées, donnant la distinction des formulations obtenues. Ces formulations ont été écrites dans le cas général, en considérant des scénarios stochastiques et des pas de temps de production différents des pas de temps semainiers de décisions d'arrêt. Pour le cas déterministe, cela revient à considérer un seul scénario stochastique, $\mathcal{S} = \{s\}$ ou à omettre les indices s . Pour une formulation sur pas de temps agrégé, il suffit de considérer $t = w$.

5.3.1 Formulation sans stretch

Notre formulation du problème sans les contraintes CT6 et CT12 est la suivante. Cette formulation donne des bornes inférieures du challenge ROADEF en retirant les extensions de contraintes développées. Notre formulation en omettant les contraintes CT6 et CT12 présente de nombreuses similitudes avec la formulation de [91], modélisant les mêmes contraintes, en ayant uniquement des variables binaires pour les dates d'arrêt. La différence majeure vient de la définition des variables, leur formulation utilisant $y_{i,k,w} = d_{i,k,w} - d_{i,k,w-1}$, avec $y_{i,k,w} = 1$ si et seulement si l'arrêt (i, k) commence exactement à la semaine w . De telles variables binaires indiquent directement la date d'arrêt. On a la contrainte de modélisation $\sum_w y_{i,k,w} \leq 1$, et les contraintes s'obtiennent d'une formulation à l'autre par $d_{i,k,w} = \sum_{w' \leq w} y_{i,k,w'}$. Il s'agit en fait de la même distinction que pour le chapitre 4, entre les variables de paliers et les variables de niveaux, formulations équivalentes et reliées par un isomorphisme. Un tel choix est justifié par la résolution des solveurs. L'autre différence et amélioration par rapport à la modélisation de [91] vient de la suppression de variables opérée à la première section de ce chapitre, éliminant les variables $x_{i,k,s}^{fin}$ et $x_{i,k,s}^{init}$ à l'aide des contraintes CT8, CT9, et CT10.

$$\begin{aligned}
\min \quad & \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,s,t} \pi_s \mathbf{C}_{j,s,t}^{prd} \mathbf{D}^t p_{j,s,t} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
& M^{ordo} d \geq b^{ordo} \\
& T_d^{stockl} d + T_r^{stock} r + W^{stock} p \geq h^{stock} \\
\forall j, s, t, \quad & \mathbf{Pmin}_{j,t}^s \leq p_{j,s,t} \leq \mathbf{Pmax}_{j,t}^s \\
\forall i, k, s, t, \quad & p_{i,k,s,t} \leq \mathbf{Pmax}_i^t (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \\
\forall s, t, \quad & \sum_{i,k} p_{i,k,s,t} + \sum_j p_{j,s,t} = \mathbf{Dem}_t^s \\
\forall i, k, s, \quad & x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + M_i (d_{i,k,W} - d_{i,k+1,W}) \\
& d \in \{0; 1\}^M, r, p, x^{fin} \geq 0
\end{aligned}$$

5.3.2 Formulation avec bornes supérieures du Stretch

Pour introduire des contraintes de stretch dans la formulation précédente sans devoir rajouter des variables binaires, nous avons considéré uniquement des bornes supérieures de la production en cas de stretch, comme schématisé sur la figure 5.3. La concavité de la courbe de profil décroissant permet d'écrire des contraintes linéaires reliant la borne maximale de puissance en stretch au stock résiduel. Cela nécessite ainsi d'introduire des variables continues $x_{i,s,t}$ indiquant le stock à tout instant. De telles variables ne s'obtiennent pas comme combinaison linéaire des variables précédemment introduites, et sont effectivement rajoutées, de même que des contraintes linéaires permettant de les définir exactement en fonction des autres variables. On a alors la formulation suivante, qui donne toujours des bornes inférieures du problème complet du challenge ROADEF :

$$\begin{aligned}
\min \quad & \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,s,t} \pi_s \mathbf{C}_{j,s,t}^{prd} \mathbf{D}^t p_{j,s,t} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
& A^{ordo} d \geq a^{ordo} \\
& T_d^{stockl} d + T_r^{stock} r + W^{stock} p \geq h^{stock} \\
\forall j, t, \quad & \mathbf{Pmin}_{j,t}^s \leq p_{j,s,t} \leq \mathbf{Pmax}_{j,t}^s \\
\forall i, k, s, t, m, \quad & \frac{p_{i,k,s,t}}{\mathbf{Pmax}_i^t} \leq \frac{\mathbf{c}_{i,k,m-1} - \mathbf{c}_{i,k,m}}{\mathbf{f}_{i,k,m-1} - \mathbf{f}_{i,k,m}} (x_{i,k,t} - \mathbf{f}_{i,k,m}) + \mathbf{c}_{i,k,m} \\
\forall i, k, s, t, \quad & x_{i,s,t} \leq x_{i,k,s}^{init} - \sum_{t' \leq t} \mathbf{D}^{t'} p_{i,k,s,t'} + M_i (1 - d_{i,k,w_t} + d_{i,k-1,w_t}) \\
\forall i, k, s, t, \quad & p_{i,k,s,t} \leq \mathbf{Pmax}_i^t (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \\
\forall s, t, \quad & \sum_{i,k} p_{i,k,s,t} + \sum_j p_{j,s,t} = \mathbf{Dem}_t^s \\
\forall i, k, s, \quad & x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + M_i (d_{i,k,W} - d_{i,k+1,W}) \\
& d \in \{0; 1\}^M, r, p, x^{fin}, x \geq 0
\end{aligned}$$

Les contraintes de stretch sont alors indexées sur 5 indices, ce qui donne un nombre de contraintes bien plus important que pour les autres types de contraintes.

5.3.3 Formulation sur un cas particulier de stretch

Dans la formulation précédente des contraintes de stretch, on peut supprimer l'indice k dans le cas où la courbe de profil décroissant (ie les coefficients $\mathbf{c}_{i,k,m}, \mathbf{f}_{i,k,m}$) ne dépend pas du cycle k et est une caractéristique propre à un réacteur nucléaire, et invariant dans le temps. Une telle hypothèse n'est pas toujours vérifiée dans les données du challenge ROADEF (quelques variations apparaissant dans des cycles

lointains pour quelques rares unités). Pour une application opérationnelle, étendre une telle hypothèse à tout le parc nucléaire semble raisonnable opérationnellement.

$$\begin{aligned}
\min \quad & \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,s,t} \pi_s \mathbf{C}_{j,s,t}^{prd} \mathbf{D}^t p_{j,s,t} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
& A^{ordo} d \geq a^{ordo} \\
& T_d^{stock} d + T_r^{stock} r + W^{stock} p \geq h^{stock} \\
\forall j, s, t, \quad & \mathbf{Pmin}_{j,t}^s \leq p_{j,s,t} \leq \mathbf{Pmax}_{j,t}^s \\
\forall i, s, t, \quad & 0 \leq p_{i,k,s,t} \leq \mathbf{Pmax}_i^t (d_{i,k,w_t - \mathbf{D}a_{i,k}} - d_{i,k+1,w_t}) \\
\forall i, s, t, m, \quad & \sum_k \frac{p_{i,k,s,t}}{\mathbf{Pmax}_i^t} \leq \frac{\mathbf{c}_{i,m-1} - \mathbf{c}_{i,m}}{\mathbf{f}_{i,m-1} - \mathbf{f}_{i,m}} (x_{i,s,t} - \mathbf{f}_{i,m}) + \mathbf{c}_{i,m} \\
\forall s, t, \quad & \sum_{i,k} p_{i,k,s,t} + \sum_j p_{j,s,t} = \mathbf{Dem}_t^s \\
\forall i, k, s, t, \quad & x_{i,s,t} \leq x_{i,k,s}^{init} - \sum_{t' \leq t} \mathbf{D}^{t'} p_{i,k,s,t'} + M_i (1 - d_{i,k,w_t} + d_{i,k-1,w_t}) \\
& d \in \{0; 1\}^M, r, p, x^{fin}, x \geq 0
\end{aligned}$$

Les contraintes de stretch sont alors indexées sur 4 indices. Les contraintes de stretch ajoutées sont toujours prépondérantes dans la combinatoire des contraintes.

Chapitre 6

Résolution exacte et bornes duales du challenge ROADEF

Résumé : Ce chapitre étudie la résolution exacte des formulations obtenues au chapitre 5. On analyse les structures de convergences et les limites de la résolution frontale par Branch&Bound. Les différentes formulations sont étudiées, avec ou sans stretch, avec ou sans pénalisation d'écart à une solution initiale. On a pu déduire des bornes duales calculables pour le challenge ROADEF en prouvant que les agrégations nécessaires à la calculabilité permettent encore d'obtenir des bornes duales valides. Cela a amélioré les meilleures bornes de la littérature sur toutes les instances du challenge ROADEF. Ce travail a fait l'objet de communications au congrès ROADEF 2015.

A l'issue du chapitre précédent, une modélisation PLNE du problème de planification des arrêts et de maintenances des centrales nucléaires a été fournie. Il s'agit ici d'analyser les caractéristiques de la résolution frontale, dans le cadre déterministe sur scénario moyen, en agrégeant les pas de temps de production à la semaine. Il s'agit du cadre de la résolution de l'état de l'art à EDF, dont les limites ont été analysées dans l'état de l'art du chapitre 2. Une attention toute particulière sera portée à la résolution des instances B8 et B9, les plus difficiles de par leur combinatoire avec peu de contraintes de fenêtres de temps au delà, mais aussi les plus représentatives du besoin opérationnel. La première partie analyse les caractéristiques et les limites de la résolution frontale, et fournit des premiers résultats de modélisation. La seconde partie cherchera à dériver du travail précédent un mode de calcul de bornes duales pour le problème complet du challenge ROADEF et les confronter à l'état de l'art de [35].

6.1 Résultats de résolution frontale exacte

Nous analysons ici les résultats obtenus par la résolution frontale en PLNE, nous avons utilisé Cplex version 12.5, appelé via l'interface de modélisation OPL.

6.1.1 Preprocessing exact des fenêtres de temps

On s'intéresse ici au preprocessing des fenêtres de temps. Connaître des longueurs minimales d'arrêts $\mathbf{Lmin}_{i,k}$ permet la propagation de contraintes de fenêtres de temps. Pour une borne inférieure w_0 du début de l'arrêt (i, k) , cela propage borne inférieure sur la date de l'arrêt suivant $(i, k + 1)$, aussitôt $w_0 + \mathbf{Da}_{i,k} + \mathbf{Lmin}_{i,k}$. Si un arrêt (i, k) a une date au plus tard w_1 définie sur l'horizon de temps (au plus tard $w_1 < W + 1$), l'arrêt précédent $(i, k - 1)$ devait être planifié au plus tard à la semaine $w_1 - \mathbf{Da}_{i,k-1} - \mathbf{Lmin}_{i,k-1}$.

Pour un preprocessing exact, on peut utiliser la minoration de $\mathbf{Lmin}_{i,k}$ précédemment mentionnée, en utilisant $\mathbf{Lmin}_{i,k} = \left\lceil \frac{\mathbf{Rmin}_{i,k} - \mathbf{Amax}_{i,k}}{\mathbf{D}^w P_i} \right\rceil$ où $P_i = \max_w \mathbf{Pmax}_{i,w}$. Ce preprocessing simple apporte une plus

value au preprocessing de Cplex, pour un gain en termes de nombre de variables binaires qui se répercute par de forts gains en temps de calculs. Sur les instances B8 et B9, les preprocessing sont moins efficaces que sur les autres instances, cela s'explique par le fait qu'il y a moins de contraintes de fenêtres de temps à propager. Il en résulte néanmoins une nette diminution du nombre de variables.

Instances	Sans PP	Avec PP	Gain PP	Sans PP PP cplex	Avec PP PP cplex	Gap
A1	3892	463	88,10%	483	424	12,22%
A2	7889	961	87,82%	892	761	14,69%
A3	8162	875	89,28%	841	698	17,00%
A4	17465	1798	89,71%	1998	1493	25,28%
A5	15357	2797	81,79%	2750	2494	9,31%
B6	24563	3466	85,89%	3467	3054	11,91%
B7	35768	6435	82,01%	9052	5846	35,42%
B8	69653	22482	67,72%	30626	20763	32,20%
B9	69306	25351	63,42%	35307	23675	32,95%
B10	29948	4236	85,86%	5084	3790	25,45%
X11	20081	3478	82,68%	3499	3216	8,09%
X12	27111	4348	83,96%	5321	4035	24,17%
X13	30154	4697	84,42%	4403	4104	6,79%
X14	30691	5378	82,48%	6088	4879	19,86%
X15	27233	3992	85,34%	4372	3618	17,25%
Total	417273	90757	78,25%	114183	82850	27,44%

TABLE 6.1 – Impact de notre preprocessing et du preprocessing de Cplex sur le nombre de variables binaires

6.1.2 Résultats de la formulation sans stretch

Nous analysons ici les résultats obtenus par la formulation de la section 5.3.1 sans contraintes de stretch. Nos choix de formulation se sont avérés plus judicieux que la modélisation de [91]. Nos variables binaires ont mené à des branchements plus efficaces avec Cplex que les variables SOS naturelles. Le choix d'éliminer les variables continues définies par des contraintes d'égalité était également judicieux.

La résolution frontale en PLNE est inefficace sur les instances B8 et B9 à plus forte combinatoire. En 1h, Cplex 12.3 était incapable de calculer le relâché continu par simplexe dual en vue d'une résolution ultérieure par Branch & Bound. La relaxation continue était néanmoins calculable en 1h avec l'algorithme barrière. De nettes améliorations ont été observées en passant de Cplex 12.3 à 12.5, ce que la comparaison des tableaux 8.2 et 6.2 met en évidence. De telles limites justifient d'avoir agrégé les pas de temps de production à la semaine, les limites de résolution sont fortement contraintes par les relaxations continues. La relaxation continue est de bien moins bonne qualité sur les instances B8, B9 et leurs troncatures, par rapport aux autres instances. Une explication à cela est l'absence de fin de fenêtres de temps pour les derniers cycles sur ces instances, qui font apparaître des solutions de relaxation continue fractionnaires pour tous les derniers arrêts sans fin de fenêtre de temps. Cela revient sur la solution fractionnaire à ignorer les arrêts et produire continuellement. Dans de tels cas, la relaxation linéaire est pénalisée en temps de calcul comme en qualité. De telles situations sont améliorées avec l'ajout de contraintes de longueur maximale de cycle de production imposée. Dans les autres cas, particulièrement quand plusieurs arrêts d'une tranche ne peuvent avoir lieu à une semaine donnée, les temps de calcul de relaxation continue ne sont pas critiques, et la convergence a de bonnes propriétés : dès le noeud racine, les coupes et les heuristiques primales permettent d'atteindre des valeurs proches de l'optimum.

Des bornes primales sont trouvées dès le noeud racine, sauf pour les instances B7, B8 et B9. Sur les solutions trouvées à la racines, on se situe à 0,3% des meilleures solutions trouvées en moyenne. En revanche, les premières solutions trouvées ont un coût très élevé : l'heuristique de Cplex a tendance à aller vers les fins de fenêtres de temps, et donc utilise beaucoup de production non nucléaire et coûteuse. Avec la structure de relaxation continue des instances B8 et B9, une heuristique de type RINS ne fixe que très peu de variables binaires, et donc ne peut trouver des bornes primales facilement. Un constat marquant sur la fin de convergence par Cplex, confirmé avec les heuristiques du chapitre 7, est que beaucoup de solutions

ont des coûts très proches et en deçà du seuil de tolérance de cplex de 0,01%. Pour la convergence par Branch&Bound, cela présente les mêmes désagréments que des symétries, en ralentissant la convergence et en nécessitant de nombreux branchements. Enfin, on notera sur le tableau 6.4 les forts impacts preprocessing et des restrictions de fenêtre de temps, de l'ajout des contraintes de longueur minimales et maximales de cycles, et du démarrage à chaud avec une bonne solution réalisable pour la résolution en nombre entiers.

	Valeur	LP	temps	inf0	sup0	temps	inf	sup	temps
A1	152170M	0,17%	0,09	0,00%	0,00%	1,1	0,00%	0,00%	1,1
A2	145201M	0,22%	0,4	0,02%	0,00%	3,8	0,00%	0,00%	4,9
A3	152582M	0,36%	0,11	0,05%	0,00%	4,4	0,00%	0,00%	3
A4	102421M	0,95%	3,1	0,42%	0,09%	21,4	0,00%	0,00%	334
A5	119785M	0,90%	7,5	0,53%	0,44%	55	0,15%	0,00%	3600
B6	76966M	2,23%	11,2	0,61%	0,51%	64	0,13%	0,00%	3600
B7	74233M	2,90%	66,7	0,78%	1,99%	346	0,52%	0,98%	3600
B8	73239M	9,47%	244	7,95%	NS	3320	7,95%	NS	3600
B9	72812M	8,51%	350	6,69%	NS	3600	6,69%	NS	3600
B10	69501M	3,45%	14,7	0,66%	0,27%	77	0,07%	0,00%	3600
X11	73018M	1,65%	12,1	0,58%	0,46%	124	0,37%	0,01%	3600
X12	70604M	2,98%	21,6	0,58%	0,35%	145	0,20%	0,00%	3600
X13	69230M	2,61%	17	1,30%	1,56%	411	1,09%	0,22%	3600
X14	68395M	3,09%	25	1,22%	0,81%	267	0,89%	0,04%	3600
X15	66028M	3,60%	13,4	0,47%	0,45%	100	0,10%	0,00%	3600

TABLE 6.2 – Structure de convergence avec Cplex 12.5

6.1.3 Résultats de la formulation avec stretch

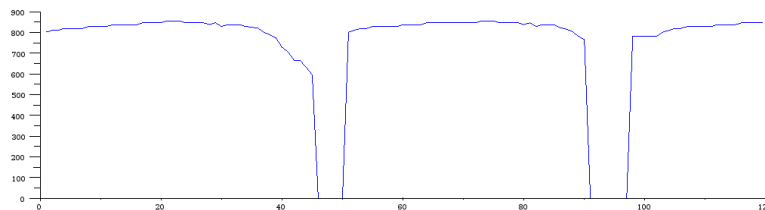


FIGURE 6.1 – Solution de production avec contraintes de stretch

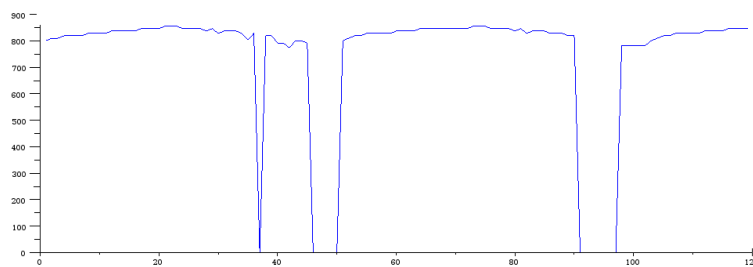


FIGURE 6.2 – Solution de production sans contraintes de stretch

FIGURE 6.3 – Impact des contraintes de stretch

La première question était la pertinence des contraintes de stretch proposée, en imposant uniquement des bornes supérieures. Si notre modélisation n'interdit pas des modulations sur les périodes de stretch, ces situations ne sont pas apparues. La production nucléaire étant économiquement plus rentable, les modulations étaient en fait assez rares. En fait, les situations où le stretch était activé se trouvaient assez rares, quand c'était inévitable de par les contraintes de fenêtres de temps, l'optimum économique privilégiant de ne pas franchir le bore nul pour ne pas avoir de pertes de puissances dues au stretch.

Avec la formulation générale du stretch, aucun calcul n'était réalisable sur les instances de taille réelle, générant des dépassements de mémoire, contrairement aux variantes de formulations du stretch. La combinatoire des contraintes ajoutées s'avérait alors réhibitoire (4 fois plus élevé que la formulation sans stretch,

rapport peu amélioré par le preprocessing de Cplex). Il se posait alors la question de l'incidence des contraintes de stretch sur les dates d'arrêts optimales. En calculant à partir des dates d'arrêt optimales d'une formulation simplifiée du stretch, leur coût associé dans la formulation complète, cela permet de chiffrer la dégradation de la qualité des solutions. Dans le cas de la formulation sans stretch, un faible surcoût (0,3 % environ) est observé après projection. De plus, de petites améliorations locales (formulations issues de la partie 3) de ces solutions engendrent des solutions très proches de l'optimum avec stretch. En terme de qualité de relaxation continue, l'ajout de contraintes de stretch ne fournit pas d'amélioration significative, pour des temps de calculs bien plus élevés. Ces constats permettent de justifier l'hypothèse simplificatrice réalisée dans [91] de négliger les contraintes de stretch dans un premiers temps. La suite de la thèse se focalisera ainsi sur la formulation sans contraintes CT6.

	valeur	Sans stretch projeté	+VNS
A1.3.120	76,12310958	0,01%	0,00%
A2.3.120	72,13777662	0,01%	0,00%
A3.3.120	76,387054383	0,02%	0,00%
A4.3.120	51,228177864	0,05%	0,00%
A5.3.120	58,523291982	0,08%	0,03%
B6.3.120	31,950404108	0,10%	0,00%
B7.3.120	30,364564916	0,18%	0,06%
B8.3.120	29,438125651	0,14%	0,13%
B9.3.120	29,238678231	0,18%	0,00%
B10.3.120	27,956624692	0,34%	0,06%
X11.3.120	30,198904154	0,20%	0,00%
X12.3.120	29,668888103	0,25%	0,00%
X13.3.120	29,1479638997	0,36%	0,00%
X14.3.120	28,438386747	0,23%	0,00%
X15.3.120	27,721548326	0,37%	0,00%

TABLE 6.3 – Projection des coûts des solutions optimales sans stretch en considérant le stretch

6.1.4 Influence de la solution initiale

Instances	Toutes var	Frontal	warmstart	penal
A1.3.120	0,12	0,12	0,04	0,13
A2.3.120	0,31	0,29	0,27	0,58
A3.3.120	0,23	0,21	0,21	0,14
A4.3.120	1,55	0,89	0,45	0,39
A5.3.120	127	82	70,86	15,3
Sous Total A	129,21	83,51	71,83	16,54
B6.3.120	32,3	17	13,8	4,98
B7.3.120	916,9	512	437,9	20,1
B8.3.120	3600	1483	1461	45
B9.3.120	3600	3600	3600	264
B10.3.120	60,8	52	43,8	5,4
Sous Total B	8210	5664	5556,5	339,48
X11.3.120	60,5	40,7	21,75	5,7
X12.3.120	111	69,4	43,6	17,2
X13.3.120	27,6	22	16,2	5,98
X14.3.120	98,8	56,1	53,2	16,3
X15.3.120	79,3	14	8,7	15,4
Sous Total X	377,2	202,2	143,45	60,58
Total	8716,41	5949,71	5771,78	416,6

TABLE 6.4 – Tableau des temps de calculs de la formulation sans stretch

La solution initiale utilisée provient de la section 7.1.2, de qualité moyenne et pas forcément réalisable. L'introduction d'une pénalisation dans la fonction objectif s'est avérée bénéfique pour les temps de calculs, comme le montre le tableau 6.4. Les temps de calculs sont d'autant plus bas que la pénalisation est forte. La pénalisation à la solution initiale casse en effet la structure pénalisante pour les branchements où beaucoup de solutions ont des coûts proches, ne permettant pas de couper facilement des branches de l'arbre de Branch&Bound.

Une première application est de réparer une solution : en fournissant au modèle une solution initiale potentiellement infaisable avec une pénalisation forte, on obtient un calcul cherchant la solution réalisable la plus proche, qui présente des analogies avec l'heuristique "feasibility pump". Nous avons ainsi utilisé

ce modèle pour réparer des solutions heuristiques constructives ne garantissant pas de respecter toutes les contraintes. L'accélération de convergence obtenue permettait d'obtenir des solutions pour toutes les instances, y compris les instances B8, B9 les plus difficiles.

Une autre application est de faire varier les pondérations de la fonction objectif, et de tracer des courbes de Pareto arbitrant le coût de la solution sans pénalisation à l'écart à une solution initiale donnée. De telles courbes sont potentiellement intéressantes à l'opérationnel. Pour un faible surcoût sur la fonction objectif, des plannings nécessitant peu de réorganisation sont préférables. Dans notre implémentation, les pénalisations étaient identiques pour tous les arrêts. Distinguer des catégories de pénalisations selon les arrêts, indiqués par l'opérateur répondrait plus au besoin opérationnel.

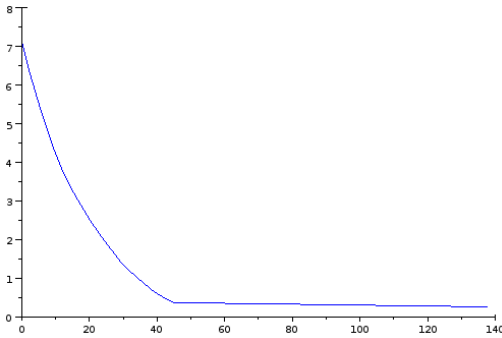


FIGURE 6.4 – Instance B7-3-120

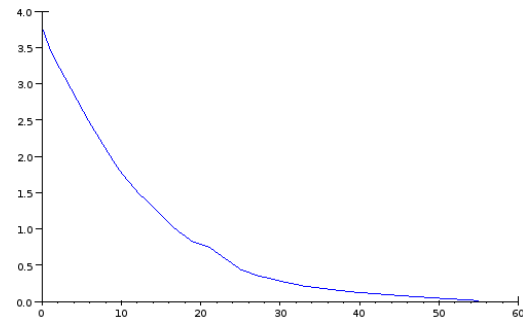


FIGURE 6.5 – Instance X11-3-120

FIGURE 6.6 – Courbes de Pareto coûts de solution/écart à la solution initiale

6.1.5 Bilan des résultats de formulation frontale

A ce stade, l'analyse de la résolution exacte nous a fourni des enseignements sur la convergence. Le preprocessing manuel des fenêtres de temps s'avère primordial pour une résolution efficace, se complétant avec la détection de structures et les algorithmes sophistiqués de Cplex. Les temps de relaxation continues s'avèrent critiques même sans les contraintes CT6 et CT12, posant des difficultés en 1h sur les instances B8 et B9 à plus fortes combinatoires, et ne permettant pas d'aboutir à des solutions primales de coût acceptable. La structure de convergence des branchements est exploitable opérationnellement pour les combinatoires des instances autres que B8 et B9. L'ajout de stretch impacte fortement les temps et capacités de résolution, pour une faible plus value sur les qualités de relaxation continue ou en surcoût de la qualités des solutions de dates d'arrêts. Ces constats permettent de justifier l'hypothèse simplificatrice réalisée dans [91] de négliger les contraintes de stretch CT6. Nous notons également un fort impact à considérer une solution initiale dans la capacité de résolution, en supprimant des symétries et pseudo-symétries (solutions de coûts très proches vu du Branch&Bound).

6.2 Bornes duales pour le challenge ROADEF

Cette section a pour but de dériver du travail des chapitres précédents pour calculer des bornes duales du challenge ROADEF, à comparer avec les meilleures de la littérature publiées dans [34]. La modélisation du chapitre 5 procédait par conditions nécessaires en partant des contraintes du challenge ROADEF, sans simplifications autres que des relaxations. Lorsqu'on omet les contraintes CT6 et CT12, et en ne considérant pas les extensions de solutions initiales et de longueur minimales et maximales de cycles, toute borne duale obtenue fournit une borne duale du problème du challenge ROADEF. Face aux limites de la résolution exacte en terme de capacités et temps de calcul précédemment mises en évidence, amenant à restreindre

des modèles agrégés pour les pas de temps de production à la semaine, et sur un scénario déterministe. Dans cette section, nous allons prouver que l'on peut calculer des bornes du challenge ROADEF en se ramenant à des calculs de cette taille. Par facilité d'écriture pour les preuves, nous écrirons la formulation exacte sans les contraintes CT6 et CT12 suivant la formulation vante, en définissant $x_{i,k,s}^{init}$ et $x_{i,k,s}^{fin}$ comme des variables :

$$v_0 = \min \quad \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,s,t} \pi_s \mathbf{C}_{j,t}^{prd} \mathbf{D}^t p_{j,s,t} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \quad (6.1)$$

$$M^{ordo} d \geq b^{ordo} \quad (6.2)$$

$$\mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \quad (6.3)$$

$$\forall i, s, \quad x_{i,0,s}^{init} = \mathbf{Xi}_i \quad (6.4)$$

$$\forall i, k, s, \quad x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \quad (6.5)$$

$$\forall i, k, s, \quad x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \mathbf{D}^t p_{i,k,s,t} \quad (6.6)$$

$$\forall i, k, s, \quad x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \quad (6.7)$$

$$\forall i, k, s, \quad x_{i,k,s}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k}-1}{\mathbf{Q}_{i,k}} (x_{i,k-1}^{fin} - \mathbf{Bo}_{i,k-1}) \quad (6.8)$$

$$\forall i, k, s, \quad x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \quad (6.9)$$

$$\forall j, s, t, \quad \mathbf{Pmin}_{j,t}^s \leq p_{j,s,t} \leq \mathbf{Pmax}_{j,t}^s \quad (6.10)$$

$$\forall i, k, s, t, \quad p_{i,k,s,t} \leq \mathbf{Pmax}_{i,t} (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \quad (6.11)$$

$$\forall s, t, \quad \sum_{i,k} p_{i,k,s,t} + \sum_j p_{j,s,t} = \mathbf{Dem}^{t,s} \quad (6.12)$$

$$d \in \{0, 1\}^N, r, p, x \geq 0 \quad (6.13)$$

v_0 est bien une borne inférieure du problème ROADEF, toute solution du challenge ROADEF vérifie ces contraintes, le coût d'une solution optimale ROADEF est donc supérieur à ce problème de minimisation comme les fonctions objectifs sont identiques. Le preprocessing exact est toujours valide pour les solutions du challenge ROADEF, il est conservé dans la suite.

6.2.1 Bornes duales par agrégation des pas de temps

À présent, nous prouvons que le problème obtenu par agrégation des pas de temps à la semaine donne des bornes inférieures du problème désagrégé. On définit $\bar{D}^w = \sum_{t,w_t=w} \mathbf{D}^t$, $\overline{Dem}^{t,s} = \sum_{t,w_t=w} \frac{\mathbf{D}^t}{\bar{D}^w} \mathbf{Dem}^{t,s}$, $\overline{Pmin}_{j,w}^s =$

$$\sum_{t,w_t=w} \frac{\mathbf{D}^t}{\bar{D}^w} \mathbf{Pmin}_{j,t}^s, \overline{Pmax}_{j,w}^s = \sum_{t,w_t=w} \frac{\mathbf{D}^t}{\bar{D}^w} \mathbf{Pmax}_{j,t}^s.$$

$$v_1 = \min \quad \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,s,t} \pi_s \mathbf{C}_{j,w}^{prd} \overline{D}^w P_{j,s,w} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \quad (6.14)$$

$$M^{ordo} d \geq b^{ordo} \quad (6.15)$$

$$\mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \quad (6.16)$$

$$\forall i, s, \quad x_{i,0,s}^{init} = \mathbf{Xi}_i \quad (6.17)$$

$$\forall i, k, s, \quad x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \quad (6.18)$$

$$\forall i, w, s, \quad x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \overline{D}^w P_{i,k,s,w} \quad (6.19)$$

$$\forall i, k, \quad x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \quad (6.20)$$

$$\forall i, k, s, \quad x_{i,k,s}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1,s}^{fin} - \mathbf{Bo}_{i,k-1}) \quad (6.21)$$

$$\forall i, k, s, \quad x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \quad (6.22)$$

$$\forall j, s, w, \quad \overline{Pmin}_{j,w}^s \leq p_{j,s,w} \leq \overline{Pmax}_{j,w}^s \quad (6.23)$$

$$\forall i, s, w, \quad P_{i,k,s,w} \leq \overline{Pmax}_{j,w}^s (d_{i,k,w} - \mathbf{Da}_{i,k} - d_{i,k+1,w}) \quad (6.24)$$

$$\forall s, w, \quad \sum_{i,k} P_{i,k,s,w} + \sum_j P_{j,s,w} = \overline{Dem}^{w,s} \quad (6.25)$$

$$d \in \{0, 1\}^N, r, p, x \geq 0 \quad (6.26)$$

Lemme 6.2.1 *L'agrégation des pas de temps de production à la semaine donne une borne inférieure du problème désagrégé. Mathématiquement, $v_1 \leq v_0$.*

En particulier, toute borne inférieure de v_1 est une borne inférieure du problème du challenge ROADEF.

Preuve : Soit (d^*, r^*, p^*, x^*) une solution optimale du PLNE définissant v_0 . On définit :

$$P_{j,s,w}^* = \sum_{t, w_t=w} \frac{\mathbf{D}^t}{\overline{D}^w} p_{j,s,t}^*, \quad P_{i,k,s,w}^* = \sum_{t, w_t=w} \frac{\mathbf{D}^t}{\overline{D}^w} p_{i,k,s,t}^*. \quad \text{Montrons que } (d^*, r^*, P^*, x^*) \text{ est alors une solution}$$

réalisable du PLNE définissant v_1 . Les équations (6.15), (6.16), (6.17), (6.18), (6.20), (6.21), (6.22) sont trivialement vérifiées comme inchangées. Les autres contraintes s'obtiennent en regroupant les pas de temps t suivant leur semaine de réalisation w_t .

$$x_{i,k,s}^{fin*} = x_{i,k,s}^{init*} - \sum_t \mathbf{D}^t p_{i,k,s,t}^* = x_{i,k,s}^{init*} - \sum_w \sum_{t:w_t=w} \mathbf{D}^t p_{i,k,s,t}^* = x_{i,k,s}^{fin*} = x_{i,k,s}^{init*} - \sum_w \overline{D}^w P_{i,k,s,w}^*,$$

(6.19) est ainsi prouvée. Les équations (6.23), (6.24), (6.25) s'obtiennent pour tout w en additionnant les contraintes pour tout t tel que $w_t = w$. Toutes les équations définissant v_1 sont vérifiées, (d^*, r^*, P^*, x^*) est une solution réalisable du PLNE définissant v_1 . La valeur de l'objectif est donc minimisée par l'optimum

$$v_1 : v_1 \leq \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k}^* + \sum_{j,s,t} \pi_s \mathbf{C}_{j,w}^{prd} \overline{D}^w P_{j,s,w}^* - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin*} = v_0$$

6.2.2 Bornes duales en décomposant par scénarios

A la suite de la section précédente, on a pu justifier que la réduction du problème sur pas de temps de production agrégés à la semaine donnait toujours des bornes inférieures du challenge ROADEF. L'objet de cette section est de réduire le nombre de scénarios, autre point limitant dans la résolution. Dans un premier temps, on se ramènera à un calcul de bornes basé sur des problèmes déterministes, avant de généraliser en considérant des résolutions frontales pour un faible nombre de scénarios. On définit pour tout scénario $s \in \mathcal{S}$, le problème décisionnel où les décisions sont basées uniquement sur le scénario s :

$$\begin{aligned}
 v_s^{det} = \min \quad & \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,t} \mathbf{C}_{j,w}^{prd} \bar{D}^w P_{j,w} - \sum_i \mathbf{C}_i^{val} x_i^{fin} \\
 & M^{ordo} d \geq b^{ordo} \\
 \forall i, k, \quad & \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \\
 \forall i, \quad & x_{i,0}^{init} = \mathbf{Xi}_i \\
 \forall i, k, \quad & x_{i,k}^{init} \leq \mathbf{Smax}_{i,k} \\
 \forall i, w, \quad & x_{i,k}^{fin} = x_{i,k}^{init} - \sum_t \bar{D}^w P_{i,k,w} \\
 \forall i, k, \quad & x_{i,k}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
 \forall i, k, \quad & x_{i,k}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1}^{fin} - \mathbf{Bo}_{i,k-1}) \\
 \forall i, k, \quad & x_i^{fin} \leq x_{i,k}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \\
 \forall j, w, \quad & \mathbf{Pmin}_{j,w}^s \leq P_{j,w} \leq \mathbf{Pmax}_{j,w}^s \\
 \forall i, w, \quad & P_{i,k,w} \leq \mathbf{Pmax}_{i,t} (d_{i,k,w} - \mathbf{Da}_{i,k} - d_{i,k+1,w}) \\
 \forall w, \quad & \sum_{i,k} P_{i,k,w} + \sum_j P_{j,w} = \overline{Dem}^{w,s} \\
 & d \in \{0, 1\}^N, r, P, x \geq 0
 \end{aligned}$$

Proposition 6.2.1 *On a $\sum_{s \in \mathcal{S}} \pi_s v_s^{det} \leq v_1 \leq v_0$. Autrement dit, on peut calculer des bornes inférieures du problème ROADEF en se ramenant à S calculs de bornes inférieures de types déterministes sur les différents scénarios, avec pas de temps agrégés à la semaine.*

Preuve : On réécrit le problème en dupliquant des variables $d^s = d$, $r^s = r$, $x_{i,s}^{fin} = x_i^{fin}$ et en utilisant $\sum_{s \in \mathcal{S}} \pi_s = 1$:

$$\begin{aligned}
 v_1 = \min \quad & \sum_{i,k} \sum_s \pi_s \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,s,t} \pi_s \mathbf{C}_{j,w}^{prd} \bar{D}^w P_{j,s,w} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
 \forall s, \quad & d^s = d \\
 \forall s, \quad & r^s = r \\
 \forall i, s, \quad & x_{i,s}^{fin} = x_i^{fin} \\
 \forall s, \quad & M^{ordo} d^s \geq b^{ordo} \\
 \forall i, k, s, \quad & \mathbf{Rmin}_{i,k} d_{i,k,W}^s \leq r_{i,k}^s \leq \mathbf{Rmax}_{i,k} d_{i,k,W}^s \\
 \forall i, s, \quad & x_{i,0,s}^{init} = \mathbf{Xi}_i \\
 \forall i, k, s, \quad & x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \\
 \forall i, w, s, \quad & x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \bar{D}^w P_{i,k,s,w} \\
 \forall i, k, s, \quad & x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
 \forall i, k, s, \quad & x_{i,k,s}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1,s}^{fin} - \mathbf{Bo}_{i,k-1}) \\
 \forall i, k, s, \quad & x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \\
 \forall j, s, w, \quad & \overline{\mathbf{Pmin}}_{j,w}^s \leq P_{j,s,w} \leq \overline{\mathbf{Pmax}}_{j,w}^s \\
 \forall i, s, t, \quad & 0 \leq P_{i,k,s,t} \leq \mathbf{Pmax}_{i,t} (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \\
 \forall s, w, \quad & \sum_{i,k} P_{i,k,s,w} + \sum_j P_{j,s,w} = \overline{Dem}^{w,s} \\
 & d \in \{0, 1\}^N, r, P, x \geq 0
 \end{aligned}$$

En relâchant les contraintes $x_{i,s}^{fin} = x_i^{fin}$, $d^s = d$ et $r^s = r$, on obtient une borne inférieure de v_1 .

$$\begin{aligned}
v_1 \geq \min \quad & \sum_{i,k} \sum_s \pi_s \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,s,t} \pi_s \mathbf{C}_{j,w}^{prd} \overline{D}^w P_{j,s,w} - \sum_{i,s} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
& \forall s, \quad \text{Mordo } d^s \geq b^{ordo} \\
& \forall i, k, s, \quad \mathbf{Rmin}_{i,k} d_{i,k,W}^s \leq r_{i,k}^s \leq \mathbf{Rmax}_{i,k} d_{i,k,W}^s \\
& \forall i, s, \quad x_{i,0,s}^{init} = \mathbf{Xi}_i \\
& \forall i, k, s, \quad x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \\
& \forall i, w, s, \quad x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \overline{D}^w P_{i,k,s,w} \\
& \forall i, k, s, \quad x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
& \forall i, k, s, \quad x_{i,k,s}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1,s}^{fin} - \mathbf{Bo}_{i,k-1}) \\
& \forall i, k, s, \quad x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \\
& \forall j, s, w, \quad \overline{Pmin}_{j,w}^s \leq P_{j,s,w} \leq \overline{Pmax}_{j,w}^s \\
& \forall i, s, t, \quad 0 \leq P_{i,k,s,t} \leq \mathbf{Pmax}_{i,t} (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \\
& \forall s, w, \quad \sum_{i,k} P_{i,k,s,w} + \sum_j P_{j,s,w} = \overline{Dem}^{w,s} \\
& d \in \{0, 1\}^N, r, P, x^{fin} \geq 0
\end{aligned}$$

Ce dernier problème de minimisation se décompose alors blocs par blocs selon les scénarios, la valeur obtenue pour chaque scénario étant exactement $\pi_s v_s^{det}$. On remarque que la preuve précédente peut se généraliser en considérant plusieurs scénarios (en faible nombre de manière à garantir la calculabilité de bornes inférieures sur le nombre restreint de scénarios). Soit $S^0 \subset \mathcal{S}$. On note $v_2(S^0)$ le problème stochastique suivant restreint aux scénarios $s \in S^0$:

$$\begin{aligned}
v_2(S^0) = \min \quad & \sum_{i,k} \sum_{s \in S^0} \pi_s \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{s \in S^0} \sum_{j,t} \pi_s \mathbf{C}_{j,w}^{prd} \overline{D}^w P_{j,s,w} - \sum_i \sum_{s \in S^0} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
& \text{Mordo } d \geq b^{ordo} \\
& \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \\
& \forall i, s \in S^0, \quad x_{i,0,s}^{init} = \mathbf{Xi}_i \\
& \forall i, k, s \in S^0, \quad x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \\
& \forall i, w, s \in S^0, \quad x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \overline{D}^w P_{i,k,s,w} \\
& \forall i, k, s \in S^0, \quad x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
& \forall i, k, s \in S^0, \quad x_{i,k,s}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1,s}^{fin} - \mathbf{Bo}_{i,k-1}) \\
& \forall i, k, s \in S^0, \quad x_{i,s}^{fin} \leq x_{i,k,s}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \\
& \forall w, j, s \in S^0, \quad \mathbf{Pmin}_{j,t}^s \leq P_{j,s,w} \leq \overline{Pmax}_{j,w}^s \\
& \forall w, i, s \in S^0, \quad P_{i,k,s,w} \leq \overline{Pmax}_{j,w}^s (d_{i,k,w} - \mathbf{Da}_{i,k} - d_{i,k+1,w}) \\
& \forall w, s \in S^0, \quad \sum_{i,k} P_{i,k,s,w} + \sum_j P_{j,s,w} = \overline{Dem}^{w,s} \\
& d \in \{0, 1\}^N, r, p, x \geq 0
\end{aligned}$$

On peut calculer des bornes inférieures du problème ROADEF en se ramenant à des calculs de bornes inférieures avec un nombre restreint de scénarios, avec pas de temps agrégés à la semaine :

Proposition 6.2.2 Soit S_i pour $i \in \mathcal{I}$ une partition de \mathcal{S} , ie $\mathcal{S} = \bigcup_{i \in \mathcal{I}} S_i$ et $\bigcap_{i \in \mathcal{I}} S_i = \emptyset$. On a :

$$\sum_{s \in \mathcal{S}} \pi_s v_s^{det} \leq \sum_{i \in \mathcal{I}} v_2(S_i) \leq v_1 \leq v_0 \quad (6.27)$$

La preuve de $\sum_{i \in I} v_2(S_i) \leq v_1$ est identique au cas déterministe de la section précédente, en opérant la relaxation des contraintes $x_{i,s}^{fin} = x_{i,s'}^{fin}$, $d^s = d^{s'}$ et $r^s = r^{s'}$ pour s et s' dans un ensemble différent de la partition de \mathcal{S} . La décomposition scénario par scénario est en fait le cas particulier où la partition des scénarios se fait suivant les singletons de scénarios, $\mathcal{S} = \bigcup_{s \in \mathcal{S}} \{s\}$, ce qui s'obtient par une nouvelle relaxation, en considérant toutes les relaxations considérées à la section précédente. En pratique, on considérera les partitions en prenant un nombre fixe de scénarios et en les considérant successivement selon leur ordre d'indexation. Un tel résultat permet d'avoir un ordre de grandeur des pertes engendrées par la relaxation scénario par scénarios, en fixant les décisions r et d sur un unique scénario.

6.2.3 Bornes inférieures par relaxation de contraintes

À présent, le calcul de bornes du problème ROADEF peut se ramener à des calculs de bornes inférieures de types déterministes sur les différents scénarios, avec pas de temps agrégés à la semaine, on dispose donc de bornes calculables. Pour accélérer les temps de calculs de bornes ou dépasser les limites en terme de nombre de scénarios simultanément considérés dans les sous problèmes d'optimisation, on peut envisager de calculer des bornes sur des relaxations plus faciles à calculer que des problèmes déterministes, avec significativement moins de variables et de contraintes. Relâcher des contraintes comme des contraintes d'ordonnancement CT14 à CT21, relâcher les contraintes d'anticipation maximales fournit toujours des bornes, sans diminuer significativement la combinatoire des variables et des contraintes. On envisage ici des relaxations plus importantes, convergeant plus facilement vers des bornes inférieures de moins bonne qualité que les bornes obtenues avec la formulation complète. Comme pour les heuristiques primales de fixation de variables, le choix se fera selon le compromis entre dégradation de la qualité des bornes et l'accélération de la convergence et les stades de convergence qu'il est possible d'atteindre.

Relaxation des arrêts Une première simplification drastique revient à négliger complètement les arrêts des centrales nucléaires, en se disant que le coût de la solution est majoritairement le résultat d'un grand Unit Commitment Problem où il faut juste pouvoir minorer le coût de la production nucléaire. La formulation définissant $v_3(S^0)$ pour tout $S^0 \subset \mathcal{S}$ se passe ainsi des variables de stocks autres que les $x_{i,s}^{fin}$, et n'utilise comme variables binaires que la famille $d_{i,k}$ correspondant aux variables précédentes $d_{i,k,W}$ pour chiffrer le nombre et les quantités de rechargement effectués, directement liés aux coûts de productions nucléaires.

$$v_3(S^0) = \min \sum_{i,k} \sum_{s \in S^0} \pi_s \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{s \in S^0} \sum_{j,t} \pi_s \mathbf{C}_{j,w}^{prd} \overline{D}^w P_{j,s,w} - \sum_i \sum_{s \in S^0} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \quad (6.28)$$

$$\forall i, k, \quad d_{i,k+1} \leq d_{i,k} \quad (6.29)$$

$$\forall i, k, \quad \mathbf{Rmin}_{i,k} d_{i,k} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k} \quad (6.30)$$

$$\forall w, j, s \in S^0, \quad \overline{Pmin}_{j,w}^s \leq p_{j,s,w} \leq \overline{Pmax}_{j,w}^s \quad (6.31)$$

$$\forall w, s \in S^0, \quad \sum_i p_{i,s,w} + \sum_j p_{j,s,w} = \overline{Dem}^{w,s} \quad (6.32)$$

$$\forall i, s \in S^0, \quad x_{i,s}^{fin} \leq \mathbf{Bo}_{i,k} - \mathbf{Bo}_{i,0} + \mathbf{Xi}_i + \sum_k r_{i,k} - \sum_w \mathbf{D}^w p_{i,s,w} \quad (6.33)$$

$$\forall i, k, s \in S^0, \quad x_{i,s}^{fin} \leq \mathbf{Smax}_{i,k} (d_{i,k} - d_{i,k+1}) \quad (6.34)$$

Proposition 6.2.3 Soit $S^0 \subset \mathcal{S}$, $v_3(S^0) \leq v_2(S^0)$.

Preuve : $S^0 \subset \mathcal{S}$, soit $(d_{i,k}^*, r_{i,k}^*, p_{i,k,s,w}^*, p_{j,s,w}^*, x^*)$ une solution optimale du PLNE définissant $v_2(S^0)$. Montrons que $(d_{i,k,W}^*, r_{i,k}^*, \sum_k p_{i,k,s,w}^*, p_{j,s,w}^*, x^{fin*})$ est alors une solution réalisable du PLNE définissant $v_3(S^0)$. Seules les équations (6.33) ne sont pas trivialement vérifiées, elles résultent du fait que $\frac{\mathbf{Q}_{i,k}-1}{\mathbf{Q}_{i,k}} \leq 1$ en additionnant de proche en proche les inégalités obtenues à chaque rechargement.

Formulation paramétrique Pour améliorer la qualité de la précédente relaxation, on envisage une formulation mixte paramétrique : pour des arrêts inférieurs à l'indice k^0 , on prend la modélisation exacte, la relaxation des arrêts portant sur les arrêts postérieurs, le niveau de puissance nucléaire étant alors donné par $p_{i,k^0,w}$. La contrainte de positivité des stocks finaux est valable pour les cycles d'indice inférieurs à k^0 . Cela donne la formulation suivante où les contraintes $M_{k^0}^{ordo} d \geq b_{k^0}^{ordo}$ regroupent les contraintes d'ordonnement tronquées en ne considérant que les variables d'indice $k \leq k^0$:

$$\begin{aligned}
v_3(S^0, k^0) = \min & \sum_{i,k} \sum_{s \in S^0} \pi_s \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{s \in S^0} \sum_{j,t} \pi_s \mathbf{C}_{j,w}^{prd} \bar{D}^w P_{j,s,w} - \sum_i \sum_{s \in S^0} \pi_s \mathbf{C}_i^{val} x_{i,s}^{fin} \\
& \forall i, k, & d_{i,k+1} \leq d_{i,k} \\
& \forall i, k \leq k^0, & d_{i,k} = d_{i,k,W} \\
& & M_{k^0}^{ordo} d \geq b_{k^0}^{ordo} \\
& \forall i, k, & \mathbf{Rmin}_{i,k} d_{i,k} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k} \\
& \forall i, s \in S^0, & x_{i,0,s}^{init} = \mathbf{Xi}_i \\
& \forall i, k \leq k^0, s \in S^0, & x_{i,k,s}^{init} \leq \mathbf{Smax}_{i,k} \\
& \forall i, k \leq k^0, w, s \in S^0, & x_{i,k,s}^{fin} = x_{i,k,s}^{init} - \sum_t \bar{D}^w p_{i,k,s,w} \\
& \forall i, k < k^0, s \in S^0, & x_{i,k,s}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
& \forall i, k \leq k^0, s \in S^0, & x_{i,k,s}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1,s}^{fin} - \mathbf{Bo}_{i,k-1}) \\
& \forall i, k \leq k^0, s \in S^0, & x_{i,k,s}^{fin} \leq x_{i,k,s}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \\
& \forall w, j, s \in S^0, & \overline{Pmin}_{j,w}^s \leq p_{j,s,w} \leq \overline{Pmax}_{j,w}^s \\
& \forall w, i, k \leq k^0, s \in S^0, & p_{i,k,s,w} \leq \overline{Pmax}_{j,w}^s (d_{i,k,w} - \mathbf{Da}_{i,k} - d_{i,k+1,w}) \\
& \forall w, s \in S^0, & \sum_{i,k \leq k^0} p_{i,k,s,w} + \sum_j p_{j,s,w} = \overline{Dem}^{w,s} \\
& \forall i, k \leq k^0, s \in S^0, & x_{i,s}^{fin} \leq \mathbf{Bo}_{i,k} - \mathbf{Bo}_{i,0} + x_{i,k,s}^{init} - \sum_w \mathbf{D}^w p_{i,k,w} + \mathbf{Smax}_{i,k} (1 + d_{i,k+1} - d_{i,k}) \\
& \forall i, s \in S^0, & x_{i,s}^{fin} \leq x_{i,k^0,s}^{init} + \sum_{k > k^0} r_{i,k} - \sum_w \mathbf{D}^w p_{i,k^0,w} + \mathbf{Smax}_{i,k} (1 - d_{i,k^0+1}) \\
& & d \in \{0, 1\}^N, r, p, x \geq 0
\end{aligned}$$

6.2.4 Résultats

Nous comparons les bornes obtenues suivant différentes approches avec les seules bornes publiées dans la littérature dans [35], où deux types de bornes sont publiées. Nous noterons **Brandt1** leurs bornes obtenues en temps de calculs courts, et **Brandt2** leurs bornes obtenues en s'autorisant de longs temps de calculs. L'implémentation a été réalisée avec OPL, en utilisant Cplex 12.5, avec des temps de calculs limités à 1h pour les calculs par scénarios. Au vu du temps de calcul nécessaire, plusieurs ordinateurs portables ont été utilisés. Les calculs les plus critiques ont été réalisés avec un ordinateur portable de processeur Intel Core2 Duo processor, 2.80GHz, avec Linux Ubuntu 12.04, pour les calculs plus petits où la convergence était assurée, des ordinateurs portables moins puissants ont été utilisés. Le preprocessing exact du chapitre 6 était également utilisé sur les fenêtres de temps, facilitant grandement les calculs. Les contraintes CT19 ont également été relâchées dans nos tests, ce qui est sans doute peu impactant en terme de bornes, cela concernait au plus une seule contrainte d'ordonnement comme le montre le tableau 8.2. Notons que des tests ont été fait sur les petites instances en incorporant les contraintes de bornes supérieures de productions dues au stretch, leur impact était quasi nul sur la qualité des bornes obtenues, pour une dégradation nette des temps et capacités de calculs.

Bornes duales obtenues sur des temps de calculs très rapides Le tableau suivant compare les bornes obtenues par notre approche en temps très rapides, en considérant la formulation paramétrique de la section 7.3.2 avec comme paramètre maximal $k^0 = 2$, en considérant un seul scénario dans les sous problèmes. La résolution d'un sous problème était instantanée pour $k^0 = 0$, de quelques secondes pour $k^0 = 1$ sauf sur B8, B9 où la convergence exacte était atteinte en une minute, et pour $k^0 = 2$, la convergence était complète en 1 minute pour toutes les instances sauf sur B8, B9 où le calcul était tronqué à 600s. A ce stade, on observe déjà une amélioration des bornes de [35].

Instances	Primal	Brandt1	Brandt2	$k^0 = 0$	$k^0 = 1$	$k^0 = 2$
A1	169538	5,13%	2,35%	2,22%	1,73%	1,48%
A2	146048	10,89%	4,15%	3,87%	3,09%	2,4%
A3	154429	10,8%	3,87%	4,1%	3,16%	2,4%
A4	111591	26,13%	8,3%	8,54%	6,62%	4,65%
A5	125822	24,99%	10,61%	10,33%	8,24%	6,83%
Total A	707428	14,4%	5,46%	5,41%	4,25%	3,32%
B6	83424	56,44%	16,58%	19,18%	15,12%	11,68%
B7	81174	52,86%	15,58%	16,31%	13,73%	11,18%
B8	81926	65,32%	23,6%	23,82%	20,21%	17,06%
B9	81750	63,3%	21,72%	22,51%	18,74%	15,45%
B10	77767	60,92%	18,03%	19,58%	16,71%	13,43%
Total B	315586	59,76%	19,11%	20,29%	16,9%	13,76%
X11	79116	57,81%	15,4%	15,27%	13,12%	11,07%
X12	77589	52,65%	14,22%	14,79%	12,43%	10,32%
X13	76449	66,27%	18,7%	18,56%	16,12%	14,11%
X14	76172	64,68%	17,23%	17,63%	15,18%	13,11%
X15	75101	62,13%	17,62%	19,02%	16,5%	13,5%
Total X	384427	60,65%	16,61%	17,03%	14,65%	12,4%

TABLE 6.5 – Bornes duales obtenues rapidement pour le problème du challenge ROADEF

Bornes scénarios par scénarios En limitant la convergence à 1h pour les calculs de bornes pour chaque scénario, les bornes précédentes ont pu être améliorées nettement. La formulation paramétrique de la section 7.3.2 avec comme paramètre $k^0 = 5$ permettait une légère dégradation de la borne duale de convergence, pour une nette accélération de la convergence. Ainsi, sur les instances les plus difficiles (B7, B8, B9), les valeurs obtenues en 1h étaient bien meilleures avec la formulation paramétrique qu'en considérant la formulation exacte. Un tel résultat est le pendant en heuristiques duales des fixations de variables sur les heuristiques primales du chapitre 4, sur des temps de calculs définis, il peut être rentable de considérer une formulation avec moins de variables convergeant vers une valeur de moins bonne qualité que la formulation exacte, mais en convergeant plus rapidement.

Instances	Primal	Brandt1	Brandt2	k=4	k=5	PL+cuts	PLNE
A1	169538	5,13%	2,35%	0,89%	0,79%	0,8%	0,75%
A2	146048	10,89%	4,15%	1,13%	0,84%	0,82%	0,73%
A3	154429	10,8%	3,87%	1,1%	0,85%	0,94%	0,85%
A4	111591	26,13%	8,30%	2,2%	1,77%	2,2%	1,67%
A5	125822	24,99%	10,61%	3,99%	3,62%	3,91%	3,56%
Total A	707428	14,4%	5,46%	1,74%	1,47%	1,61%	1,41%
B6	83424	56,44%	16,58%	6,85%	5,36%	5,72%	5,3%
B7	81174	52,86%	15,58%	7,23%	6,31%	8,94%	7,64%
B8	81926	65,32%	23,60%	14,52%	12,39%	15,51%	15,51%
B9	81750	63,3%	21,72%	12,71%	11,65%	15,21%	15,21%
B10	77767	60,92%	18,03%	8,33%	7,32%	8,27%	7,71%
Total B	406041	59,76%	19,11%	9,94%	8,61%	10,74%	10,28%
X11	79116	57,81%	15,40%	6,45%	6,01%	5,86%	5,58%
X12	77589	52,65%	14,22%	7,42%	6,68%	7,49%	7,06%
X13	76449	66,27%	18,70%	10,17%	8,09%	8,12%	7,78%
X14	76172	64,68%	17,23%	9,64%	7,85%	8,79%	8,36%
X15	75101	62,13%	17,62%	10,02%	8,98%	9,67%	9,18%
Total X	384427	60,65%	16,61%	8,72%	7,5%	7,96%	7,57%
Total	1497896	38,57%	12,02%	5,75%	4,95%	5,71%	5,4%

TABLE 6.6 – Bornes duales obtenues scénarios par scénarios pour le problème du challenge ROADEF

Bornes en considérant des scénarios multiples En considérant plusieurs scénarios simultanément dans l'optimisation, on a pu observer de légères améliorations des bornes précédentes, suivant la section

7.2.2. En faisant augmenter les nombres de scénarios, on s’approchait des limites de résolution, et il s’avérait plus souvent rentable d’utiliser la formulation paramétrique de la section 7.3.2 avec comme paramètre $k^0 = 5$. Il est à noter que dans notre implémentation, on a considéré et regroupé les scénarios suivant leur ordre d’indexation. D’autres stratégies de partition seraient à tester, par exemple en triant les scénarios suivant les demandes cumulées (et les coûts escomptés), pour considérer des calculs intermédiaires où les scénarios considérés ont des demandes très différentes, menant a priori séparément à des solutions en d et r bien différentes.

Meilleurs résultats Le tableau 6.7 synthétise les meilleurs résultats de bornes obtenues sur les différentes instances, en précisant le mode de calcul menant à la meilleure borne duale. De nettes améliorations sont alors réalisées par rapport aux bornes duales fournies par [35].

Instances	Primal	Brandt2	Gap	These	Gap	k	nbScenar
A1	169538	165560	2,34%	168319	0,72%	∞	10
A2	146048	139991	4,15%	145073	0,67%	∞	20
A3	154429	148454	3,87%	153186	0,80%	∞	20
A4	111591	102326	8,30%	110029	1,40%	∞	10
A5	125822	112467	10,61%	122239	2,85%	5	15
Total A	707428	668798	5,46%	698845	1,21%		
B6	83424	69592	16,58%	79205	5,06%	5	3
B7	81174	68528	15,58%	76356	5,94%	5	3
B8	81926	62594	23,60%	72023	12,09%	5	3
B9	81750	63991	21,72%	72437	11,39%	5	3
B10	77767	63747	18,03%	72378	6,93%	5	5
Total B	406041	328452	19,11%	372399	8,29%		
X11	79116	66931	15,40%	74715	5,56%	∞	5
X12	77589	66558	14,22%	72768	6,21%	5	5
X13	76449	62155	18,70%	70840	7,34%	∞	3
X14	76172	63045	17,23%	70373	7,61%	∞	3
X15	75101	61866	17,62%	68700	8,52%	5	5
Total X	384427	320555	16,61%	357396	7,03%		
Total	1497896	1317805	12,02%	1428640	4,62%		

TABLE 6.7 – Meilleures bornes duales obtenues pour le problème du challenge ROADEF

6.3 Conclusions et perspectives

À ce stade, nous avons analysé les caractéristiques de la résolution frontale avec l’algorithme de Branch&Bound implémenté dans Cplex. Nous avons tout d’abord justifié l’importance d’un preprocessing manuel des fenêtres de temps pour compléter le preprocessing de Cplex. Même en omettant les contraintes de stretch (CT6 selon la dénomination du challenge ROADEF), la résolution s’avère limitée pour des raisons de trop grandes tailles de données. En réduisant les instances sur un unique scénario stochastique agrégé, et en agrégeant les pas de temps de production à la semaine, les temps de relaxation continues s’avèrent toujours critiques, posant des difficultés en 1h sur les instances B8 et B9 à plus fortes combinatoires.

La structure de convergence des branchements est exploitable opérationnellement pour les combinatoires des instances autres que B8 et B9, où une réduction des amplitudes de fenêtres de temps a été réalisées.

On est alors dans les hypothèses de la résolution d'EDF à l'opérationnel, dans le même cadre d'hypothèses qu'ORION et TANNAT au stretch près, après le preprocessing FUSCOU. Les progrès réalisés par les solveurs de PLNE depuis TANNAT permettent actuellement une résolution frontale efficace en 1h de temps de calcul.

Une considération opérationnelle qui justifiait ORION, était la volonté d'avoir des plannings proches de la solution courante calculée le mois précédent. Suivant ce besoin, nous avons proposé une modélisation de la pénalisation de l'écart à une solution initiale par la fonction objectif, sans rajouter de nouvelles variables, ou d'ajouter des contraintes de nombre maximal de modifications sans rajouter de nouvelles variables, alternative à la stabilisation par restriction des contraintes de fenêtres de temps. En termes de temps de calcul, cela facilite même la résolution frontale par Branch&Bound, en orientant la recherche autour de la solution courante, et en explorant moins de noeuds par Branch&Bound en coupant plus tôt des branches éloignées de la solution courante. Une application d'un tel modèle pouvait aussi être de tracer des courbes de Pareto pour arbitrer deux objectifs potentiellement contradictoires : optimiser le coût de la solution et minimiser l'écart à la solution courante. De telles courbes de Pareto peuvent appuyer une décision opérationnelle humaine, pour un choix de bon compromis. Pour chiffrer de telles considérations, les données réelles d'EDF et la donnée de la solution courante sont nécessaires.

L'ajout de stretch impacte fortement les temps et capacités de résolution, pour une faible plus value sur les qualités de relaxation continue ou en surcoût de la qualité des solutions de dates d'arrêts. Ces constats permettent de justifier l'hypothèse simplificatrice réalisée dans [91] de négliger les contraintes de stretch CT6 pour un modèle académique. A l'opérationnel, il s'agit d'une question ouverte à ce chapitre de savoir si l'ajout de nos contraintes simplifiées de stretch (qui sont en fait les mêmes que les contraintes ajoutées dans ORION), impactent la qualité des solutions, ou si la relaxation de toute contrainte de stretch fournit au final des solutions de coût acceptable.

Enfin, ce chapitre a permis d'appliquer le travail précédent pour déduire des bornes duales du challenge ROADEF, en prouvant que certaines agrégations permettent toujours de calculer des bornes. On a amélioré les bornes duales de l'état de l'art, le sujet étant peu étudié auparavant comme les équipes se consacraient à la recherche de meilleures bornes primales. De plus, pour un tel travail, nous avons réalisé des calculs sur plusieurs scénarios et observé que ces calculs étaient toujours réalisables avec un ordinateur portable avec la dernière version de Cplex. De telles considérations laissent à penser qu'avec des serveurs de calculs performants, des calculs sur un nombre conséquent de scénarios pourrait être réalisable.

A ce stade, on pourrait remarquer qu'une approche de reformulation de Dantzig-Wolfe par dualisation des contraintes couplant les réacteurs et une résolution par génération de colonnes pourrait être envisagée. En fait, les sous problèmes se retrouveraient alors être continus, ce qui ne fournirait pas d'amélioration de borne duale que la formulation compacte et sa résolution frontale. Une telle perspective est ainsi écartée.

Dans la suite de la thèse, les efforts seront portés sur le modèle sans stretch et sans condition initiale. Un premier objectif est d'avoir des approches constructives pour exhiber des solutions réalisables pour les données B8 et B9 à plus fortes combinatoire.

Chapitre 7

Résolutions déterministes heuristiques

Résumé : Ce chapitre dérive du travail de résolution PLNE des chapitres 5 et 6 des matheuristiques pour le problème de planification des maintenances et des rechargements des centrales nucléaires. On élabore des heuristiques constructives par processus hiérarchiques de fixations de variables. Les solutions construites sont améliorées par recherche locale, en adaptant l'algorithme de VND proposé au chapitre 4. Outre l'obtention d'approches performantes de résolution, l'approche de VND chiffre l'impact de différents voisinages sur le design d'une métaheuristique pour résoudre le problème de planification des maintenances et des rechargements des centrales nucléaires. Ce travail a fait l'objet de communications au congrès PGMO 2013, META 2014.

A l'issue du chapitre précédent, on a pu analyser les caractéristiques et les limites de la résolution frontale sur le problème déterministe, où les instances B8 et B9 les plus difficiles de par leur combinatoire ne permettaient pas d'avoir des solutions primales. La priorité de ce chapitre est de dériver de la formulation PLNE des matheuristiques permettant d'une part d'avoir des solutions sur B8 et B9, et d'autre part fournir des approches de résolution rapides et efficaces. La formulation PLNE retenue pour ce chapitre est la formulation PLNE sans stretch ni modulation. La première partie cherchera à dériver des heuristiques constructives de la formulation PLNE. La seconde partie développera une approche de recherche locale suivant le formalisme VNS en utilisant la formulation PLNE.

7.1 Heuristiques constructives

Les heuristiques constructives de cette section utilisent la PLNE par des stratégies de fixation de variables sur la formulation frontale, par une formulation simplifiée avec peu de variables continues, et enfin suivant des stratégies gloutonnes.

7.1.1 Fixation de variables sur la relaxation continue

Avec notre structure de variables, fixer une variable $d_{i,k,w}$ revient à restreindre les fenêtres de temps. Une première stratégie heuristique est de fixer toutes les variables entières dans la relaxation continue à leur valeur dans la relaxation continue. Les questions qui se posent à ce stade sont de savoir si un tel preprocessing de fixation de variable dégrade faiblement la fonction objectif, et si le gain en temps de calcul permet une nette amélioration de la résolution frontale. Le tableau 7.1 donne les résultats obtenus en terme de qualité de solution après 1h de temps de calcul après fixation, en les confrontant au gain de variables binaires correspondant.

Il est à noter que pour ces résultats, nous avons activé les contraintes de longueur maximale de cycle (avec une valeur $\mathbf{Lmax}_{i,k} = 90$, transparente sur la qualité des solutions) et qui aidait bien la fixation de variables dans le cas des instances B8 et B9 sans fenêtres de temps. Dans le cas contraire, on se

Instances	Valeur	Valeur heur	Gap	Nb var Frontl	Nb var heur	% var
A1	152170M	152179M	0,01%	424	131	30,90%
A2	145201M	145204M	0,00%	761	219	28,78%
A3	152582M	152610M	0,02%	698	323	46,28%
A4	102421M	102444M	0,02%	1493	1124	75,28%
A5	119785M	119822M	0,03%	2494	1579	63,31%
Sous total A	672162M	672259M	0,01%	5870	3376	57,51%
B6	76966M	76991M	0,03%	3054	1455	47,64%
B7	74233M	74303M	0,09%	5846	2840	48,58%
B8	73239M	96565M	31,85%	20763	7179	34,58%
B9	72812M	76210M	4,67%	23675	8564	36,17%
B10	69501M	69549M	0,07%	3790	1138	30,03%
Sous Total B	366753M	393619M	7,33%	57128	21176	37,07%
X11	73018M	73063M	0,06%	3216	1025	31,87%
X12	70604M	70657M	0,07%	4035	1363	33,78%
X13	69230M	69340M	0,16%	4104	1462	35,62%
X14	68395M	68489M	0,14%	4879	1980	40,58%
X15	66028M	66089M	0,09%	3618	1076	29,74%
Sous Total X	347277M	347638M	0,10%	19852	6906	34,79%
Total	1386195M	1413516M	1,97%	82850	31458	37,97%

TABLE 7.1 – Résultat de la fixation de variable sur la relaxation PL

retrouvait dans des situations de relaxation continues comme analysé à la section 6.1.2, engendraient une faible réductions de variables sur les derniers pas de temps des instances B8 et B9, et cela ne permettaient toujours pas d’accélérer suffisamment la convergence pour avoir des solutions primales de valeur raisonnable.

Si les résultats de la fixation LP en terme de qualités de solutions sont plutôt bons, cette approche n’a pas été retenue dans la suite. Sur les instances hors B8, B9, la formulation frontale fournit toujours des meilleures solutions, la fixation de variables n’apporte pas de nette plus valeur sur la convergence comme ce fut le cas au chapitre 4 pour le problème UCPd. Dans le cas des instances B8 et B9, les temps de calculs longs nécessaire au preprocessing et après convergence ne s’avérait pas compétitif par rapport aux autres approches développées dans la suite.

7.1.2 Formulation sans variables de production nucléaire

Une idée pour simplifier la modélisation PLNE était de faire l’approximation que la puissance des unités nucléaires était toujours la puissance maximale sur les cycles de production. Une justification de ce fait est que la production nucléaire est économiquement plus rentable que la production flexible, et pour un pas de temps donné où la demande est supérieure à la puissance nucléaire totale disponible, le problème d’optimisation locale à ce pas de temps utilise toutes les centrales nucléaires à leur maximum, avant de faire appel aux unités non nucléaires, selon les ordres croissants de coûts marginaux.

Mathématiquement, cela revient à imposer $p_{i,k,t} = \mathbf{Pmax}_{i,t}(d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t})$, permettant de se séparer des variables de production nucléaire. Tel quel, cela mène à des formulations irréalisables la plupart du temps. Avec une production à puissance maximale, la consommation de combustible est maximale, et cela va en porte à faux avec les contraintes de positivité du stock sur les cycles de production longs, où des fenêtres de temps lointaines imposent de devoir moduler la production. Pour éviter de telles infaisabilités, on introduit $\Delta_{i,k}$ comme une quantité de stock additionnelle en fin de cycle, pour permettre la positivité du stock en fin de cycle. Ces violations de contraintes sont pénalisées proportionnellement au stock additionnel utilisé, avec un coefficient $C_{i,k}^{\Delta}$ élevé, pour se rapprocher au plus d’une solution réalisable à production maximale. De plus, la contrainte de demande est écrite comme une inégalité, pour rester faisable dans les cas où la demande est plus basse. Cela donne ainsi la formulation simplifiée suivante :

$$\begin{aligned}
v^{simpl} = \min \quad & \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,t} \mathbf{C}_{j,t}^{prd} \mathbf{D}^t p_{j,t} - \sum_i \mathbf{C}_i^{val} x_i^{fin} + \sum_i C_{i,k}^{\Delta} \Delta_{i,k} \\
& A^{ordo} d \geq a^{ordo} \\
& \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \\
\forall i, k, \quad & x_{i,k}^{init} \leq \mathbf{Smax}_{i,k} \\
\forall i, k, \quad & x_{i,k}^{fin} = x_{i,k}^{init} - \sum_t \mathbf{D}^t p_{i,k,t} + \Delta_{i,k} \\
\forall i, k, \quad & x_{i,k}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
\forall i, k, \quad & x_{i,k}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k}-1}{\mathbf{Q}_{i,k}} (x_{i,k-1}^{fin} - \mathbf{Bo}_{i,k-1}) \\
\forall i, k, \quad & x_i^{fin} \leq x_{i,k}^{fin} + \mathbf{Smax}_i (d_{i,k,W} - d_{i,k+1,W}) \\
\forall j, t, \quad & \mathbf{Pmin}_{j,t}^s \leq p_{j,t} \leq \mathbf{Pmax}_{j,t}^s \\
\forall i, t, \quad & p_{i,k,t} = \mathbf{Pmax}_{i,t} (d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \\
\forall t, \quad & \sum_{i,k} p_{i,k,t} + \sum_j p_{j,t} \geq \mathbf{Dem}^t \\
& d \in \{0, 1\}^N, r, p, x, \Delta \geq 0
\end{aligned}$$

De telles solutions ne permettent pas toujours de fournir des dates d'arrêts fournissant une solution réalisable après un calcul de production selon la formulation 5.3.1. Une réparation de solution est alors utilisée en post traitement en pénalisant les écarts d'une solution réalisable à la solution construite par le modèle simple. Un tel processus permettrait d'obtenir des solutions primales pour toutes les instances en temps limité. Pour B8 et B9, 15 minutes de temps de calculs étaient suffisant pour obtenir des solutions. L'avantage d'une telle formulation est qu'en se concentrant sur les variables binaires, la résolution était accélérée, et se concentrait vite sur la phase de branchements, générant beaucoup de solutions primales rapidement. Ce constat incite à utiliser cette formulation pour générer des populations de solutions.

7.1.3 Algorithmes de type gloutons/Relax and Fix

Des algorithmes gloutons ont été implémentés en construisant une solution par résolution successive de petits PLNE d'optimisation locale. Contrairement aux gloutons classiques, ces problèmes d'optimisations vont placer plusieurs arrêts à la fois, garantissant une optimisation de meilleure qualité. Deux variantes de construction ont été considérées, construire la solution unités par unités, ou cycles par cycles.

Algorithme glouton unités par unités La stratégie gloutonne "par unités" considère autant de problèmes que d'unités nucléaires, en optimisant simultanément les arrêts d'un même réacteur. Une première implémentation considèrerait séquentiellement les planning à optimiser. Les unités non considérées dans l'optimisation courante et qui n'ont pas été déjà traitées étaient alors considérées comme des unités de type 1, en leur affectant un coût de production d'une unité de type 1 existante. Les unités déjà optimisées précédemment avait alors leur planning figé à la valeur de l'optimisation réalisée. Les plannings traités étaient alors figés dans la suite à la valeur de leur optimisation via les contraintes de fenêtres de temps, les variables considérées de production nucléaires non figées à 0 étant alors de cardinalité inférieure aux variables de production. On peut alors utiliser directement la formulation de la section 5.3.1 avec de telles modifications.

Avec cette méthode, on fige successivement des dates d'arrêts, telles que les contraintes couplantes d'ordonnement (CT14 à CT21) soient vérifiées avec les précédentes décisions, sans pouvoir les remettre en question. Il pouvait ainsi arriver qu'un problème de planification des arrêts d'une unité soit infaisable, à cause des contraintes d'ordonnement et des arrêts précédemment figés. Pour avoir une méthode robuste,

il s'avérait alors nécessaire de relâcher les contraintes couplantes d'ordonnement de CT14 à CT21, et de réparer les violations des contraintes d'ordonnement comme précédemment en considérant le PLNE frontal de la section 5.3.1 avec un fort poids de minimisation de l'écart à la solution ainsi construite.

Face à ces désagréments, et pour réduire l'asymétrie provoquée par le choix de l'ordre de fixation, une approche parallélisable en considérant des sous problèmes indépendants de fixation des variables s'avérait adapté. Le problème d'optimisation pour chaque réacteur considère alors la formulation 5.3.1 en agrégeant toutes les autres unités en une unité de type 1. Il s'agit d'un planning idéal de l'unité considérée indépendamment des autres, l'arbitrage des modifications pour respecter les contraintes d'ordonnement étant alors effectué par la réparation. Par une telle construction, des solutions sont générées très rapidement, les problèmes de résolution par unités étant instantanés.

Algorithme glouton cycles par cycles Une autre stratégie gloutonne est de procéder cycles par cycles planifiant successivement les arrêts. De manière à avoir des calculs intermédiaires plus faciles à résoudre, on utilise une formulation tronquée ne considérant que les Ko premiers arrêts, la production étant alors proche du type 1 après l'arrêt Ko :

$$\begin{aligned}
v_{Ko} = \min & \quad \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{j,t} \mathbf{C}_{j,t}^{prd} \mathbf{D}^t p_{j,t} \\
& \quad A_{Ko}^{ordo} d \geq a_{Ko}^{ordo} \\
\forall i, k < Ko, & \quad \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \\
\forall i, k \leq Ko, & \quad x_{i,k}^{init} \leq \mathbf{Smax}_{i,k} \\
\forall i, k < Ko, & \quad x_{i,k}^{fin} = x_{i,k}^{init} - \sum_t \mathbf{D}^t p_{i,k,t} + \Delta_{i,k} \geq 0 \\
\forall i, k < Ko, & \quad x_{i,k}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
\forall i, k \leq Ko & \quad x_{i,k}^{init} - \mathbf{Bo}_{i,k} = r_{i,k} + \frac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}} (x_{i,k-1}^{fin} - \mathbf{Bo}_{i,k-1}) \\
\forall j, t, & \quad \mathbf{Pmin}_{j,t}^s \leq p_{j,t} \leq \mathbf{Pmax}_{j,t}^s \\
\forall t, & \quad \sum_{i,k} p_{i,k,t} + \sum_j p_{j,t} \geq \mathbf{Dem}^t \\
& \quad d \in \{0, 1\}^N, r, p, x^{init}, \geq 0
\end{aligned}$$

Les contraintes $A_{Ko}^{ordo} d \geq a_{Ko}^{ordo}$ sont les contraintes d'ordonnement restreintes aux variables $d_{i,k,w}$ pour $k \leq Ko$. Dans cette formulation, on a dû relâcher la contrainte de positivité du stock du cycle Ko pour que la production nucléaire puisse se faire jusqu'au dernier pas de temps à puissance maximale, et en espérant que les arrêts suivants soient suffisamment loin pour ne pas perturber la décision prise localement. En outre, la valorisation du stock final est supprimée, comme cela n'a plus de sens.

Un premier algorithme glouton procéderait en calculant à chaque itération k où les $k-1$ premiers cycles ont été fixés le problème v_k après avoir fixé les fenêtres de temps des $k-1$ premiers cycles sur les valeurs calculées aux itérations précédentes, fixant les solutions obtenues comme valeur définitive des k premiers cycles avant de passer à l'itération $k+1$.

Une telle approche est séquentielle, non parallélisable, chaque sous-problème nécessitant la sortie du précédent. Dans le cas où un sous problème se trouve irréalisable ou qu'aucune solution primale n'a été exhibée, la résolution s'en trouve arrêtée, et il n'y a aucun mécanisme de réparation d'implémenté. Cette approche ne garantit donc pas de fournir une solution en temps contraint. De nombreuses irréalisabilités apparaissaient, pas manque d'anticipation. Les arrêts d'un cycle k sont figés sans considérer de réalisabilité des cycles suivants : des infaisabilités pouvant apparaître suite à une mauvaise fixation d'un cycle k avec les contraintes d'anticipation maximales, des fenêtres de temps des cycles suivants, et des ordonnancements restreignant la faisabilité.

Pour ce faire, on a donc considéré un cycle supplémentaire, pour garantir l'existence de solutions futures au cycle suivant. À chaque itération k où les $k-1$ premiers cycles ont été fixés, on considère alors le problème v_{k+1} après avoir fixé les fenêtres de temps des $k-1$ premiers cycles sur les valeurs calculées aux itérations précédentes, fixant les solutions obtenues comme valeur définitive des k premiers cycles avant de passer à l'itération $k+1$. Par une telle approche, les problèmes d'optimisation considèrent le placement de $2I$ arrêts, contre K pour l'approche de construction unités par unités. Cela génère des problèmes d'optimisation qui ne se résolvent pas instantanément, mais avec une convergence suffisamment accélérée pour que toutes les instances soient calculables. Les temps de calculs à l'optimalité étaient inférieurs à 5 minutes pour les instances des jeux B et X hors B8 et B9.

La résolution en temps fixé, avec un paramètre de résolution activant les heuristiques (`mipemphasis=4`) était très bien adapté. La variabilité des temps de calculs suivant les cycles (par l'intermédiaire des amplitudes des fenêtres de temps) a été prise en compte dans les temps de calculs impartis. Pour les instances B8 et B9, 10 minutes était un temps de convergence insuffisant, les itérations les plus critiques nécessitant 20 minutes pour fournir une solution primale d'un ordre de grandeur acceptable.

On remarque que l'on pourrait réutiliser de l'information pour un démarrage à chaud. Pour le placement du cycle k , la boucle précédente avait donné une solution réalisable pour les variables d'arrêt du cycle k . En fixant ces valeurs dans v_{k+1} , on a un problème de taille I , qui peut fournir des solutions initiales pour un démarrage à chaud de l'itération k , et ainsi faciliter la convergence du PLNE.

7.1.4 Résultats des approches constructives

Les résultats des heuristiques primales constructives sont référencés sur le tableau 7.2. A ce stade, les heuristiques primales ont pu fournir des solutions pour toutes les instances, ce qui était le but premier de cette étude. L'approche par calcul PLNE simplifié a des résultats proches de l'algorithme glouton "unités par unités", pour des temps de calculs courts on obtient des solutions assez moyennes, les temps de calcul étant quasi instantanés pour le glouton par unités. Le glouton cycles par cycles fournit des solutions d'excellente qualité, au prix de temps de calculs bien plus longs.

Instances	Meilleure Valeur	Frontal 1h	PLNE simplifié	Glouton Unités	Glouton Cycle
B6	76966	0,00%	3,22%	3,8%	0,34%
B7	74234	0,02%	3,68%	0,75%	0,25%
B8	73240	NS	52,54%	52,21%	2,02%
B9	72812	NS	27,46%	28,95%	1,15%
B10	69501	0,02%	0,21%	0,2%	0,21%
X11	73018	0,00%	0,45%	0,53%	0,38
X12	70604	0,00%	0,25%	0,24%	0,31%
X13	69231	0,04%	7,37%	9%	0,71%
X14	68395	0,02%	6,02%	1,21%	0,54%
X15	66029	0,06%	0,12%	0,12%	0,38%
Total	714031	NS	10,31%	9,89%	0,63%

TABLE 7.2 – Résultats des heuristiques primales constructives

7.2 Amélioration de solutions par descente VNS

A ce stade, des solutions ont été exhibées pour toutes les instances, en temps de calcul brefs. L'objet de cette section est d'appliquer les idées génériques de la section 4.2.1 déjà appliquées au problème UCPd, avec une recherche locale basée sur des calculs intermédiaires de PLNE réduits.

7.2.1 Déploiement et enjeux des VNS avec voisinages PLNE

Fixer des variables binaires sur ce problème revient exactement à réduire les fenêtres de temps. Les voisinages sont définis directement sur les dates d'arrêt, structure naturelle des voisinages locaux, la résolution PLNE prenant alors en charge les considérations de coûts et de faisabilité issues des variables continues.

Cette étude présente deux enjeux. Tout d'abord, il s'agit de pouvoir dériver une heuristique VNS donnant de bons résultats sur le problème, et d'obtenir des minima locaux d'excellente qualité pour la VNS. Le temps de calcul n'étant pas limité dans cette application, contrairement au chapitre 4 sur le problème UCPd. Le paramétrage tronque moins la résolution aux tous premiers stades de résolution PLNE pour une recherche agressive de solutions, et nous nous concentrons sur des valeurs de convergence, sans optimiser la vitesse de convergence (assez rapide dans le cas présent de toute façon). Une autre application de l'algorithme utilisé est de chiffrer indépendamment les effets des différents choix de voisinages, calculant les minima locaux obtenus en ne considérant qu'un seul type de voisinage. La PLNE fourni des grands voisinages, dans lequel les voisinages utilisés par les différentes approches de recherche locale peuvent se plonger.

7.2.2 Définition des voisinages

Autour d'un planning initial, on réoptimise suivant les fixations de variables suivantes :

- voisinage N-Universel : on définit des fenêtres de temps dans un rayon de N semaines autour de la solution courante.
- voisinage universel $v(a, b)$: on définit des fenêtres de temps dans un rayon de $ak + b$ avec, $a, b \geq 0$ semaines autour de la solution courante. Le voisinage N-universel correspond alors à $v(O, N)$.
- voisinage k-Cycle : On réoptimise juste les placements d'arrêts des k ièmes cycles. Cela permet d'échanger les ordres des arrêts de mee inide.
- voisinage k-Pairs : On réoptimise juste les placements d'arrêts des cycles k et $k + 1$. Cela permet d'échanger les ordres de plus d'arrêts encore.
- voisinage i-planning : on réoptimise tout le planning d'arrêt d'un réacteur donné i .
- voisinage i-planning(N) : on réoptimise tout le planning d'arrêt des N réacteurs d'indice i' tel que $i \leq i' \leq i + N - 1$. Le voisinage i-planning précédent est alors par extension i-planning(1).

7.2.3 Résultats des voisinages séparés

On considère dans cette section les résultats obtenus en considérant les différents type de voisinages séparément, pour analyser les minima locaux correspondant. Le critère d'arrêt de la convergence est que le voisinage choisi ne permet plus d'améliorations locales. Une telle approche permet de comparer la qualité des minimums locaux en partant d'une même solution de départ, donnée par la formulation de la section 5.3.1 dans nos expérimentations numériques.

Voisinages à rayon universel Pour chiffrer les minima locaux des approches à rayon universel, on considère toujours le même type de voisinage. Le voisinage 1-universel, $v(0, 1)$, comprend entre autres les voisinages en considérant un unique déplacement d'un arrêt nucléaire de 1 semaine, ce qui est un voisinage

utilisé naturellement, ou encore les translations de tous les arrêts d'une centrale de 1 semaine. La qualité du minimum local obtenu par un tel voisinage s'avérait médiocre, le minimum local obtenu avec le sous voisinage ne considérant qu'un seul déplacement de la solution courante serait donc encore plus mauvaise. Une telle structure de voisinage mène alors à de nombreux minimums locaux.

En augmentant la taille du voisinage, on améliore la qualité des minimas locaux, en restant toutefois toujours éloigné des meilleures solutions trouvées. Cela justifie de devoir utiliser d'autres type de voisinages que des optimisations locales autour de la solution courante. Sur la qualité des minimas locaux, on notera qu'une structure de voisinage "en entonnoir" $v(a, b)$ avec $a > 0$ permettant d'avoir plus de marges de manœuvre sur les derniers cycles est souhaitable. En prenant $v(0, 5)$, la qualité des solutions était perturbée par les temps de calculs limites de 30s, suffisants pour les voisinages plus petits.

Instances	Valeur	Frontal	Init	$v(0,1)$	$v(1,2)$	$v(0,3)$	$v(1,3)$	$v(0,5)$
B6	76966M	0	3,22	0,54	0,13	0,26	0,13	0,13
B7	74233M	0,02	3,68	1,14	0,41	0,47	0,35	0,19
B8	73239M	NS	52,54	38,34	19,70	31,12	16,05	26,73
B9	72812M	NS	27,46	15,91	7,54	12,35	7,54	8,89
B10	69501M	0,02	0,21	0,1	0,05	0,06	0,02	0,05
Total B	366753M	NS	17,40	11,18	5,55	8,83	4,80	7,18
X11	73018M	0	0,45	0,31	0,10	0,26	0,22	0,27
X12	70604M	0	0,25	0,13	0,10	0,09	0,07	0,06
X13	69230M	0,04	7,37	2,23	1,30	1,47	0,93	1,01
X14	68395M	0,02	6,02	1,64	0,48	0,61	0,45	0,54
X15	66028M	0,06	0,12	0,08	0,06	0,07	0,06	0,06
Total X	347277M	0,02	2,82	0,87	0,41	0,50	0,34	0,39
Total B et X	714030M	NS	10,31	6,17	3,05	4,78	2,63	3,87

TABLE 7.3 – Résultats des voisinages à rayon universel

Voisinages réoptimisant par cycles Le minimum local "par cycle" est obtenu lorsque les voisinages k -cycles pour $1 \leq k \leq K$ ne donnent pas d'améliorations de la solution courante, tandis que le minimum local "par 2 cycles" considère tous les voisinages. Ces voisinages permettent d'échanger les ordres entre des arrêts soumis aux contraintes d'espacement CT14-CT15, ce que les voisinages universels ne permettraient pas de faire, avec des paramètres petits. Cependant, la qualité des minima locaux obtenus reste toujours loin des meilleures valeurs.

Instances	Valeur	Frontal 1h	Init	Par cycles	Par 2 cycles
B6	76966M	0	3,22	0,31	0,13
B7	74233M	0,02	3,68	0,55	0,40
B8	73239M	NS	52,54	35,79	27,63
B9	72812M	NS	27,46	14,48	12,12
B10	69501M	0,02	0,21	0,08	0,03
Total B	366753M	NS	17,40	10,21	8,04
X11	73018M	0	0,45	0,30	0,26
X12	70604M	0	0,25	0,15	0,10
X13	69230M	0,04	7,37	2,43	0,28
X14	68395M	0,02	6,02	1,40	0,60
X15	66028M	0,06	0,12	0,10	0,08
Total X	347277M	0,02	2,82	0,87	0,26
Total B et X	714030M	NS	10,31	5,67	4,26

TABLE 7.4 – Résultats des voisinages réoptimisant par cycles

Voisinages reoptimisant par unités Contrairement aux situations précédentes, réoptimiser tout le planning d'une unité permet des réoptimisations radicales de plannings d'arrêts qui ont une justification applicative. Globalement, la planification des arrêts est plus rentable l'été, où le coût de substitution à la production nucléaire est le moins élevé. De grands déplacements sont utiles dans la définition des voisinages pour permettre de passer de part et d'autre de l'hiver. De tels voisinages permettent aussi d'ajouter ou d'enlever des arrêts en fin de planning.

En utilisant les voisinages réoptimisant tout le planning d'une unité, on a des calculs très rapides, et la convergence donne des extremums locaux en considérant toutes ces réoptimisations très proches

des meilleurs valeurs. Considérer plusieurs unités dans la réoptimisation améliorerait encore la qualité des minimas locaux obtenus, en autorisant des interversions de l'ordre des arrêts des unités réoptimisées. L'ordre des voisinages était déterministe, en considérant toujours les unités adjacentes. Afin de faire diminuer les valeurs des minimas locaux, et de mieux gérer les possibilités de réorganisations sur un même site (avec les contraintes CT14 et CT15 qui vont avec), les regroupements se font suivant deux séquences interverties. Si n est le nombre d'unités réoptimisées simultanément, on considère les séquences d'unités d'index entre $mn + 1$ et $mn + n$ puis les regroupements suivants les séquences $mn + \lfloor \frac{n}{2} \rfloor + 1$ et $mn + \lfloor \frac{n}{2} \rfloor + n$.

Instances	Valeur	Frontal 1h	Init	1 unité	5 unités	8 unités
B6	76966M	0,00%	3,22%	0,26%	0,03%	0,04%
B7	74233M	0,02%	3,68%	0,34%	0,06%	0,02%
B8	73239M	NS	52,54%	1,18%	0,36%	0,36%
B9	72812M	NS	27,46%	1,31%	0,05%	0,05%
B10	69501M	0,02%	0,21%	0,05%	0,03%	0,03%
Total B	366753M	NS	17,40%	0,63%	0,11%	0,10%
X11	73018M	0,00%	0,45%	0,16%	0,09%	0,09%
X12	70604M	0,00%	0,25%	0,08%	0,05%	0,05%
X13	69230M	0,04%	7,37%	0,51%	0,07%	0,07%
X14	68395M	0,02%	6,02%	0,28%	0,13%	0,13%
X15	66028M	0,06%	0,12%	0,09%	0,05%	0,05%
Total X	347277M	0,02%	2,82%	0,22%	0,08%	0,08%
Total B et X	714030M	NS	10,31%	0,43%	0,09%	0,09%

TABLE 7.5 – Résultats des voisinages réoptimisant par unités

7.2.4 Résultats de l'approche VNS

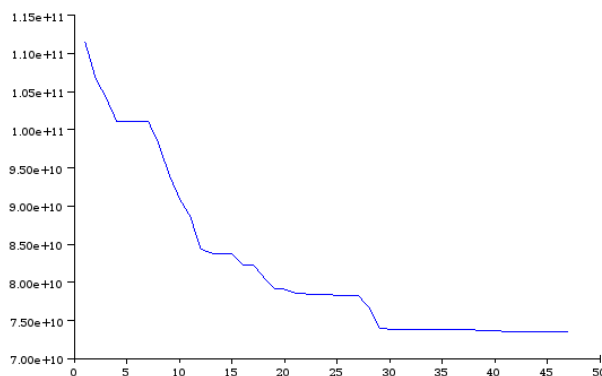


FIGURE 7.1 – Convergence VNS

En combinant les différents types de voisinages, on obtient des graphiques de convergence analogue à la figure 7.1, sur l'instance difficile B8, avec des phases marquées suivant les types de voisinages. Si l'utilisation de certains voisinages semble être moins efficace en rapport amélioration/temps de calcul, ces voisinages permettent néanmoins d'améliorer significativement les qualités des extremums locaux. Les temps de calcul n'étant pas dimensionnant sur ces calculs, et assez courts qui plus est, c'est bien la recherche de minimas locaux de la meilleure qualité possible qui doit guider la paramétrisation et le choix des voisinages.

Les tableaux 8.3 et 7.6 illustrent en terme de qualité de minimas locaux, l'intérêt de croiser les différents voisinages. Dans notre cas applicatif et avec les jeux de données ROADEF, la combinaison de voisinages

qui donnait les meilleurs extremums locaux étaient les $i - planning(8)$, tous les $k - Pairs$ et $v(1,3)$. Avec de tels voisinages, on retrouvait dans tous les cas (test effectué également sur les instances tronquées) une valeur dans les 0,01% de l'optimum quand il était prouvé, ou de la meilleure solution courante trouvée. En fait, on obtenait très souvent une amélioration de l'ancienne meilleure valeur connue, ce type d'approche semblait mieux faire la différence entre les solutions dans les 0,01% de tolérance du solveur PLNE. En fin de compte, avec cette VNS, l'approche la plus intéressante est d'avoir très rapidement une première solution primale et de l'améliorer par VNS plutôt que de partir d'une solution primale de très bonne qualité obtenue dans un temps de calcul conséquent.

Instances	Valeur	Frontal 1h	Init	v(1,3)	Pairs	8 sautsHiv	VNS
B6	76966M	0,00%	3,22%	0,13%	0,13%	0,04%	0,00%
B7	74233M	0,02%	3,68%	0,35%	0,4%	0,02%	0,00%
B8	73239M	NS	52,54%	16,05%	27,63%	0,36%	0,00%
B9	72812M	NS	27,46%	7,54%	12,12%	0,05%	0,00%
B10	69501M	0,02%	0,21%	0,02%	0,03%	0,03%	0,00%
Total B	366753M	NS	17,4%	4,8%	8,04%	0,1%	0,00%
X11	73018M	0,00%	0,45%	0,22%	0,26%	0,09%	0,00%
X12	70604M	0,00%	0,25%	0,07%	0,1%	0,05%	0,00%
X13	69230M	0,04%	7,37%	0,93%	0,28%	0,07%	0,00%
X14	68395M	0,02%	6,02%	0,45%	0,6%	0,13%	0,00%
X15	66028M	0,06%	0,12%	0,06%	0,08%	0,05%	0,00%
Total X	347277M	0,02%	2,82%	0,34%	0,26%	0,08%	0,00%
Total B et X	714030M	NS	10,31%	2,63%	4,26%	0,09%	0,00%

TABLE 7.6 – Résultats VNS finaux

7.3 Synthèse des résultats et perspectives

Les résultats des heuristiques primales constructives ont pu fournir des solutions pour toutes les instances, ce qui était le but premier de cette étude. L'approche par calcul PLNE simplifié a des résultats proches de l'algorithme glouton "unités par unités", pour des temps de calculs courts on obtient des solutions assez moyennes, les temps de calcul étant quasi instantanés pour le glouton par unités. Le glouton cycles par cycles fournit des solutions d'excellente qualité, au prix de temps de calculs bien plus longs.

Partant de telles solutions initiales, l'approche VNS déjà introduite au chapitre 4 a également donné d'excellents résultats. Un premier apport est d'avoir chiffré les impacts individuels de différents types de voisinages, et justifié l'importance pressentie à l'opérationnel de voisinages de "sauts d'hiver", ou retrouvé que l'utilisation de voisinages avec un décalage de 1 semaine au plus donnait des minimums locaux de mauvaises qualité. Sous le formalisme VNS, on a pu choisir un sous ensemble de voisinages donnant selon l'approche VND des résultats qui égalaient les meilleurs résultats connus, dans les 0,01% de tolérance de Cplex, fournissant même souvent des améliorations par rapports aux solutions obtenues par Cplex, même à terminaison. En fin de compte, avec cette VNS, l'approche la plus intéressante est d'avoir très rapidement une première solution primale et de l'améliorer par VNS plutôt que de partir d'une solution primale de très bonne qualité obtenue dans un temps de calcul conséquent.

Ce travail et ces constats ouvrent des perspectives. Tout d'abord, les solutions obtenues n'ont pas été projetées et calculées suivant l'espace et le critère d'optimisation du challenge ROADEF. Un tel travail serait intéressant pour étudier l'impact des simplifications d'agrégations des scénarios stochastiques et des

pas de temps de production à la semaine. De plus, les temps de calculs étant très faibles avec l'approche VNS, considérer des scénarios stochastiques, à minimum 3 le scénario moyen et les extrêmes de demande cumulée, et même pour un nombre de scénarios bien supérieur semble accessible avec la méthode au vu de la facilité de résolution des sous-problèmes restreints.

Chapitre 8

Planification robuste des arrêts nucléaires

Résumé : Ce chapitre introduit des aléas sur les durées des arrêts de maintenance. Un cadre de modélisation robuste est privilégié pour des raisons opérationnelles. Ce chapitre introduit la robustesse dans la modélisation du challenge ROADEF tout en conservant les aléas stochastiques du challenge, pour développer un algorithme de résolution générique par décomposition de Benders. Des modélisations plus souples de la robustesse ont également été proposées. Une approche paramétrique de la robustesse fournit des fronts de Pareto arbitrant niveaux de robustesse et surcoûts de solutions dûs à la couverture robuste d'aléas. Ce travail a fait l'objet de communications aux congrès ISCO 2012 et EURO 2012.

Les aléas considérés pour le challenge ROADEF considéraient les demandes, les coûts et les capacités de production des unités T1 non nucléaires. Ces aléas étaient surtout impactants en termes de coûts de solution, plus qu'en termes de faisabilité du planning nucléaire. Une modélisation de programmation stochastique s'avérait alors bien adaptée. Ce chapitre considère suivant le besoin opérationnel des aléas de prolongations d'arrêt. De tels aléas peuvent impacter la faisabilité du problème tel que défini par le challenge ROADEF. On étudie ici l'intégration d'une gestion robuste de tels aléas dans le modèle.

8.1 Considérations de robustesse dans le modèle

Dans cette section, nous introduisons une gestion robuste d'aléas non considérés dans le challenge ROADEF, de la modélisation aux techniques de résolutions associées.

8.1.1 Modélisation des aléas

Prolongations des arrêts Les arrêts des réacteurs nucléaires pour maintenance et rechargement sont des opérations industrielles complexes soumises à des aléas, des retards peuvent alors survenir. La grandeur $\mathbf{Da}_{i,k}$, est ainsi soumise à des aléas. De tels aléas peuvent perturber la faisabilité du planning, directement via les contraintes d'ordonnancement des arrêts, ou indirectement via la production et les contraintes d'anticipations maximales. Le besoin opérationnel préconisait une gestion robuste de tels aléas, sur un ensemble d'incertitude à définir. Une tel besoin opérationnel présente des différences majeures avec l'état de l'art en optimisation robuste. D'une part, l'incertitude porte sur une quantité discrète $\mathbf{Da}_{i,k}$, là où l'état de l'art considère des ensembles d'incertitudes continus. De plus, dans le modèle en PLNE, les dates d'arrêt interviennent de manière non linéaire, puisque les $\mathbf{Da}_{i,k}$ se trouvent en indice de certaines contraintes.

Ensemble d'incertitude considéré Du fait de la non linéarité de notre formulation PLNE, il nous a été nécessaire pour garder une structure linéaire d'énumérer un nombre discret de scénarios de prolongation

d'arrêts. On notera Ω , l'ensemble d'incertitude, ensemble de scénarios discrets.

Pour tout $\delta \in \Omega$, on associe la prolongation d'arrêt du cycle (i, k) , $\delta_{i,k}$, la longueur de l'arrêt (i, k) pour le scénario de prolongation d'arrêt (i, k) est à présent $\mathbf{Da}_{i,k,\delta} = \mathbf{Da}_{i,k} + \delta_{i,k}$. On notera $\overline{\mathbf{Da}}_{i,k} = \max_{\delta \in \Omega} \mathbf{Da}_{i,k,\delta}$ la prolongation maximale de l'arrêt (i, k) . Les scénarios de prolongation d'arrêt considérés privilégiés correspondent aux scénarios où un seul arrêt peut prendre un retard extrême, tandis que les autres arrêts n'ont pas de prolongation (ou une prolongation moyenne incorporée dans $\mathbf{Da}_{i,k}$).

On peut considérer uniquement des prolongations non nulles pour les premiers arrêts, en pensant que les arrêts plus lointains pourront être réoptimisés, et donc que les exigences de robustesse sont plus fortes sur les premiers arrêts. Cela permet de définir des ensembles d'incertitudes paramétriques $\Omega(k_0, w_0)$, ensemble où des incertitudes ne portent que sur les k_0 premiers arrêts, un scénario comportant une prolongation non nulle sur un unique arrêt, où la prolongation est de w_0 semaines.

8.1.2 Écriture générique des problèmes déterministes

Pour toutes les formulations MIP considérées, le problème s'écrit sous la forme générique suivante :

$$\min_{x \in \{0,1\}^n \times \mathbb{R}_+^m, y \geq 0} cx + qy \quad (8.1)$$

$$Ax \leq a \quad (8.2)$$

$$T_\delta x + W y \leq h \quad (8.3)$$

$$By \leq b \quad (8.4)$$

Le vecteur x comprend les jeux de variables $d_{i,k,w}$ et $r_{i,k}$, dates d'arrêts et les niveaux de rechargements. Le vecteur y correspond aux variables de second niveau, variables de productions et de stocks nucléaires. correspondant à toutes les décisions relatives au plan de production une fois que les décisions de premier niveau x ont été fixées, décisions de production et variables auxiliaires nécessaires (éventuellement pour tout scénario stochastique s). Ce sont uniquement des variables continues.

Les contraintes couplantes entre les variables de premier niveau où les durées d'arrêt et donc les prolongations δ n'interviennent pas sont écrites avec l'ensemble de contraintes (8.2). Il s'agit des contraintes (5.11). Les contraintes (8.4) regroupent les contraintes de bornes de production flexibles et les demandes, et éventuellement les contraintes dues à une modélisation du stretch. Les contraintes (8.3) comprennent les autres contraintes, faisant intervenir les durées d'arrêt et/ou couplant variables de premier et de second niveau. Il s'agit des contraintes d'ordonnancement sauf CT16 et CT19, des contraintes (5.11) et aussi des contraintes (5.13), (5.14), (5.15) comme elles font intervenir les variables $r_{i,k}$ couplantes.

8.1.3 Schéma robuste à 2 étapes

La structure du problème se prêtait bien à une modélisation robuste à deux étapes, les aléas sont surtout définis sur des ensembles d'incertitudes. Les variables sont de deux types : les variables x sont décidées sans connaître les aléas, tandis que les ajustements de production (et les niveaux de stocks dépendants) ne sont mis en oeuvre et connus qu'après connaissance des aléas. Bien qu'étant dans un cadre mathématique différent de l'état de l'art, on adapte le formalisme robuste à deux niveaux en considérant les problèmes :

$$\begin{aligned} \min_{x, y \geq 0} \quad & cx + \mathcal{Q}(x) & \text{où } \mathcal{Q}(x) = \max_{\delta \in \Omega} \min_y \quad & qy \\ \text{s.c : } \quad & Ax \leq a & \text{s.c : } \quad & T_\delta x + W y \leq h \\ & x \in \{0,1\}^n \times \mathbb{R}_+^m & & By \leq b \end{aligned} \quad (8.5)$$

Un tel paradigme adresse des questions de faisabilités et d'optimalité. Le choix d'une solution x est conditionné par la faisabilité pour toute occurrence $\delta \in \Omega$ d'une solution de second niveau réalisable y . Dans le cas contraire, le problème de maximisation de l'adversaire fictif donnerait $\mathcal{Q}(x) = +\infty$ en choisissant le scénario d'irréalisabilité. De plus, le choix de x est conditionné par la meilleure valeur de l'objectif après le recours, dans le pire des cas pouvant survenir suite à la réalisation de l'aléa.

Linéarisation du problème min-max-min robuste Pour linéariser le problème min-max-min, on énumère les valeurs possibles des variables y sur les différents scénarios énumérés δ , ce qui donne des variables y_δ reliés par les blocs matriciels $T_\delta x + W y_\delta \leq d$ et on introduit la variable continue C , qui représentera $C = \max_{\delta \in \Omega} \min q(\delta)y$. Le problème se linéarise alors en :

$$\min_{x \in \{0,1\}^n \times \mathbb{R}_+^m, y_\delta \geq 0} cx + C^{rob} \quad (8.6)$$

$$Ax \leq a \quad (8.7)$$

$$\forall \delta \quad T_\delta x + W y_\delta \leq h \quad (8.8)$$

$$\forall \delta \quad B y_\delta \leq b \quad (8.9)$$

$$\forall \delta \quad q y_\delta \leq C^{rob} \quad (8.10)$$

La contrainte (8.10) écrit le coût robuste comme le pire coût sur les scénarios considérés. On considère la variable C^{rob} comme une variable de premier niveau. Dans le cas de l'approche déterministe robustifiée, on a $q = 0$, et l'écriture reste valide. La structure matricielle obtenue est alors assez analogue au cadre de l'optimisation stochastique avec recours, avec comme variables couplantes x, C . On a alors des sous blocs indépendants pour chaque δ . Le prérequis d'une telle linéarisation est de disposer d'un ensemble d'incertitude discret énumérable, ce qui est notre cas particulier.

Coupes de Benders

Sur la structure matricielle générique précédemment définie, on peut adapter les coupes de Benders. Lorsque les variables de premier niveau C^{rob} et x sont fixées, il ne reste qu'une question de faisabilité : Existe t'il pour tous les scénarios δ un plan de production y_δ , satisfaisant les contraintes du problème et dont le coût se retrouve inférieur à C^{rob} . Cela fait autant de questions de faisabilités qu'il y a de scénarios et les sous problèmes sont décomposés. La question de la faisabilité des ensembles de contraintes (8.3), (8.4) et (8.10), peut se reformuler comme le problème d'optimisation :

$$\begin{aligned} \eta^* &= \min_{\eta, y_\delta \geq 0} \eta \quad (= -\max -\eta) \\ W y_\delta &\leq \eta \mathbb{1} + h - T_\delta x & (\alpha) \\ B y_\delta &\leq \eta \mathbb{1} + b & (\beta) \\ q y_\delta &\leq \eta + C^{rob} & (\gamma) \end{aligned} \quad (8.11)$$

$\mathbb{1}$ étant un vecteur colonne composé uniquement de 1. La réalisabilité de (8.3), (8.4) et (8.10) se ramène alors à $\eta^* = 0$. Le calcul d'optimisation correspondant au calcul de η^* est bien réalisable. On est alors dans les hypothèses du théorème de dualité forte. Le calcul de η^* se ramène alors à :

$$-\eta^* = \min_{\alpha, \beta, \gamma \geq 0} \alpha.(h - T_\delta x) + \beta.b + \gamma.C^{rob} \quad (8.12)$$

$$\alpha.W + \beta.b + \gamma.q \geq 0 \quad (y_\delta) \quad (8.13)$$

$$-\alpha.\mathbb{1} - \beta.\mathbb{1} - \gamma.\mathbb{1} \geq -1 \quad (\eta) \quad (8.14)$$

On reconnaît alors qu'il s'agit de la recherche de directions extrêmes de la matrice formée par les contraintes de (8.13), la contrainte (8.14) permettant la convexification. Cette matrice est non seulement indépendante des décisions de premiers niveaux, mais également indépendante de δ . Cela permet de reformuler le problème, pour tout x satisfaisant les contraintes (8.2), pour toute direction extrême (α, β, γ) définie par (8.13), on a :

$$\alpha.(h - T_\delta x) + \beta.b + \gamma.C^{rob} \geq 0 \quad (8.15)$$

Cela permet d'écrire ainsi le problème uniquement avec les variables x et C^{rob} , avec les contraintes (8.17) et toutes les coupes précédentes. Cette formulation comporte ainsi un nombre exponentiel de contraintes :

$$\min_{x \in \{0,1\}^n \times \mathbb{R}_+^m, C^{rob} \geq 0} \quad cx + C^{rob} \quad (8.16)$$

$$Ax \leq a \quad (8.17)$$

$$\forall(\alpha, \beta, \gamma) \quad \alpha.(h - T_\delta x) + \beta.b + \gamma.C^{rob} \geq 0 \quad (8.18)$$

De plus, cette section a fourni un algorithme de séparation pour générer une coupe pour toute solution partielle x . Dans un tel cadre robuste, la décomposition de Benders permet de générer une coupe pour chaque scénario. A une itération de génération de coupes, on peut explorer un nombre restreint de scénarios δ tant que cela génère des coupes. Ce n'est que pour prouver la convergence que le calcul de tous les sous-problèmes sur les différentes valeurs de δ est nécessaire.

8.1.4 Formulation mixte robuste et stochastique

On peut généraliser la formulation précédente en considérant des scénarios stochastiques s doivent être optimisés suivant l'espérance sur le second niveau, tandis que certains aléas δ doivent être traités de manière robuste, suivant un critère de pire cas. L'ensemble des scénarios est ici en fait $\delta \times \mathcal{S}$, on croise ces différents scénarios indépendants. Le pire cas, est alors calculé sur l'espérance des scénarios \mathcal{S} :

$$\begin{aligned} & \min_{x, C, y_\delta^s \geq 0} (cx + C) \\ s.t : & \quad Ax \leq a \quad (8.19.1) \\ & \quad \forall \delta \in \Omega, \quad \sum_s \pi_s q_s y_\delta^s \leq C \quad (8.19.2) \\ & \quad \forall (s, \delta) \in \mathcal{S} \times \Omega, \quad T_\delta x + W y_\delta^s \leq h \quad (8.19.3) \\ & \quad \forall (s, \delta) \in \mathcal{S} \times \Omega, \quad B y_\delta^s \leq b_s \quad (8.19.4) \\ & \quad x \in \mathbb{N}^m \times \mathbb{R}_+^n, C \end{aligned}$$

Cette écriture généralise le cas uniquement robuste, en ne considérant qu'un seul scénario stochastique s , et pour un seul scénario δ , il s'agit bien d'une reformulation. Les grandeurs stochastiques étant uniquement les coûts de production des unités non nucléaires, et leurs bornes de production, seul les vecteurs b_s et q_s dépendent des scénarios stochastiques.

Reformulation de Benders En adaptant la décomposition de Benders comme précédemment, avec les C et x comme variables de premier niveau, les coupes de Benders s'obtiennent en répondant la question de faisabilité suivante : Pour ces variables x , existe-t'il pour tout scénario δ des variables de second niveau y_δ^s , satisfaisant les contraintes (8.19.3) et (8.19.4) avec un coût stochastique inférieur à C ? Cela se ramène au problème d'optimisation suivant pour tout $\delta \in \Omega$:

$$\begin{aligned}
\eta^* &= \min_{\eta, y^s \geq 0} \eta & (= -\max_{\eta, y^s} -\eta) \\
\forall s \in \mathcal{S}, & \quad W y^s & \leq \eta \mathbf{1} + h - T_\delta x & (\alpha^s) \\
\forall s \in \mathcal{S}, & \quad B y^s & \leq \eta \mathbf{1} + b_s & (\beta^s) \\
& \quad \sum_s \pi_s q_s y_\delta^s & \leq \eta + C & (\gamma)
\end{aligned} \tag{8.20}$$

La question de faisabilité se ramenant à $\eta^* = 0$. Par dualité forte :

$$\begin{aligned}
-\eta^* &= \min_{\alpha^s, \beta^s \geq 0, \gamma \geq 0} \sum_s \alpha_s \cdot (h - T_\delta x) + \sum_s \beta_s \cdot b_s + \gamma \cdot C \\
& \quad \sum_s \alpha_s \cdot W + \sum_s \beta_s \cdot B + \sum_s \pi_s \gamma \cdot q_s \geq 0 & (y^s) & (8.21.1) \\
& \quad -\sum_s \alpha_s \cdot \mathbf{1} - \sum_s \beta_s \cdot \mathbf{1} - \gamma \cdot \mathbf{1} \geq -1 & (\eta) & (8.21.2)
\end{aligned} \tag{8.21}$$

On note \mathcal{D}_δ l'ensemble des directions extrêmes du cône défini par les contraintes (8.21.1). La caractérisation des points réalisables x, C pour le scénario δ , pour tout $(\alpha^s, \beta^s, \gamma) \in \mathcal{D}_\delta$, nous avons :

$$\varphi_\delta(\alpha^s, \beta^s, \gamma) = \sum_s \alpha_s \cdot (h - T_\delta x) + \sum_s \beta_s \cdot b_s + \gamma \cdot C \geq 0 \tag{8.22}$$

Cela donne la reformulation de Benders suivante :

$$\begin{aligned}
\min_{x \in \mathbb{N}^m \times \mathbb{R}_+^n, C \geq 0} & \quad cx + C \\
& \quad Ax \leq a \\
\forall (\alpha^s, \beta^s, \gamma) \in \mathcal{D}_\delta, & \quad \varphi_\delta(\alpha^s, \beta^s, \gamma) \geq 0
\end{aligned} \tag{8.23}$$

Avec cette reformulation, on décompose selon les scénarios de prolongations d'arrêt δ , mais pas selon les scénarios stochastiques s . La séparation traite un problème non décomposé selon s , une telle taille de problème ne doit pas être rédhibitoire.

Variante multicoupe De manière à élaborer une décomposition selon les scénarios stochastiques, on procède de similairement à la section 1.2.3. La reformulation est alors la suivante, en scindant dans l'objectif C les contributions de chaque scénario :

$$\begin{aligned}
& \min_{x, C, y_\delta^s \geq 0} (cx + \sum_s C_s) \\
s.t : & \quad Ax & \leq a & (8.19.1) \\
\forall (s, \delta) \in \mathcal{S} \times \Omega & \quad \pi_s q_s y_\delta^s & \leq C_s & (8.24.2) \\
\forall (s, \delta) \in \mathcal{S} \times \Omega, & \quad T_\delta x + W y_\delta^s & \leq h & (8.24.3) \\
\forall (s, \delta) \in \mathcal{S} \times \Omega, & \quad B y_\delta^s & \leq b_s & (8.24.4) \\
& \quad x \in \mathbb{N}^m \times \mathbb{R}_+^n
\end{aligned}$$

La structure en ayant fixée le premier niveau x est alors bloc diagonale avec des blocs indépendants selon les scénarios stochastiques et robustes. De manière analogue, les coupes de Benders pour une solution de premier niveau fixée (x, C_s) , et un couple de scénarios (s, δ) nécessitent de calculer les sous-problèmes :

$$\begin{aligned}
\eta^* &= \min_{\eta, y \geq 0} \eta & (= -\max_{\eta, y} -\eta) \\
& \quad W y & \leq \eta \mathbf{1} + h - T_\delta x & (\alpha) \\
& \quad B y & \leq \eta \mathbf{1} + b_s & (\beta) \\
& \quad \pi_s q_s y & \leq \eta + C_s & (\gamma)
\end{aligned} \tag{8.25}$$

Si $\eta^* = 0$ pour tout (s, δ) , la solution partielle (x, C_s) est prouvée optimale. Dans le cas contraire, les coupes suivantes sont ajoutées :

$$\alpha.(h - T_\delta x) + \beta.b_s + \gamma.C_s \geq 0 \quad (8.26)$$

8.2 Décomposition de Benders déployée sur le modèle robuste

Il s'agit à présent d'explicitier les coupes de Benders obtenues sur le modèle robuste. Nous considérons ici et dans la suite de ce chapitre le problème obtenu après agrégation des pas de temps à la semaine.

8.2.1 Contraintes d'ordonnancement en environnement robuste

On explicite tout d'abord les contraintes d'ordonnancement en environnement robuste. Pour les contraintes d'espacement CT16, la durée d'arrêt et donc les prolongations n'interviennent pas. Ces contraintes sont ainsi placées naturellement dans le problème maître. Dans le cas des contraintes des ressources CT19, il faudrait en toute rigueur savoir comment les prolongations d'arrêt impactent le temps d'utilisation $\mathbf{Tu19}_{i,k}$ ou le début d'utilisation $\mathbf{L19}_{i,k}$. Pour ne pas compliquer inutilement le modèle, on supposera que les prolongations d'arrêt n'ont pas d'incidence, les retards étant pris après l'utilisation de la ressource. Ces deux types de contraintes s'écrivent donc toujours :

$$\forall w, \sum_{(i,k) \in \mathbf{A16}} (d_{i,k,w} - d_{i,k,w - \mathbf{Se16}}) \leq 1$$

$$\forall w, \sum_{(i,k) \in \mathbf{A19}} (d_{i,k,w - \mathbf{L19}_{i,k}} - d_{i,k,w - \mathbf{L19}_{i,k} - \mathbf{Tu19}_{i,k}}) \leq \mathbf{Q19}$$

Pour les autres contraintes d'ordonnancement, la durée d'arrêt intervient. Cela nécessite d'écrire de telles contraintes pour tous les scénarios de prolongations d'arrêt pouvant survenir. On a ainsi les jeux de contraintes suivants pour les contraintes CT17, CT18, CT20 et CT21 :

$$\forall w, \delta, \sum_{(i,k) \in \mathbf{A17}} (d_{i,k,w - \mathbf{Da}_{i,k,\delta}} - d_{i,k,w - \mathbf{Da}_{i,k,\delta} - \mathbf{Se17}}) \leq 1$$

$$\forall w, \delta, \sum_{(i,k) \in \mathbf{A18}} (d_{i,k,w} - d_{i,k,w - \mathbf{Se18}}) + (d_{i,k,w - \mathbf{Da}_{i,k,\delta}} - d_{i,k,w - \mathbf{Da}_{i,k,\delta} - \mathbf{Se18}}) \leq 1$$

$$\forall w, \delta, \sum_{(i,k) \in \mathbf{A20}_w} (d_{i,k,w} - d_{i,k,w - \mathbf{Da}_{i,k,\delta}}) \leq \mathbf{N20}_w$$

$$\forall w, \delta, \sum_{i,k} \mathbf{Pmax}_i^w (d_{i,k,w} - d_{i,k,w - \mathbf{Da}_{i,k,\delta}}) \leq \mathbf{Imax}$$

La dépendance en δ inciterait à écrire toutes ces contraintes dans le problème esclave. Le fait que ces contraintes soient très redondantes du fait de scénarios de prolongations d'arrêt très proches incite à considérer dans le problème maître un de ces jeux de contraintes pour un scénario δ fixé. On pourra prendre le scénario nominal sans prolongation d'arrêt.

Le cas des contraintes d'espacement CT14 et CT15 est spécifique. A première vue, ces contraintes nécessitent comme précédemment de les dupliquer sur tous les scénarios de prolongations, en fait, on pourra se ramener à un cas déterministe pour ces contraintes. Comme il a été développé dans le chapitre 5, CT14 s'écrit en notant $W_{i,k}$ la semaine où débute l'arrêt (i, k) avec des contraintes disjonctives du type :

$$\forall \delta, (i, k), (i', k'), \quad W_{i,k} \leq W_{i',k'} + \mathbf{Da}_{i',k',\delta} + \mathbf{Se14} \quad \text{ou} \quad W_{i',k'} \leq W_{i,k} + \mathbf{Da}_{i,k,\delta} + \mathbf{Se14}$$

Ces inégalités étant vraies pour tout les scénario δ , en particulier, en considérant les scénarios donnant la prolongation maximale de l'arrêt (i, k) et de l'arrêt (i', k') :

$$\forall (i, k), (i', k'), \quad W_{i,k} \leq W_{i',k'} + \overline{\mathbf{Da}}_{i',k'} + \mathbf{Se14} \quad \text{ou} \quad W_{i',k'} \leq W_{i,k} + \overline{\mathbf{Da}}_{i,k} + \mathbf{Se14}$$

De tels jeux d'inégalités dominant toutes les réalisations possibles. On est alors ramené au cas déterministe en considérant toujours les prolongations maximales, et on a l'écriture suivante pour les contraintes CT14 :

$$\forall w, \quad \sum_{(i,k) \in \mathbf{A14}} (d_{i,k,w} - d_{i,k,w - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se14})+}) \leq 1 \quad (8.27)$$

Pour les contraintes CT15, les contraintes sont identiques, restreintes aux fenêtres de temps :

$$\forall w \in [\mathbf{d}_{15}, \mathbf{f}_{15}], \quad \sum_{(i,k) \in \mathbf{A15}} (d_{i,k,w} - d_{i,k,w - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se15})+}) \leq 1 \quad (8.28)$$

Un tel résultat présente des analogies avec l'approche robuste de Soysters, dans ce cas, les contraintes robustes se ramènent au cas déterministe en considérant tous les pires cas individuels.

8.2.2 Réécriture du modèle

Pour appliquer directement les coupes de Benders de la section 8.1.3, une réécriture des contraintes du problème en inégalités s'avère nécessaire. Lors du chapitre 5, les contraintes d'égalité impliquant les contraintes de stocks ont déjà été transformées en inégalités, en supprimant les variables de stocks résiduels de fin et de début de cycle, les définissant comme des combinaisons linéaires des variables de rechargement $r_{i,k}$ et des puissances. On gardera les notations $x_{i,k,\delta}^{init}$ et $x_{i,k,\delta}^{fin}$, en gardant à l'esprit qu'il s'agit d'expressions linéaires. Il reste alors les contraintes de demandes pour tout w, δ , $\forall \sum_{i,k} p_{i,k,\delta,w} + \sum_j p_{j,\delta,w} = \mathbf{Dem}^w$.

Pour avoir des contraintes d'inégalité, on supprime les variables associées à la première unité thermique $p_{0,\delta,w} = \mathbf{Dem}^w - \sum_{i,k} p_{i,k,\delta,w} - \sum_{j>0} p_{j,\delta,w}$ que nous considérerons alors comme une expression linéaire. De même que pour les stocks résiduels, on conservera la notation $p_{0,\delta,w}$ en gardant à l'esprit qu'il s'agit d'une expression linéaire. La positivité de $p_{0,\delta,w}$ est écrite ou dominée par la contrainte de puissance minimale. Le problème robuste à 2 étapes a alors l'écriture suivante :

$$\begin{aligned}
& \min_{d \in \{0;1\}^M, r, p, s \geq 0} && \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + C^{rob} \\
& \forall i, k, w, && d_{i,k,w-1} - d_{i,k,w} \leq 0 \\
& \forall i, k, w < \mathbf{To}_{i,k}, && d_{i,k,w} \leq 0 \\
& \forall i, k, w \geq \mathbf{To}_{i,k}, && -d_{i,k,w} \leq -1 \\
& \forall i, k, && \mathbf{Rmin}_{i,k} d_{i,k,W} - r_{i,k} \leq 0 \\
& \forall i, k, && r_{i,k} - \mathbf{Rmax}_{i,k} d_{i,k,W} \leq 0 \\
& \forall i, k, w, && d_{i,k+1,w} - d_{i,k,w} - \mathbf{Lmin}_{i,k} \leq 0 \\
& \forall i, k, w, && d_{i,k,w} - \mathbf{Lmax}_{i,k} - d_{i,k+1,w} \leq 0 \\
& \forall w, && \sum_{(i,k) \in \mathbf{A14}} (d_{i,k,w} - d_{i,k,w} - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se14})^+) \leq 1 \\
& \forall w \in [\mathbf{d15}, \mathbf{f15}], && \sum_{(i,k) \in \mathbf{A15}} (d_{i,k,w} - d_{i,k,w} - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se15})^+) \leq 1 \\
& \forall w, && \sum_{(i,k) \in \mathbf{A16}} (d_{i,k,w} - d_{i,k,w} - \mathbf{Se16}) \leq 1 \\
& \forall w, && \sum_{(i,k) \in \mathbf{A19}} (d_{i,k,w} - \mathbf{L19}_{i,k} - d_{i,k,w} - \mathbf{L19}_{i,k} - \mathbf{Tu19}_{i,k}) \leq \mathbf{Q19} \\
& \forall w, \delta, && \sum_{i,k} \mathbf{Pmax}_i^w (d_{i,k,w} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta}) \leq \mathbf{Imax} \\
& \forall w, \delta, && \sum_{(i,k) \in \mathbf{A20}_w} (d_{i,k,w} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta}) \leq \mathbf{N20}_w \\
& \forall w, \delta, && \sum_{(i,k) \in \mathbf{A18}} (d_{i,k,w} - d_{i,k,w} - \mathbf{Se18}) + (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - \mathbf{Se18}) \leq 1 \\
& \forall w, \delta, && \sum_{(i,k) \in \mathbf{A17}} (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - \mathbf{Se17}) \leq 1 \\
& \forall \delta, && \sum_{j,w} \mathbf{C}_{j,w}^{prd} \mathbf{D}^w p_{j,\delta,w} - \sum_i \mathbf{C}_i^{val} s_{i,\delta}^{fin} \leq C^{rob} \\
& \forall i, w, \delta, && p_{i,k,\delta,w} - \mathbf{Pmax}_i^w (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k+1,w}) \leq 0 \\
& \forall i, k, \delta, && -x_{i,k,\delta}^{fin} \leq 0 \\
& \forall i, k, \delta, && x_{i,k,\delta}^{init} \leq \mathbf{Smax}_{i,k} \\
& \forall i, k, \delta, && x_{i,k,\delta}^{fin} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) d_{i,k+1,W} \leq \mathbf{Amax}_{i,k} \\
& \forall i, k, \delta, && s_{i,\delta}^{fin} \leq x_{i,k,\delta}^{fin} + \mathbf{M}_i (1 + d_{i,k+1,W} - d_{i,k,W}) \\
& \forall j > 0, w, \delta, && -p_{j,\delta,w} \leq -\mathbf{Pmin}_{j,w} \\
& \forall j > 0, w, \delta, && p_{j,\delta,w} \leq \mathbf{Pmax}_{j,w} \\
& \forall w, \delta, && \sum_{i,k} p_{i,k,\delta,w} + \sum_{j>0} p_{j,\delta,w} \leq \mathbf{Dem}^w \\
& \forall w, \delta, && -\sum_{i,k} p_{i,k,\delta,w} - \sum_{j>0} p_{j,\delta,w} \leq \mathbf{Pmax}_{0,w} - \mathbf{Dem}^w
\end{aligned}$$

8.2.3 Problème maître

Le problème maître de la décomposition de Benders comprend les contraintes portant sur les variables r , d , où les prolongations d'arrêt δ n'interviennent pas. Avec la notation $A^{ordo} d \geq a^{ordo}$ pour les contraintes déterministes d'ordonnancement à prolongations nulles, le problème maître sans coupes de Benders s'écrit :

$$\begin{aligned}
& \min_{d \in \{0;1\}^M, r, C^{rob} \geq 0} \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + C^{rob} \\
& \quad A^{ordo} d \geq a^{ordo} \\
& \quad \forall i, k, \quad \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \\
& \quad \forall w, \quad \sum_{(i,k) \in \mathbf{A14}} (d_{i,k,w} - d_{i,k,w - (\overline{\mathbf{D}}\mathbf{a}_{i,k} + \mathbf{Se14})}) \leq 1 \\
& \quad \forall w \in [\mathbf{d}_{15}, \mathbf{f}_{15}], \quad \sum_{(i,k) \in \mathbf{A15}} (d_{i,k,w} - d_{i,k,w - (\overline{\mathbf{D}}\mathbf{a}_{i,k} + \mathbf{Se15})}) \leq 1
\end{aligned}$$

8.2.4 Coupe de Benders sans stretch

Suivant les calculs de la section 8.1, pour une solution partielle d, r, C^{rob} du problème maître et un scénario de prolongation δ , le problème de génération de coupes se ramène à la résolution du PL :

$$\begin{aligned}
& \min_{\eta, p \geq 0} \quad \eta \\
& \quad \forall i, k, w, \quad p_{i,k,w} - \mathbf{Pmax}_i^w (d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta}} - d_{i,k+1,w}) \leq \mathbf{Pmax}_i^w \eta \quad (\alpha_1) \\
& \quad \forall i, k, \quad s_{i,\delta}^{fin} \leq x_{i,k,\delta}^{fin} + \mathbf{Smax}_{i,k} (1 + d_{i,k+1,W} - d_{i,k,W}) + \mathbf{Smax}_{i,k} \eta \quad (\alpha_2) \\
& \quad \forall i, k, \quad x_{i,k}^{fin} - (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) d_{i,k+1,W} \leq \mathbf{Smax}_{i,k} + \mathbf{Smax}_{i,k} \eta \quad (\alpha_3) \\
& \quad \forall i, k, \quad -x_{i,k,\delta}^{fin} \leq \mathbf{Smax}_{i,k} \eta \quad (\alpha_4) \\
& \quad \forall i, k, \quad x_{i,k}^{init} \leq \mathbf{Smax}_{i,k} + \mathbf{Smax}_{i,k} \eta \quad (\alpha_5) \\
& \quad \forall w, \quad \sum_{(i,k) \in \mathbf{A17}} (d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta}} - d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta} - \mathbf{Se17}}) \leq 1 + \eta \quad (\alpha_{17}) \\
& \quad \forall w, \quad \sum_{(i,k) \in \mathbf{A18}} (d_{i,k,w} - d_{i,k,w - \mathbf{Se18}}) + (d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta}} - d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta} - \mathbf{Se18}}) \leq 1 + \eta \quad (\alpha_{18}) \\
& \quad \forall w, \quad \sum_{(i,k) \in \mathbf{A20}_w} (d_{i,k,w} - d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta}}) \leq \mathbf{N20}_w (1 + \eta) \quad (\alpha_{20}) \\
& \quad \forall w, \quad \sum_{i,k} \mathbf{Pmax}_i^w (d_{i,k,w} - d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta}}) \leq \mathbf{Imax} (1 + \eta) \quad (\alpha_{21}) \\
& \quad \forall j > 0, w, \quad -p_{j,w} \leq -\mathbf{Pmin}_{j,w} + \mathbf{Pmax}_{j,w} \eta \quad (\beta_1) \\
& \quad \forall j > 0, w, \quad p_{j,w} \leq \mathbf{Pmax}_{j,w} + \mathbf{Pmax}_{j,w} \eta \quad (\beta_2) \\
& \quad \forall w, \quad \sum_{i,k} p_{i,k,w} + \sum_{j>0} p_{j,w} \leq \mathbf{Dem}^w + \mathbf{Pmax}_{0,w} \eta \quad (\beta_3) \\
& \quad \forall w, \quad -\sum_{i,k} p_{i,k,\delta,w} - \sum_{j>0} p_{j,\delta,w} \leq \mathbf{Pmax}_{0,w} (1 + \eta) - \mathbf{Dem}^w \quad (\beta_4) \\
& \quad \sum_{j,w} \mathbf{C}_{j,w}^{prd} \mathbf{D}^w p_{j,w} - \sum_i \mathbf{C}_i^{val} s_i^{fin} \leq C^{rob} + \kappa \eta \quad (\gamma)
\end{aligned}$$

On notera que pour cette écriture, on a pondéré les η par des valeurs positives connues de l'ordre des valeurs des membres de la contrainte, sauf pour la contrainte de coût, où κ est à définir empiriquement (par exemple, la valeur d'une bonne solution du problème déterministe calculée avec les heuristiques du chapitre 6). De tels choix permettent de ne pas survaloriser l'importance de certains jeux de contraintes dans la génération de coupe de Benders. Le critère d'arrêt est d'avoir pour tout δ , une valeur nulle au PL précédent. Dans le cas contraire, les variables duales optimales générées par le calcul de PL fournissent les coupes de Benders suivantes où on rappelle la notation $a_{m,n} = \prod_{l=m}^n \frac{\mathbf{Q}_{i,l} - 1}{\mathbf{Q}_{i,l}}$:

$$\sum_{i,k,w} \alpha_{i,k,w}^{(1)} \mathbf{Pmax}_i^w (d_{i,k,w - \mathbf{D}\mathbf{a}_{i,k,\delta}} - d_{i,k+1,w})$$

$$\begin{aligned}
& + \sum_{i,k} \alpha_{i,k}^{(2)} \left(\mathbf{Smax}_{i,k} (1 + d_{i,k+1,W} - d_{i,k,W}) + \mathbf{q}_{0,k-1} \mathbf{Xi}_i + \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{Bo}_{i,l}) \right) \\
& + \sum_{i,k} \alpha_{i,k}^{(3)} \left(\mathbf{Smax}_{i,k} - (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) d_{i,k+1,W} - \mathbf{q}_{0,k-1} \mathbf{Xi}_i - \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{Bo}_{i,l}) \right) \\
& + \sum_{i,k} \alpha_{i,k}^{(4)} \left(\mathbf{q}_{0,k-1} \mathbf{Xi}_i + \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{Bo}_{i,l}) \right) + \sum_{i,k} \alpha_{i,k}^{(5)} \left(\mathbf{Smax}_{i,k} - \mathbf{q}_{0,k-1} \mathbf{Xi}_i - \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{Bo}_{i,l}) \right) \\
& + \sum_w \alpha_w^{(17)} \left(1 - \sum_{(i,k) \in \mathbf{A17}} (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - \mathbf{Se17}) \right) \\
& + \sum_w \alpha_w^{(18)} \left(1 - \sum_{(i,k) \in \mathbf{A18}} (d_{i,k,w} - d_{i,k,w} - \mathbf{Se18}) + (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - \mathbf{Se18}) \right) \\
& + \sum_w \alpha_w^{(20)} \left(\mathbf{N20}_w - \sum_{(i,k) \in \mathbf{A20}_w} (d_{i,k,w} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta}) \right) + \sum_w \alpha_w^{(21)} \left(\mathbf{Imax} - \sum_{(i,k)} \mathbf{Pmax}_i^w (d_{i,k,w} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta}) \right) \\
& - \sum_w \beta_w^{(1)} \mathbf{Pmin}_{j,w} + \sum_w \beta_w^{(2)} \mathbf{Pmax}_{j,w} + \sum_w \beta_w^{(3)} \mathbf{Dem}^w + \sum_w \beta_w^{(4)} (\mathbf{Pmax}_{0,w} - \mathbf{Dem}^w) + \gamma C^{rob} \geq 0
\end{aligned}$$

8.3 Formulations "robustifiées"

Face à la complexité de la formulation de la section 8.2, d'autres formulations robustes ont été envisagées. Ces formulations robustes peuvent fournir un comparatif sur les solutions obtenues, ou des solutions partielles et des coupes qui peuvent aider l'approche précédente.

8.3.1 Formulations de faisabilité d'ordonnancement robustes

La question originelle était d'incorporer d'avoir un ordonnancement robuste, que le planning des dates d'arrêt résiste aux prolongations d'arrêt vis à vis des contraintes d'ordonnancement. Une telle formulation robustifiée permet de n'avoir des variables de puissances que sur un scénario, ce que les résultats du chapitre 7 préconisent, le nombre de variables continues étant un facteur limitant :

$$\begin{aligned}
\min \quad & \sum_{i,k} C_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} C_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,w} C_{j,t}^{prd} \mathbf{D}^w p_{j,w} - \sum_i C_i^{val} x_i^{fin} \\
& M^{ordo} d \geq b^{ordo} \\
\forall i, k, \quad & \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \\
\forall w, c \in CT14 \quad & \sum_{(i,k) \in \mathbf{A14}^c} (d_{i,k,w} - d_{i,k,w} - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se14})^+) \leq 1 \\
\forall c \in CT15, w \in [\mathbf{d}_{15}^c, \mathbf{f}_{15}^c], \quad & \sum_{(i,k) \in \mathbf{A15}^c} (d_{i,k,w} - d_{i,k,w} - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se15})^+) \leq 1 \\
\forall i, k, \quad & x_i^{fin} \leq x_{i,k}^{fin} + M_i (d_{i,k,W} - d_{i,k+1,W}) \\
\forall j, w, \quad & \mathbf{Pmin}_{j,w}^s \leq p_{j,w} \leq \mathbf{Pmax}_{j,w}^s \\
\forall i, w, \quad & 0 \leq p_{i,k,\delta,w} \leq \mathbf{Pmax}_i^w (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k+1,w}) \\
\forall w, \delta, \quad & \sum_{i,k} p_{i,k,\delta,w} + \sum_j p_{j,\delta,w} = \mathbf{Dem}^w \\
\forall i, k, \delta, \quad & 0 \leq x_{i,k,\delta}^{init} \leq \mathbf{Smax}_{i,k} \\
\forall i, k, \delta, \quad & x_{i,k,\delta}^{fin} \leq \mathbf{Amax}_{i,k+1} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) (1 - d_{i,k+1,W}) \\
\forall w, \delta, \quad & \sum_{i,k} \mathbf{Pmax}_i^w (d_{i,k,w} - d_{i,k,w} - \mathbf{Da}_{i,k,\delta}) \leq \mathbf{Imax}
\end{aligned}$$

Pour la résolution d'une telle formulation, une approche frontale est envisageable pour un faible nombre de scénarios de prolongations d'arrêt. Une génération dynamique des contraintes d'ordonnement robustes violées est aussi envisageable, d'autant que ces contraintes peuvent avoir de très fortes redondances dans le cas où le scénario de prolongations d'arrêt sont très proches (ce qui est le cas avec notre définition d'ensemble d'incertitude). La décomposition de Benders fournit une séparation, et les coupes de Benders générées sont valides pour la formulation de la section 8.2.

Au vu des faibles nombres de contraintes CT17,CT18,CT20 et CT21 et de leur importance moindre opérationnellement par rapport aux contraintes CT14 et CT15, considérer de la robustesse sur les contraintes CT14 et CT15 permet d'obtenir une formulation de taille similaire au problème déterministe. Les techniques de résolution frontale exactes et heuristiques du chapitre 6 s'y appliquent également, permettant de fournir une approche calculable pour les tailles d'instances réelles.

8.3.2 Anticipations maximales robustes

Dans la section précédente, l'accent avait été mis sur l'ordonnement robuste, et sur la faisabilité des contraintes de CT14 à CT21 aux prolongations d'arrêt, en négligeant les impacts des niveaux de production sur la faisabilité globale du planning. Une infaisabilité que les variables et contraintes de premier niveau ne peuvent pas voir est liée aux contraintes d'anticipation maximales : si le planning entre deux arrêts consécutifs (i, k) et $(i, k + 1)$ est trop serré, la consommation de combustible nucléaire sera insuffisante, même à puissance maximale pour atteindre un niveau de fuel avant le rechargement inférieur au niveau $\mathbf{Amax}_{i,k+1}$. On peut écrire de telles contraintes uniquement avec les variables d et r , le cas le plus critique étant à puissance maximale. On va écrire ici des contraintes pour exprimer qu'à puissance maximale, les variables d et r doivent garantir le respect de l'anticipation maximale. Mathématiquement, cela se prouve par minoration successives. La contrainte d'anticipation maximale porte sur les stocks en fin de cycles, conditionnellement au début de l'arrêt suivant sur l'horizon de temps : $x_{i,k,\delta}^{fin} + (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) d_{i,k+1,W} \leq \mathbf{Smax}_{i,k}$. On rappelle l'expression de $x_{i,k,\delta}^{fin}$ en fonction des variables de puissances et de rechargement :

$$x_{i,k,\delta}^{fin} = \mathbf{q}_{0,k-1} \mathbf{X} \mathbf{i}_i + \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{q}_{l+1,l+1} p_{i,l,\delta,w} - \mathbf{Bo}_{i,l}) - \sum_w \mathbf{D}^w p_{i,k,\delta,w}$$

Or $p_{i,l,\delta,w} \leq \mathbf{Pmax}_i^w (d_{i,l,w} - \mathbf{Da}_{i,l,\delta} - d_{i,l+1,w})$. Comme $\mathbf{q}_{m,n} \geq 0$, on a la minoration suivante, où l'égalité est obtenue pour une production toujours égale à la puissance maximale, qui nous intéresse particulièrement pour avoir les domaines critiques pour la contrainte d'anticipation maximale :

$$\begin{aligned} x_{i,k,\delta}^{fin} \geq & \mathbf{q}_{0,k-1} \mathbf{X} \mathbf{i}_i - \sum_w \mathbf{D}^w \mathbf{Pmax}_i^w (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k+1,w}) \\ & + \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{Bo}_{i,l} - \mathbf{q}_{l+1,l+1} \mathbf{Pmax}_i^w (d_{i,l,w} - \mathbf{Da}_{i,l,\delta} - d_{i,l+1,w})) \end{aligned} \quad (8.29)$$

En combinant (5.15) et (8.29), on obtient la minoration suivante pour tout (i, k, δ) , qui fournit des coupes sur les variables d et r permettant de remonter des infaisabilités vue des contraintes d'anticipation maximale :

$$\begin{aligned} \mathbf{Smax}_{i,k} \geq & (\mathbf{Smax}_{i,k} - \mathbf{Amax}_{i,k+1}) d_{i,k+1,W} - \sum_w \mathbf{D}^w \mathbf{Pmax}_i^w (d_{i,k,w} - \mathbf{Da}_{i,k,\delta} - d_{i,k+1,w}) \\ & + \mathbf{q}_{0,k-1} \mathbf{X} \mathbf{i}_i + \sum_{l=0}^{k-1} \mathbf{q}_{l+1,k-1} (r_{i,l+1} - \mathbf{Bo}_{i,l} - \mathbf{q}_{l+1,l+1} \mathbf{Pmax}_i^w (d_{i,l,w} - \mathbf{Da}_{i,l,\delta} - d_{i,l+1,w})) \end{aligned} \quad (8.30)$$

On remarque que ces coupes ne doivent être écrites pour (i, k) donnés que sur les scénarios où une prolongation non nulle est considérée pour un arrêt de la centrale i sur un cycle antérieur à k , ce qui limite la combinatoire des contraintes à ajouter. Dans le cas où on considère que des scénarios où un unique arrêt a une prolongation non nulle (comme introduit section 8.1.1), le nombre de telles contraintes est de l'ordre de $|IK^2|$, négligeable devant les jeux de contraintes nécessitant d'écrire plus de contraintes.

Ces contraintes peuvent introduire de la robustesse induite de contraintes de production, dans les formulations de la section 8.3.2 sans trop perturber la combinatoire et les capacités de résolution. Ces coupes sont de plus utiles à incorporer dans le problème maître de la décomposition de Benders de la section 8.2 ou 8.3.1, pour réduire le nombre de coupes d'infaisabilités générées.

8.3.3 Paramétrisation de la robustesse des contraintes CT14 et CT15

Sur les formulations précédentes, l'ensemble des solutions réalisables se devaient d'être robustes à toutes les prolongations d'arrêt. Une telle approche peut ainsi fournir des solutions optimales très conservatives, au coût prohibitif (ou dans le cas le plus défavorable, il se pourrait que le problème soit infaisable), de manière similaire à l'approche robuste de Soyster. Similairement à la modélisation robuste de Ben-Tal et Nemirovski, on cherche à paramétrer le niveau de robustesse. On considère de la robustesse que sur les contraintes CT14 et CT15, ce qui fournit un modèle de taille similaire au modèle déterministe. Les contraintes robustes sont écrites en ajoutant des variables continues positives pour toute contrainte $c \in CT14$ $z_{c,w}^{(14)}$ (et respectivement $z_{c,w}^{(15)}$ pour $c \in CT15$), permettant de violer la contrainte écrite avec les prolongations d'arrêt, au prix d'une pénalisation dans la fonction objectif de \mathbf{Cpen}^{rob} . On aboutit alors à la formulation suivante :

$$\min_w \quad \sum_w \mathbf{Cpen}^{rob}(z_{c,w}^{(14)} + z_{c,w}^{(15)}) + f_{obj}^{det} \quad (8.31)$$

$$M^{ordo} d \geq b^{ordo} \quad (8.32)$$

$$\forall i, k, \quad \mathbf{Rmin}_{i,k} d_{i,k,W} \leq r_{i,k} \leq \mathbf{Rmax}_{i,k} d_{i,k,W} \quad (8.33)$$

$$\forall j, t, \quad \mathbf{Pmin}_{j,t} \leq p_{j,t} \leq \mathbf{Pmax}_{j,t} \quad (8.34)$$

$$\forall i, t, \quad 0 \leq p_{i,k,t} \leq \mathbf{Pmax}_i^t(d_{i,k,w_t} - \mathbf{Da}_{i,k} - d_{i,k+1,w_t}) \quad (8.35)$$

$$\forall t, \quad \sum_{i,k} p_{i,k,t} + \sum_j p_{j,t} = \mathbf{Dem}^t \quad (8.36)$$

$$\forall i, k, \quad x_i^{fin} \leq x_{i,k}^{fin} + M_i(d_{i,k,W} - d_{i,k+1,W}) \quad (8.37)$$

$$\forall w, c \in CT14 \quad \sum_{(i,k) \in \mathbf{A14}^c} (d_{i,k,w} - d_{i,k,w} - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se14})^+) \leq 1 + z_{c,w}^{(14)} \quad (8.38)$$

$$\forall c \in CT15, w \in [\mathbf{d}_{15}^c, \mathbf{f}_{15}^c], \quad \sum_{(i,k) \in \mathbf{A15}^c} (d_{i,k,w} - d_{i,k,w} - (\overline{\mathbf{Da}}_{i,k} + \mathbf{Se15})^+) \leq 1 + z_{c,w}^{(15)} \quad (8.39)$$

$$z_{c,w}^{(15)}, z_{c,w}^{(14)} \geq 0 \quad (8.40)$$

où $f_{obj}^{det} = \sum_{i,k} \mathbf{C}_{i,k}^{rld} r_{i,k} + \sum_{i,k,w} \mathbf{C}_{i,k,w}^{pen} (d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,t} \mathbf{C}_{j,t}^{prd} \mathbf{D}^t p_{j,t} - \sum_i \mathbf{C}_i^{val} x_i^{fin}$ est la fonction objectif du modèle déterministe.

Un tel modèle peut fournir deux types d'approches d'implémentation. En faisant varier le coefficient \mathbf{Cpen}^{rob} , on oriente la robustesse des solutions, la robustesse des solutions optimales et d'autant plus forte (ie le nombre de violations robustes est d'autant plus faible) que le coefficient \mathbf{Cpen}^{rob} est élevé. Cela permet de tracer des courbes de Pareto arbitrant les surcoûts de solutions au niveau de robustesse, et de permettre à un opérateur la solution de bon compromis.

Une seconde application permet de mettre en oeuvre l'approche "light robustness". Une première étape est de calculer la valeur C^{det} du problème déterministe (une excellente solution fournie de manière heuristique suffit pour les grandes instances, l'approche VNS du chapitre 6 est alors utilisée pour ce faire).

En se donnant un surcoût maximum α (par exemple 0,1%) autorisé par rapport au coût de la solution déterministe précédemment calculée, on recherche alors la solution la plus robuste avec un surcoût inférieur à α . Dans le modèle précédent, on prend un paramètre \mathbf{Cpen}^{rob} très élevé, et on ajoute la contrainte $f_{obj}^{det} \leq (1 + \alpha)C^{det}$.

8.4 Résultats, conclusions et perspectives

	nbVar	nbContr	temps	gap opt	Non satisfaites
A1_3_120	2787	825	0,08	0,04%	0
A2_3_120	5107	1307	0,25	0,10%	4
A3_3_120	4930	1208	0,41	0,13%	4
A4_3_120	7920	1878	0,41	0,16%	9
A5_3_120	9240	4497	86,9	0,21%	0
B6_3_120	9698	2708	7,5	0,70%	5
B7_3_120	11292	4902	133	0,19%	0
B8_3_120	13570	9198	891	0,48%	8
B9_3_120	14688	11000	3600	0,60%	3
B10_3_120	10474	3685	8,8	0,74%	3
X11_3_120	9932	2912	43,2	0,29%	6
X12_3_120	10909	4092	63	0,41%	0
X13_3_120	10432	3995	13,9	0,26%	1
X14_3_120	10775	4384	43,9	0,21%	2
X15_3_120	10709	3967	12,2	0,28%	2
A1	6014	2152	1,8	0,04%	0
A2	11520	4313	2,8	0,13%	6
A3	10973	3885	2,6	0,14%	4
A4	18844	7553	74,1	0,13%	9
A5	20543	11622	3600	0,19%	1

TABLE 8.1 – Caractéristiques de la convergence du modèle robuste 8.3.3 en minimisant les violations robustes

Les premiers résultats sur la formulation robuste de la section 8.3.1 donnaient beaucoup d'infaisabilités, même en considérant l'ensemble d'incertitude minimaliste de $|I|$ scénarios de prolongations, où juste le premier arrêt d'une tranche nucléaire a une prolongation d'une semaine, comme le montre le tableau 8.1. Les contraintes d'ordonnancement sont ainsi trop dimensionnantes pour considérer un modèle robuste à 100%. Ce constat a ainsi incité l'élaboration de la modélisation de la section 8.3.3.

Un tel constat rendait la modélisation de la section 8.2 également vouées à de trop nombreuses infaisabilités. Relaxer la robustesse des contraintes d'ordonnancement ne répond pas au besoin opérationnel, les considérations de robustesse étaient surtout introduites pour des raisons de faisabilité, les surcoûts robustes étant une question secondaire sans les contraintes d'ordonnancement, les surcoûts étant même a priori très faibles. Ces raisons ont conduit à ne pas poursuivre l'implémentation de la décomposition de Benders de 8.2.

Notons toutefois les problèmes d'instabilités numériques liés à la méthode de la décomposition de Benders. De tels problèmes sont apparus sur un modèle jouet très simplifié, même sur des instances de très petites tailles. Les instabilités numériques portant sur les arrondis des variables duales données par les sous-

problèmes se reportent dans le problème maître et s'ajoutent. Ainsi, pour des variables duales engendrant des coefficients très petits dans les coupes de Benders (inférieurs à 10^{-6}), ces coefficients étaient arrondis à 0, et donc non écrits dans le problème maître. Cela est particulièrement critique lorsque cela arrive sur le coefficient de la variable C^{rob} , toujours très petits par rapports aux autres coefficients des variables de décision d'arrêt, des coupes d'optimalité devenaient alors des coupes de faisabilité, coupant des solutions réalisables du problème au lieu d'impliquer une minoration du coût de solution.

Pour contourner de telles difficultés, l'idée était de changer l'échelle des unités des coûts et des recharges en les normalisant, pour ne pas avoir de trop fortes disparités dans les ordres de grandeur des coefficients. Cela amenait à considérer alors les variables $\tilde{r}_{i,k}$ et \tilde{C}^{rob} telles que $\tilde{r}_{i,k} = \mathbf{Rmax}_{i,k} r_{i,k}$ et $C^{rob} = C^0 \tilde{C}^{rob}$ où C^0 est représentative de l'ordre de grandeur du coût de la solution, assez bien connu en pratique ou on peut prendre le coût donné par une bonne solution heuristique donnée par le chapitre 5.

L'approche paramétrique de robustesse des contraintes d'ordonnancement, plus simple conceptuellement et dans sa résolution fournit des résultats plus utilisables pour une application industrielle. Tout d'abord, les caractéristiques de la résolution frontale sont peu différentes du modèle frontal déterministe comme l'analyse le tableau 8.1. La résolution des instances de taille réelle (jeux B et X non tronqués) est ainsi réalisable à l'aide des heuristiques VNS du chapitre 6, en partant d'une solution réalisable que les heuristiques du chapitre 6 peuvent aussi fournir. Les solutions obtenues par l'approche robuste paramétrique avec une très forte pondération sur la robustesse fournit des solutions avec un très faible surcoût par rapport à l'optimum déterministe comme le montre le tableau 8.1. Nous avons déjà remarqué que de nombreuses solutions du modèle déterministe avaient des solutions proches, la résolution robuste paramétrique permet d'orienter le choix parmi les solutions les plus robustes. Un tel procédé casse des pseudo symétries, impliquées par des solutions de coût proches, ce qui est toujours bénéfique pour une résolution de type Branch&Bound.

En faisant varier les paramètres de la fonction objectif valorisant la robustesse, on a pu tracer des courbes de Pareto arbitrant le surcoût de solutions au niveau de robustesse mesuré en nombre de violations des contraintes d'ordonnancement robustes, comme sur la figure 8.1. Trois configurations bien distinctes sont apparues. Sur l'instance B7_3_120, une solution d'ordonnancement robuste à 100% existe et présente un faible surcoût (0,2% par rapport à l'optimum sans robustesse). La décroissance est régulière, un choix de solution opérationnel est alors guidé sur le seuil de surcoût autorisé. Sur l'instance X11_3_120, une solution d'ordonnancement robuste à 100% existe et présente un surcoût de 0,3% par rapport à l'optimum sans robustesse. La décroissance est très irrégulière, s'autoriser jusqu'à 6 violations robustes n'a que très peu d'effet sur le coût de solution, à partir de 7, la diminution du surcoût est nette avec une décroissance plus régulière. Sur l'instance X12_3_120, une solution d'ordonnancement robuste à 100% n'existe pas, la solution la plus robuste présente un surcoût de 0,4% par rapport à l'optimum sans robustesse. La décroissance est ensuite très rapide, en s'autorisant peu de violations robustes supplémentaires, le surcoût de solution diminue nettement, ce qui ferait préconiser de telles solutions. Dans tous ces cas, de telles courbes de Pareto s'avèrent intéressantes pour orienter le choix de solutions à implémenter opérationnellement, pour arbitrer le niveau de robustesse implémenté. On remarque que les surcoûts de robustesse sont très faibles. Les enjeux liés aux techniques de résolutions et choix de voisinages de recherche locale fournissent des surcoûts bien plus importants. Une approche préconisée serait donc de calculer d'abord une solution du problème déterministe avec la VNS du chapitre 6 pour ensuite utiliser le modèle de la section 8.3.4 pour chercher des solutions robustifiées autour de la solution déterministe, et arbitrer le niveau de robustesse souhaité en fonction des surcoûts.

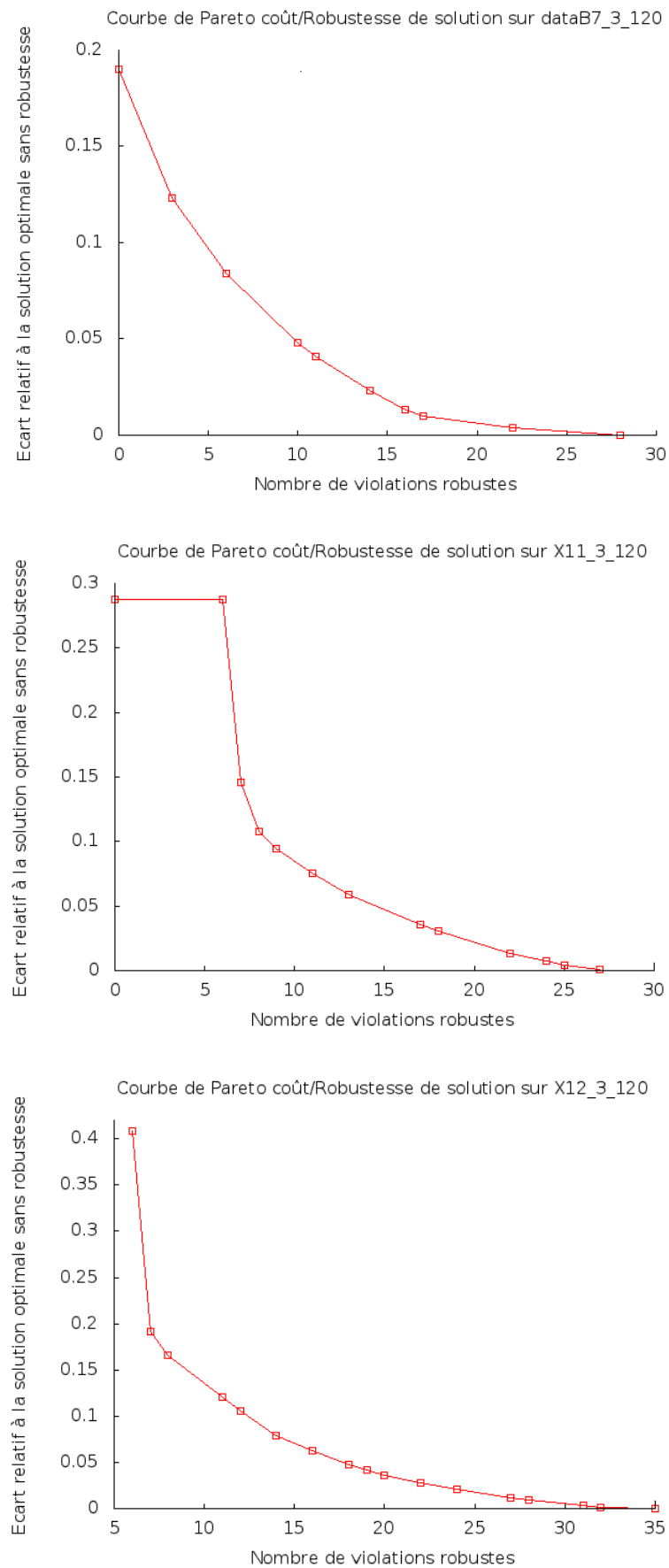


FIGURE 8.1 – Courbes de Pareto coûts de solution/niveau de robustesse sur les instances B7-3-120, X11-3-120 et X12-3-120

Quatrième partie

Conclusions et perspectives

Conclusions

Cette thèse a été fructueuse tant au plan méthodologique que sur les problèmes d'applications. Nous synthétisons ici les différents apports de la thèse.

Contributions générales

EDF ayant une forte culture de l'optimisation continue, une motivation initiale de la thèse était d'étudier les approches présentées au chapitre 1 et de les appliquer aux problèmes d'EDF. Cette thèse a aussi eu des apports non quantifiables dans les discussions internes à EDF, pour y faire avancer la réflexion collective, mais aussi par des présentations en interne des méthodologies du chapitre 1. Un piège de la PLNE est la facilité de modélisation, qui occulte la complexité de calcul engendrée ou l'impact de la taille de données. Ainsi, même un problème polynomial présente des limites quand la taille des données augmente. Dans l'esprit de ma thèse, il n'était pas question de faire des approximations suivant un besoin de méthode autre que la PLNE, pour faire rentrer un problème dans le canevas d'une méthode très spécifique.

Un écueil est de voir les méthodes de décomposition de Benders et de Dantzig-Wolfe comme des solutions miracles et génériques, là où leur cadre d'application ou d'utilisation efficace admet des hypothèses précises. La reformulation de Dantzig-Wolfe n'est pas une méthode universelle pour des problèmes où des contraintes couplent des sous-problèmes indépendants, dépendant du compromis entre temps de calcul et valeurs des bornes duales. La décomposition de Benders présente de grands challenges en espace mémoire et en arrondis numériques comme nous l'avons expérimenté au chapitre 8. Finalement, l'approche qui a eu les meilleurs résultats sur cette thèse a été de conception simple, en hybridant VNS et petits calcul de PLNE, une pratique naturelle pour des ingénieurs en Recherche Opérationnelle.

Contributions spécifiques sur le problème UCPd

Cette thèse a fourni des contributions pour le problème UCPd, poursuivant des pistes de recherches entrevues avec [51]. Le travail de reformulation des formulations compactes, uniquement empirique dans le cadre de [51], a été poursuivi avec des résultats théoriques pour expliquer les qualités des descriptions polyédrales. Ce travail a permis d'obtenir trois formulations polyédralement équivalentes. Les comparaisons empiriques de leur convergence Branch&Bound ont fourni des résultats sur l'implémentation efficace, la résolution frontale Branch&Bound préconisée combinant les avantages de ces trois formulations polyédralement équivalentes.

Les résultats numériques de la décomposition de Dantzig-Wolfe sur le problème UCPd ont été décevants, il en reste que ce travail m'a permis de mieux comprendre la méthode et de nourrir un questionnement. De tels résultats sont dûs à la modélisation des contraintes du parc thermique à flamme. L'utilisation de la PLNE en mathématiques s'avère plus efficace pour une résolution en temps contraint. Avoir une intuition des bonnes solutions d'un problème permet d'améliorer la résolution aveugle et sophistiquée que sont les heuristiques primales de Branch&Bound.

Contributions sur le problème de planification des arrêts de maintenances et rechargement des centrales nucléaires

Une première étape de la thèse a modélisé en PLNE le problème de planification des arrêts de maintenances et rechargement des centrales nucléaires conformément au besoin opérationnel à EDF. La modélisation du challenge ROADEF présentait a posteriori de nombreuses faiblesses vis-à-vis du besoin réel, de longs travaux préparatoires ont été nécessaires pour arriver à maturité dans la modélisation. Dans notre modélisation, on a déclaré des variables binaires uniquement pour les dates d'arrêts. Les contraintes de stretch sont alors approximées, et de considérer uniquement une borne supérieure. Cette dernière approximation ne s'est pas avérée critique, la production nucléaire étant plus rentable que les autres productions thermiques. L'impact sur les coûts de solution en négligeant le stretch a pu être chiffré.

Le choix des méthodes de résolution était une question à mûrir. La décomposition de Dantzig-Wolfe n'est pas adaptée à un tel cadre de programmation stochastique. Même dans le cadre déterministe, notre modélisation rentre dans le cadre théorique où la génération de colonnes n'apporte pas de gain de qualité des bornes duales. Une approche frontale Branch&Bound est alors plus adaptée avec un temps de calcul de bornes plus faibles, tout en permettant d'utiliser l'intelligence développée dans Cplex.

La modélisation compacte en PLNE a été travaillée pour repousser les limites des calculs réalisables. L'utilisation de variables binaires différentes des SOS naturelle s'est avérée plus judicieuse, menant à des branchements plus efficaces avec Cplex. Notre travail a aussi chiffré l'intérêt de supprimer des variables binaires par preprocessing, ou de fournir une solution initiale réalisable comme démarrage à chaud de l'algorithme de Branch&Bound. La taille des données est critique pour le calcul de relaxation continue malgré l'agrégation des scénarios et des pas-de-temps. Notre travail de reformulation et les améliorations de Cplex au cours de la thèse ont permis de repousser ces limites pour aboutir à des calculs abordables en PLNE sur les données de taille réelle, utilisées opérationnellement à EDF avec l'outil ORION. L'algorithme de VNS introduit pour le problème UCPd reposant sur une formulation PLNE s'est avéré d'une efficacité redoutable sur toutes les instances. Nos développements sur la VNS ont également une portée opérationnelle en permettant de chiffrer les impacts des différents choix de voisinages sur les minimums locaux de la recherche locale. On a ainsi justifié et chiffré la nécessité d'avoir des voisinages "de saut d'hiver" dans la recherche locale.

Comme il a été remarqué au cours du challenge ROADEF, beaucoup de solutions ont des coûts très proches. Il est intéressant de savoir favoriser les solutions qui seraient préconisées opérationnellement, en exhibant des critères de choix préconisés et en développant une approche multi-critère. Pour ce faire, deux critères ont été analysés dans cette thèse, un critère de stabilité pénalisant la distance à la solution courante issue de l'optimisation précédente, et une approche favorisant des solutions plus robustes. Des courbes de Pareto ont été tracées pour arbitrer le "coût de stabilité" ou le "surcoût du à la robustesse" d'un planning. La robustesse a été incorporée initialement avec la définition académique usuelle, et cela a mené à des résultats trop conservatifs et souvent à la non existence de solutions robustes. Une définition robuste plus proche des considérations opérationnelles a alors été exhibée, avec une résolution traitant des problèmes de taille similaire au problème déterministe, et permettant de paramétrer le niveau de robustesse. Des courbes de Pareto arbitrant le surcoût de solution face au niveau de robustesse ont été tracées, permettant de guider des choix opérationnels de bon compromis.

Si notre travail a été guidé par les considérations opérationnelles et a ainsi pris des distances par rapport au challenge ROADEF, cette thèse a toutefois pu s'y appuyer et y revenir. Nous avons ainsi pu améliorer significativement les meilleures bornes duales de la littérature sur le problème et les instances du challenge ROADEF.

Perspectives ouvertes par la thèse

Après mes contributions par cette thèse, de nombreuses perspectives sont ouvertes. Tout d'abord, les progrès continus des algorithmes de résolution PLNE et les augmentations des capacités de calcul engendrent une continuelle amélioration des performances ou des tailles de problèmes pouvant être résolus. Ce qui n'est pas réalisable à l'heure actuelle pourrait devenir une réalité d'ici quelques années, de tels progrès se sont avérés décisifs pendant ma thèse. Des perspectives plus spécifiques restent ouvertes.

Perspectives méthodologiques générales

L'approche générique induite par l'utilisation de VNS sur des voisinages PLNE est très efficace sur les deux applications de la thèse, malgré des structures et des caractéristiques de convergence PLNE très différentes. L'avantage d'une telle approche est sa facilité d'implémentation à partir d'une modélisation PLNE quelconque, ce qui est intéressant pour une application industrielle de la PLNE. Une perspective serait d'élargir à de nouveaux problèmes d'application, pour tester les limites de la méthode sur des problèmes plus banalisés de la littérature avec des jeux de données éprouvés. VND, la variante la plus simple de VNS, était suffisante sur nos applications, convergeant vers des minimums locaux d'excellente qualité. Sur des applications où cela ne serait pas le cas, des mécanismes de diversification et de mémoire seraient indispensables dans la méthode.

Perspectives pour l'UCP à court terme

Sur la problématique de la production court terme, les résultats montrent que l'on peut aller plus loin dans la modélisation en conservant une résolution efficace. Dans le cadre du stage de master, il avait déjà été remarqué que l'on pouvait ajouter et désagréger des points de fonctionnements. Les points de fonctionnements extrêmes de nos données étaient en effet à $PCoMax$ et $PCoMin$, les points $Pmin$, $Pmax$ n'étaient pas dans nos données. De même, l'extension à la réserve tertiaire s'avérait limitante avec les heuristiques frontales de PLNE. Une utilisation des VNS dans un tel cadre serait sans doute intéressante. Enfin, notre travail n'a jamais modélisé d'unités hydrauliques, qui fournissent les contraintes les plus difficiles et handicapantes pour les solveurs de PLNE. Il serait intéressant d'envisager l'approche VNS à voisinages PLNE sur un tel problème d'application.

Perspectives pour la planification des maintenances nucléaires

Sur la problématique de planification des maintenances et des rechargements des centrales nucléaires, les premiers tests montraient des limites en mode déterministe, avec un seul scénario stochastique. Les derniers tests menés ont montré que l'on pouvait considérer plusieurs scénarios, au moins trois, sur toutes les instances B8 et B9 comprises. En particulier, envisager un scénario moyenné et les scénarios de plus grande/plus basse demande cumulée serait sans doute intéressant pour planifier les dates d'arrêts.

Notre travail a chiffré l'impact d'hypothèses comme la relaxation du stretch ou comme la prise en compte de la robustesse et d'une solution initiale en termes de qualité de solution finale et d'impact de résolution. Toutefois, cela a été évalué sur des données non confidentielles, volontairement modifiées pour des raisons de confidentialité. Notre travail a fourni une méthodologie pour chiffrer de tels impacts, offrant une perspective pour EDF R&D de les chiffrer sur données réelles.

Enfin, notre approche n'a pas fourni de bornes primales pour le challenge ROADEF, il serait intéressant de projeter nos solutions de dates d'arrêts et de rechargements avec le critère de coût du challenge ROADEF. Une telle perspective fait office de validation théorique.

Comme perspectives opérationnelles, la validation de notre modèle dans un contexte simulé de réoptimisation dynamique n'a pas été effectué. Analyser l'impact de notre définition de la robustesse dans un tel processus simulé est une perspective opérationnelle majeure ouverte par cette thèse, à valider sur données réelles.

Perspectives pour la DGA

Ma thèse a été au final majoritairement financée par la DGA, en tant que parcours de formation par la recherche. Une telle politique s'inscrit dans une démarche de long terme, où je suis engagé envers l'état permettant un retour sur l'investissement, et il me tient à coeur que la DGA puisse tirer profit de ma formation par la recherche.

Tout d'abord, la formation par la recherche est assurément un projet humain de formation qui m'a beaucoup apporté, et dont la DGA pourra tirer profit. La dimension humaine de la thèse, à mener à bien un projet pendant 3-4 ans, avec des hauts, des bas avec des moments de doutes, la difficulté de finaliser en temps au final assez court, est assurément une formation adéquate pour des personnes amenées à être des gestionnaires de projets complexes. Au plan méthodologique, la démarche de recherche scientifique diffère de la formation classique d'ingénieur, apportant une formation très complémentaire, et des nouveaux réflexes d'études et de comparaison à l'état de l'art. Au vue de la dimension internationale des grands projets industriels, la DGA a tout intérêt à voir en son sein des docteurs pour collaborer avec des nations où le doctorat a eu reconnaissance plus élevée qu'en France. Il me paraît important de souligner la dimension internationale d'une thèse de doctorat. Le fait de participer à des congrès internationaux, de faire des présentations orales en Anglais, de rédiger en anglais des résumés pour les conférences et des articles pour des revues spécialisées était aussi très formateur, cette dimension internationale me sera assurément profitable pour la suite de mon parcours à la DGA.

Outre ces aspects universels de la formation par la recherche, des perspectives se dessinent spécifiquement en Recherche Opérationnelle dans la suite de mon parcours à la DGA. Ces perspectives sont déjà entamées comme j'ai rédigé ma thèse cette dernière année en parallèle de mon nouveau poste à la DGA, ce qui me permet d'y voir plus clair. Tout d'abord, mes connaissances en Recherche Opérationnelle sont mises à profit du service de la DGA effectuant des études technico-opérationnelle pour de l'enseignement en formation interne en recherche opérationnelle, ce dont les service a un fort besoin. Mon poste actuel est en unités de gestion de projet, où l'Ingénierie Système est déployée pour modéliser dessinent systèmes industriels complexes et les processus humains associés. La RO et l'Ingénierie Système comportent des interfaces naturelles, la RO apportant des outils de modélisation et de résolutions pour des problématiques concrètes industrielles. En particulier, sur les problématiques courantes de la gestion de projet à la DGA, où des décisions complexes sont prises face à de nombreuses incertitudes, les méthodologies développées pour ma thèse en ordonnancement robuste pourraient s'appliquer.

Cinquième partie

Annexes

Résultats annexes des chapitres 5 à 7

Cet annexe regroupe des tableaux de résultats et illustrations relatifs au challenge ROADEF et aux développements des chapitre 5, 6 et 7.

Instances	Valeur	Rco	Inf0	Sup0	temps Rco	temps racine	temps total
A1_3.120	76,03	0,00%	0,00%	0,00%	0,01	0,1	0,1
A2_3.120	71,96	0,04%	0,00%	0,01%	0,03	0,4	0,4
A3_3.120	76,18	0,10%	0,00%	0,00%	0,02	0,2	0,2
A4_3.120	50,93	0,05%	0,01%	0,01%	0,08	1,25	2,2
A5_3.120	58,15	0,91%	0,53%	0,28%	0,92	8,3	55,4
B6_3.120	31,61	2,08%	0,32%	0,19%	0,28	2	24,9
B7_3.120	29,63	3,48%	1,48%	0,24%	0,78	6,1	2016
B8_3.120	29,107	11,79%	3,89%	1,25%	4,65	19,3	3600
B9_3.120	28,84	11,89%	5,62%	2,01%	8,5	35,3	3600
B10_3.120	26,9	2,57%	0,26%	0,15%	0,43	3,7	3600
A1	152,17	0,17%	0,09%	0,00%	0,06	0,75	0,9
A2	145,2	0,21%	0,08%	0,02%	0,16	1,8	5,4
A3	152,58	0,35%	0,12%	0,01%	0,11	1,55	2,6
A4	102,42	0,95%	0,53%	0,09%	2,1	11	323,6
A5	119,78	0,90%	0,63%	0,51%	9,6	46	3600
B6	76,97	2,23%	0,61%	0,51%	9,65	45,5	3600
B7	73,53	2,90%	1,54%	-	123,5	317	3600
B8	-	-	-	-	3600	-	3600
B9	-	-	-	-	3600	-	3600
B10	69,5	3,45%	1,35%	0,27%	12,5	41,8	3600

TABLE 8.2 – Structure de convergence Branch&Bound sans stretch en 1h avec Cplex 12.3

Instances	Valeur	Frontal 1h	HeurSimpl	u(3,1)	Cycle	1 sautsHiv	VNS
B6	76966M	0	3,22	0,13	0,31	0,26	0,04
B7	74233M	0,02	3,68	0,35	0,55	0,34	0,01
B8	73239M	NS	52,54	16,05	35,79	1,18	0,91
B9	72812M	NS	27,46	7,54	14,48	1,31	0,78
B10	69501M	0,02	0,21	0,02	0,08	0,05	0,03
Total B	366753M	NS	17,4	4,8	10,21	0,63	0,41
X11	73018M	0	0,45	0,22	0,30	0,16	0,04
X12	70604M	0	0,25	0,07	0,15	0,08	0,04
X13	69230M	0,04	7,37	0,93	2,43	0,51	0,12
X14	68395M	0,02	6,02	0,45	1,40	0,28	0,16
X15	66028M	0,06	0,12	0,06	0,10	0,09	0,05
Total X	347277M	0,02	2,82	0,34	0,87	0,22	0,08
Total B et X	714030M	NS	10,31	2,63	5,67	0,43	0,25

TABLE 8.3 – Résultats d’une première VNS avec des petits voisinages

Instances	CT14-15	CT16	CT17	CT18	C19	CT20	CT21
A1	7	1	3	0	1	1	1
A2	13	1	3	0	1	1	1
A3	12	1	3	2	1	1	1
A4	19	1	3	0	1	1	1
A5	18	1	3	0	1	1	3
B6	73	1	3	0	1	50	5
B7	66	1	3	0	1	50	5
B8	82	1	3	0	1	50	5
B9	82	1	3	0	1	50	5
B10	82	1	3	0	1	50	5
X11	73	1	3	0	1	50	5
X12	66	1	3	0	1	50	5
X13	82	1	3	0	1	50	5
X14	82	1	3	0	1	50	5
X15	82	1	3	0	1	50	5

FIGURE 8.2 – Contraintes d’ordonnancement du challenge ROADEF

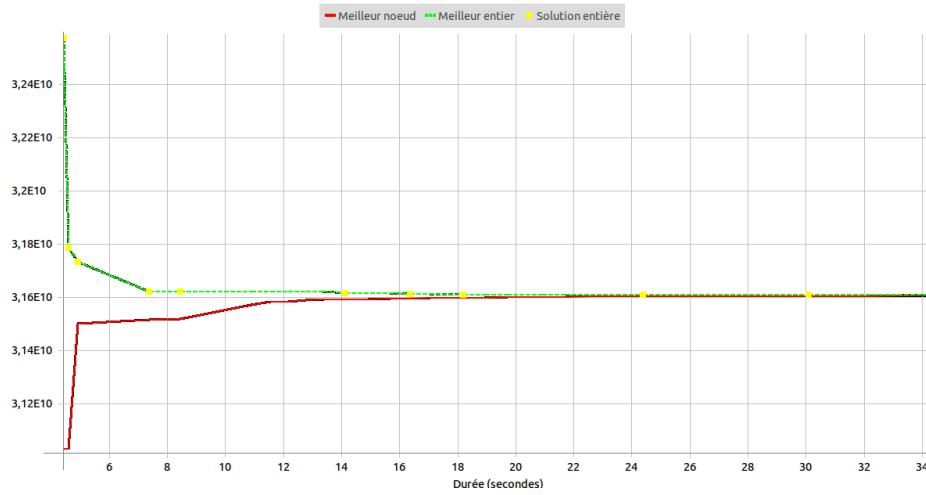


FIGURE 8.3 – Structure de convergence exacte sur B6-3-120

rang	id	score	score/5	B sets solved
1	Team S22 Frédéric Gard	1.160%	0.232%	5
2	Team S24 Johan Peekst	1.200%	0.240%	5
3	Team S23MT Laurent Alfano	7.170%	1.434%	5
4	Team S23 Laurent Alfano	7.273%	1.455%	5
5	Team J06 Niels Kjeldser	10.633%	2.127%	5
6	Team S21 Christoph Dür	23.523%	4.705%	5
7	Team J08 Roman Steine	59.942%	11.988%	5
8	Team S04 Mauro Dell'An	65.103%	13.021%	5
9	Team S14 Valentin Web	72.634%	14.527%	5
10	Team S08 Cor Hurkens	145.811%	29.162%	4
11	Team S17 François Sour	175.470%	35.094%	5
12	Team S16 Hadrien Camt	277.804%	55.561%	4
13	Team J05 Lauri Ahlroth,	531.785%	106.357%	5
14	Team S10 Bjørn Peterse	8633.089%	1726.618%	5
15	Team S10MT Bjørn Peterse	8633.189%	1726.638%	5
16	Team J16 Stefan Heinz1	9253.559%	1850.712%	4
17	Team S11 Wi Davide Anghil	11664.910%	2332.982%	3
18	Team S22MT Frédéric Gard	17119.158%	3423.832%	2
19	Team S25 Haris Gavran	17290.279%	3458.056%	0

FIGURE 8.4 – Résultats du challenge ROADEF sur les grandes instances publiques

Rank	Qualified teams	Score*	Team Id
1	Bouygues e-lab/LIF Marseille (Gardi, Estellon, Nouioua)	0.02%	S22
2	4C Cork (Cambazard, Hebrard, O'Sullivan)	0.05%	S16
3	Univ. Southern Denmark (Kjeldsen, Jensen, Godskensen, Schmidt, Larsen)	0.17%	J06
4	Univ. Hong Kong (Chan Hou, Peng, Zizhen, Haoquan)	0.51%	S26
5	Helsinki Univ. Technology (Ahlroth, Tokola, Schumacher)	0.58%	J05
6	Vienna Univ. Technology (Steiner, Pirkwieser, Prandtstetter)	0.65%	J08
7	Eindhoven Univ. Technology (Hurkens)	1.17%	S08
8	ICCOR (Dell'Amico, Diaz)	1.59%	S04
9	Zuse Institute Berlin (Heinz, Menne, Schlechte, Spiegelberg, Winkler)	2.10%	J16
10	BelImproved (Peekstok, Kuipers)	2.28%	S24
11	École Polytechnique de Montréal, Canada (Soumis, Desaulniers, Gendreau, Rousseau, Lessard, Raymond)	2.39%	S17
12	Technical Univ. Denmark (Petersen, Flindt Muller, Lusby)	2.76%	S10
13	Sarajevo Univ. (Gavranovic, Buljubasic, Catibusic, Numagic)	3.29%	S25
14	GSCOP Grenoble (Weber, Darlay, Naves, Kieffer)	5.12%	S14
15	ESSEC/LVMT/LIPN Paris (Alfandari, Chemla, Létocart, Turri, Rozenknop, Woffler Calvo)	5.64%	S23
16	Kocaeli Univ. (Figlali, Tasbogaz, Yavuz, Terzi)	7.19%	S19
17	Genova Univ. / Institut della Molla y Intelligenza Artificiale (Anghinolfi, Antero, Paolucci, Gambardella, Montemani, Toklu)	7.61%	S11
18	Kocaeli Univ. (Cihan, Tacettin, Yilmaz)	7.90%	J11
19	Polytechnique/Bouygues e-lab (Durr, Jost, Savourey, Taouti, Jeanjean)	9.49%	S21
20 ^(a,c)	Université de Bordeaux/INRIA Bordeaux (Joncour, Chicoisne, Labat, Miller, Pesneau, Sadykov, Stauffer, Vanderbeck)	0.41% ^(c)	S07
21 ^(a,b)	Eindhoven Univ. Technology (Firat)	1.60% ^(b)	J17

* average distance from best solution on data instance set A

(a) qualified by jury decision (b) on 4 instances out of 6

(c) on 5 instances out of 6.

FIGURE 8.5 – Résultats de la phase qualificative du challenge ROADEF

Communications et publications réalisées

Rédaction et soumission d'articles scientifiques

- N. Dupin, *Integer Linear Programming reformulations of the Unit Commitment Problem with minimum stop constraints*. Soumis à EJCO. Contenu : chapitre 3
- N. Dupin, E.G. Talbi, *Solving industrial problems with VNS and MILP neighborhoods. Application to energy management problems*. Preprint. Contenu : chapitres 4 et 7.
- En preparation N. Dupin, E.G. Talbi, *MILP formulation and new dual bounds for the EURO/ROADEF 2010 challenge*. Preprint. Contenu : chapitre 6.
- En preparation N. Dupin, E.G. Talbi, *Adjustable Robust Scheduling to maintain and refuel French nuclear power plants*. Preprint. Contenu : chapitre 8.
- En preparation N. Dupin, E.G. Talbi, *MILP matheuristics and VNS to solve a very constrained multiple depot VRPTW*. Résultats de JPOC8 et applications VNS. Preprint.

Communications dans des congrès

- **ROADEF 2010** congrès de l'Association Française de Recherche Opérationnelle et d'Aide à la Décision, Saint Etienne, 2-4 mars. N. Dupin, P. Bendotti, T. Simovic, *Problème d'ordonnancement de la production d'électricité des centrales thermiques, modélisation PLNE*.
Communication récompensée avec le Prix ROADEF Jeune chercheur.
- **7èmes Journées Polyèdres et Optimisation Combinatoire (JPOC7)**, Valenciennes, 8 - 10 juin 2011. N. Dupin, P. Bendotti, *Reformulation en PLNE des contraintes de durées minimales de paliers des centrales électriques*.
- **Conference on Optimization and Practices in Industry (COPI'11)**, Clamart, 23-25 nov, 2011. N. Dupin, P. Bendotti, *ILP resolution of Unit Commitment problem with minimum stop constraints*.
- **ROADEF 2011**, congrès de l'Association Française de Recherche Opérationnelle et d'Aide à la Décision, Angers, 11-13 avril. N. Dupin, P. Bendotti, M. Porcheron, *Planification des Arrêts des Réacteurs nucléaires : Extensions du Challenge EURO/ROADEF 2010*.
- **2nd International Symposium on Combinatorial Optimization (ISCO)**, Athens, avril 17-21 2012 N. Dupin, *A 2-stage robust optimization model for planning nuclear maintenances with uncertain durations on the maintenance*.
- **European Conference on Operations Research (EURO 2012)**, Vilnius 8-11 juillet 2012, N. Dupin, P. Bendotti, M. Porcheron, *Towards a multi stage Robust Formulation for the Nuclear Reactor Outage Scheduling Problem*.
- **International Conference on Metaheuristics and Nature Inspired Computing (META 2012)**, Port El-Kantaoui, 27-31 Oct 2012, *A parallel VNS scheme with ILP neighbourhoods. Application to a discrete Unit Commitment Problem*.
- **8èmes Journées Polyèdres et Optimisation Combinatoire (JPOC8)**, Clermont-Ferrand, 26

- 28 juin 2013. N. Dupin, *ILP formulations of a variant of multiple depot VRPTW, using modern Branch&Bound solvers.*
- **Conference PGMO**, Palaiseau, 4 oct 2013. N. Dupin, F. Clautiaux, *Formulations et résolutions pour le problème déterministe de placement des arrêts de rechargement et maintenance des tranches nucléaires.*
- **ROADEF 2014**, congrès de l'Association Française de Recherche Opérationnelle et d'Aide à la Décision, Bordeaux, 26 - 28 mars. N. Dupin, *A parallel VNS scheme with ILP neighbourhoods. Application to industrial problems : Unit Commitment Problems and VRPTW.*
- **International Conference on Metaheuristics and Nature Inspired Computing (META 2014)**, Marrakech, 27-31 Oct 2014, N. Dupin, F. Clautiaux, E.G.Talbi *Applying VNS with ILP neighbourhoods on EURO/ROADEF 2010's challenge scheduling problem of nuclear power plants' outages and refuelings.*
- **ROADEF 2015**, congrès de l'Association Française de Recherche Opérationnelle et d'Aide à la Décision, Marseille, 25 - 27 février. N. Dupin, E.G.Talbi *Éléments de réponses méthodologiques et opérationnelles pour le challenge EURO/ROADEF 2010. S*

Présentations en interne dans le cadre de la thèse

- INRIA Bordeaux, groupe de travail RO, 10 novembre 2011, *Modeling Uncertainty in Linear Programs : Stochastic and Robust Programming*
- INRIA Bordeaux, groupe de travail RO, 6 décembre 2011, *ILP resolution of Unit Commitment Problem with minimum stop constraints*
- INRIA Lille, 7 février 2012, *PhD Thesis Presentation : Scheduling nuclear plants outages, mixing exact and heuristic methods*
- EDF, R&D-OSIRIS, Clamart, 11 décembre 2012, *Décomposition de Benders et applications à la résolution de programmes linéaires incertains*
- EDF, R&D-OSIRIS, Clamart, 11 décembre 2012, *Programmation linéaire, Décomposition de Dantzig-Wolfe et résolution de programmes linéaires en nombre entiers.*
- Séminaire des doctorants de maths de Bordeaux, 19 décembre 2012, *Polyèdres, optimisation combinatoire, et applications pratiques.*
- Conférence de maths au Lycée Henri IV, Paris, 4 juin 2013, *Applications industrielles des maths, que faire après la prépa ?.*

Vie de thésard

Derrière des équations froides et interminables, on se rend rarement compte de l'aventure humaine que représente une thèse, je tenais à donner un visage humain du thésard dans mon manuscrit. Pour partager des états d'âme universels comme des moments de doutes, des hauts et des bas, des questionnements, de la solitude et de l'espérance, je partage ces quelques vers écrits en déplacements de thèse pour m'évader.

Thésard, hanté par le temps qui passe et qui passe,
Tout en train d'écouter tristement la trotteuse,
Tes arts, rendent et parlent tant, qu'ils passent et qu'ils passent !
Tout entrain dégoûté, triste ment la trotteuse.

Thésard, tu pensais avoir du temps en trois ans,
Définir ton sujet, faire un état de l'art,
Puis tes contributions. Mais là tu es bizarre,
Tu réalises, tu avances trop lentement.

Thésard, dans ta tour d'ivoire, tu broies du noir.
Tes arts ? Mathématiques. Tes armes ? Polygones.
Tu veux être rockstar ? Mais tu es antistar !
Et tu attends Godot ? Mais va, cherche Antigone !

Thésard, perdu sans ton phare, tu as le cafard.
Sur le grand tableau noir, pourquoi la craie crisse-t'elle ?
Souris à tes élèves, et présente toi sans fard.
Pourquoi bats tu de l'aile ? Enfin, la vie est belle !

Thésard mathématicien, pourquoi tant de doutes ?
Face à l'inconnu(e), la peur, rien ne te rassure.
Thésard dans le brouillard, malgré ce soleil d'août,
Tu dois te dépasser. Aie confiance, sois sûr !

Thésard, chercheur en Recherche Opérationnelle,
Cela n'est peu parlant. C'est sûr, tu n'es pas bête.
Qui peut dire que le domaine où tu excelles,
Ce sont les polyèdres à multiples facettes ?

Thésard polytechnicien, c'est une étiquette,
et des préjugés qu'assurément tu détestes,
Machine à calculer sans sensibilité ?
A cela mène la superficialité.

Thésard, vois tu comment sont reconnus tes arts ?
Pour l'informaticien, ils sont mathématiques.
Mais finalement, ton domaine est bien bâtard,
Car pour le matheux pur, c'est de l'informatique !

Thésard désenchanté, et ce soir tu enrages,
Tu veux tout arrêter, maintenant, tu y penses.
Ce cri de désespoir qu'autour de toi tu lances
Est bien inutile, tu cherches du courage !

Thésard, rappelle toi quel fut ton enthousiasme,
Lorsque tu appris que tu décrochais ta thèse.
Face au renoncement, sors toi de ce marasme,
Tu l'avais tant désiré, cette noble ascèse.

Thésard, ton histoire a commencé à Clamart,
Puis de gare en gare, Paris-Lille-Bordeaux,
Ton entrain pour les conférences t'accapare,
Si bien que partout, tu te sens dans ton bureau.

Thésard rêveur, tu mènes la vie de bohème.
Tu voyages avec du papier et un crayon.
Quand même ! Ces voyages, cette vie, tu l'aimes !
La liberté, guidée par le son du violon.

Thésard solitaire, et libre comme l'air.
Alors au gré du vent, de tes déplacements,
Ton imaginaire de rêveur rêve en vers,
D'une utopie qui se dessine tout doucement.

Thésard oisif, la baie d'Arcachon te ressource.
Et si tu es amer, la mer sèche tes larmes,
Protectrice, telle une mère qui s'alarme
Près de Royan, ta mère, c'est un retour aux sources.

Thésard amoureux, c'est ta plus grande richesse,
L'amour des maths, du travail bien fait, Notre Dame !
Le doctorat, c'est bien l'amour de la sagesse,
Et qui existe grâce à l'amour d'une Femme !

Thésard, va de l'avant, il ne reste qu'un an !
Tes contributions maintenant bien en place,
Expérimente ! Publie ! Ecris bravement
Ces fastidieux chapitres et que rien ne te lasse !

Doctorant, ton labeur et tes doctes erreurs,
Te font avancer, et combler tes espérances.
Il est grand temps de penser à ta soutenance,
Doucement, tu te métamorphoses en Docteur !

Bibliographie

- [1] Aarts, E., de Bont, F., Habers, J., van Laarhoven, P. : A parallel statistical cooling algorithm. *Lecture Notes in Computer Science* **210**, 87–97 (1986)
- [2] Achterberg, T. : SCIP - a framework to integrate constraint and mixed integer programming. Tech. rep., ZIB Report (2004)
- [3] Achterberg, T. : Conflict analysis in mixed integer programming. *Discrete Optimization* **4**(1), 4–20 (2007)
- [4] Achterberg, T., Berthold, T. : Improving the feasibility pump. *Discrete Optimization* **4**(1), 77–86 (2007)
- [5] Adams, W., Sherali, H. : A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems . *Management Science* **32**(10), 1274–1290 (1986)
- [6] Adams, W., Sherali, H. : Linearization Strategies for a Class of Zero-One Mixed Integer Programming Problems. *Operations Research* **38**(2), 217–226 (1990)
- [7] Ait-Alamara, M., Sourd, F., Ali-Aloulou, M. : Approches réactives robustifiées pour l’ordonnancement des trains sur une voie unique en présence d’aléas. in *Proc ROADEF 2010*
- [8] Andrieu, L. : Optimisation sous contrainte en probabilité. Ph.D. thesis, Ecole des Ponts ParisTech (2004)
- [9] Anghinolfi, D., Gambardella, L., Montemanni, R., Nattero, C., Paolucci, M., Toklu, N. : A mathuristic algorithm for a large-scale energy management problem. *Lecture Notes in Computer Science* **7116**, 173–181 (2012)
- [10] Apparigliato, R. : Règles de décision pour la gestion du risque : Application à la gestion hebdomadaire de la production électrique. Ph.D. thesis (2008)
- [11] Arroyo, J., Carrion, M. : A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE transactions on power systems* **21**(3) (2006)
- [12] Arroyo, J., Conejo, A. : A parallel repair genetic algorithm to solve the unit commitment problem. *IEEE transactions on power systems* **17**(4) (2002)
- [13] Aruldoss, T., Ebenezer, A. : A tabu search based hybrid optimization approach for a fuzzy modelled unit commitment problem. *Electric Power Systems Research* 76 pp. 413–425 (2006)
- [14] Atamtürk, A. : Strong formulations of robust mixed 0-1 programming. *Mathematical Programming* **108**(2-3), 235–250 (2006)
- [15] Atamtürk, A., Zang, M. : Two-stage network flow and design under demand uncertainty. *Operations Research* **55**(4), 662–673 (2007)
- [16] Averbakh, I. : Minmax regret solutions for minimax optimization problem with uncertainty. *Operations Research Letters* **27**(1), 57–65 (2000)
- [17] Bai, L., Rubin, P. : Combinatorial Benders cuts for the minimum tollbooth problem. *Operations Research* **57**(6), 1510–1522 (2009)
- [18] Balci H.H., V.J. : Scheduling electric power generators using particle swarm optimization combined with the lagrangien relaxation method. *Int. J. Appl. Math. Comput. Sci* **14**(3), 411–421 (2004)
- [19] Barahona, F., Anbil, R. : The volume algorithm : producing primal solutions with a subgradient method. *Mathematical Programming* **87**(3), 385–399 (2000)
- [20] Beale, E.M.L., Tomlin, J.A. : Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *OR* **69**(447-454), 99 (1970)

- [21] Ben-Ameur, W., Neto, J. : A constraint generation algorithm for large scale linear programs using multiple-points separation. *Mathematical programming* **107**(3), 517–537 (2006)
- [22] Ben-Ameur, W., Neto, J. : Acceleration of cutting planes and column generation algorithms : Application to network design. *Networks* **49**(1), 3–17 (2007)
- [23] Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A. : Adjustable robust solutions of uncertain linear programs. *Mathematical Programming* **99**, 351–376 (2004)
- [24] Benoist, T. : Décompositions combinatoires et applications industrielles. Lavoisier- Hermès (2007)
- [25] Benoist, T., Gaudin, E., Rottembourg, B. : Constraint Programming Contribution to Benders Decomposition : A Case Study. *Lecture Notes in Computer Science* **2470**, 603–617 (2002)
- [26] Bertacco, L., Fischetti, M., Lodi, A. : A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization* **4**(1), 63–76 (2007)
- [27] Berthold, T. : Primal heuristics for mixed integer programs. Master’s thesis (2006)
- [28] Bertsimas, D., Litvinov, E., Sun, X.A., Zhao, J., Zheng, T. : Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems* **28**(1), 52–63 (2013)
- [29] Bertsimas, D., Sim, M. : Robust discrete optimization and network flows. *Mathematical Programming* **98**(1-3), 49–71 (2003)
- [30] Bertsimas, D., Sim, M. : The price of robustness. *Operations research* **52**(1), 35–53 (2004)
- [31] Bixby, E., Fenelon, M., Gu, Z., Rothberg, E. : MIP : Theory and practice—closing the gap. *System Modelling and Optimization IFIP* **46**, 19–49 (2000)
- [32] Bonnans, J.F., Gaubert, S. : Recherche opérationnelle, aspects mathématiques et applications. Cours polycopié de l’École Polytechnique (2006)
- [33] Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A. : Numerical optimization : theoretical and practical aspects. Springer Science & Business Media (2006)
- [34] Brandt, F. : Solving a large-scale energy management problem with varied constraints. Ph.D. thesis (2010)
- [35] Brandt, F., Bauer, R., Völker, M., Cardeneo, A. : A constraint programming-based approach to a large-scale energy management problem with varied constraints. *Journal of Scheduling* **16**(6), 629–648 (2013)
- [36] Briant, O., Lemaréchal, C., Meurdesoif, P., Michel, S., Perrot, N., Vanderbeck, F. : Comparison of bundle and classical column generation. *Mathematical Programming* **113**(2), 299–344 (2008)
- [37] Büsing, C., D’Andreagiovanni, F. : New results about multi-band uncertainty in robust optimization. *Experimental Algorithms* **7276**, 63–74 (2012)
- [38] Cahon, S., Melab, N., Talbi, E.G. : Paradiseo : A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics* **10** (3), 357–380 (2004)
- [39] Caroe, C., Tind, J. : L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming* **83**, 451–464 (1998)
- [40] Valério de Carvalho, J.M. : Using extra dual cuts to accelerate column generation. *INFORMS Journal on Computing* **17**(2), 175–182 (2005)
- [41] Chandran, B., Balakrishnan, H. : A dynamic programming algorithm for robust runway scheduling. *American Control Conference* pp. 1161 – 1166 (2007)
- [42] Cheng, C., Liu, C., Liu, C. : Unit commitment by lagrangian relaxation and genetic algorithms. *IEEE transactions on power systems* **15**(2) (2000)
- [43] Clautiaux, F. : Bornes inférieures et méthodes exactes pour le problème de bin-packing en deux dimensions avec orientation fixe. Ph.D. thesis (2005)
- [44] Congram, R., Potts, C., van de Velde, S. : An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing* **14**(1), 52–67 (2002)
- [45] Cotta, C., Troya, J.M. : Embedding branch and bound within evolutionary algorithms. *Applied Intelligence* **18**, 137–153 (2003)

- [46] Danna, E., Rothberg, E., Pape, C.L. : Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming Ser. A* **102**, 71–90 (2005)
- [47] Dorigo, M., Maniezzo, V., Coloni, A. : Ant system : optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics* **26**(1), 29–41 (1996)
- [48] Dorigo, M., Stützle, T. : The Ant Colony Optimization Metaheuristic : Algorithms, Applications, and Advances. *International Series in Operations Research & Management Science* **57**, 250–285 (2003)
- [49] Dubost, L., Gonzalez, R., Lemaréchal, C. : A primal-proximal heuristic applied to the French Unit-commitment problem. *Mathematical Programming* **104**(1), 129–151 (2005)
- [50] Dupin, N. : Adaptation optimisée des horaires des trains en présence de travaux. SNCF Innovation& Recherche Tech Report (2009)
- [51] Dupin, N. : Optimisation conjointe des réserves aux pointes et du programme journalier pour un parc de production d'électricité thermique à flamme. EDF R&D Tech Report (2010)
- [52] Feillet, D. : A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR : A Quarterly Journal of Operations Research* pp. 407–424 (2010)
- [53] Fischetti, M., Glover, F., Lodi, A. : The feasibility pump. *Math Programming* **104**(1), 91–104 (2005)
- [54] Fischetti, M., Lodi, A. : Local branching. *Math Programming* **98**(1-3), 23–47 (2003)
- [55] Fischetti, M., Monaci, M. : Light robustness. *Lecture Notes in Computer Science* **5868**, 61–84 (2009)
- [56] Fischetti, M., Salvagnin, D., Zanette, A. : Fast approaches to improve the robustness of a railway timetable. *Transportation Science* **43**(3), 321–335 (2009)
- [57] Gardi, F., Nouioua, K. : Local search for mixed-integer nonlinear optimization : a methodology and an application. *Lecture Notes in Computer Science* **6622**, 167–178 (2011)
- [58] Gavranović, H., Buljubasić, M. : A hybrid approach combining local search and constraint programming for a large scale energy management problem. *RAIRO Operations Research* **47**(4), 481–500 (2013)
- [59] Glover, F. : Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* **13**, 533–549 (1986)
- [60] Glover, F., Laguna, M. : Tabu search. Kluwer Academic Publishers (1997)
- [61] Gondran, M., Minoux, M. : Graphes et algorithmes (1978)
- [62] Gondzio, J., González-Brevis, P. : A new warmstarting strategy for the primal-dual column generation method. *Mathematical Programming* pp. 1–34 (2014)
- [63] Gondzio, J., Gonzalez-Brevis, P., Munari, P. : New developments in the primal–dual column generation technique. *European Journal of Operational Research* **224**, 41–51 (2013)
- [64] Gorge, A., Lisser, A., Zorgati, R. : Stochastic nuclear outages semidefinite relaxations. *Computational Management Science* **9**(3), 363–379 (2012)
- [65] Gu, Z., Nemhauser, G.L., Savelsbergh, M.W. : Lifted cover inequalities for 0-1 integer programs : Computation. *INFORMS Journal on Computing* **10**(4), 427–437 (1998)
- [66] Guéret, C., Prins, C., Sevaux, M. : Applications of optimization with Xpress-MP (1999)
- [67] Hansen, P., Mladenović, N., Perez, J.M. : Variable neighborhood search : Principles and applications. *Annals of Operations Research* **175**(1), 367–407 (2010)
- [68] Hernandez-Perez, H., Salazar-Gonzalez, J. : A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics* **145**(1), 126–139 (2004)
- [69] Herroelen, W., Leus, R. : Robust and reactive project scheduling : a review and classification of procedures. *int. j. prod. res.* **42**(8), 1599–1620 (2004)
- [70] Hladik, P.E., Cambazard, H., Déplanche, A.M., Jussien, N. : Solving a real-time allocation problem with constraint programming. *The Journal of Systems and Software* **81**, 132–149 (2008)
- [71] Hooker, J., Osorio, M. : Mixed logical-linear programming. *Discrete Applied Mathematics* **96-97**, 395–442 (1999)

- [72] Iyengar, G. : Robust Dynamic Programming. *Mathematics of Operations Research* **30**(2), 257–280 (2005)
- [73] Jacquin, S., Jourdan, L., Talbi, E.G. : Dynamic Programming Based Metaheuristic for Energy Planning Problems. *Lecture Notes in Computer Science* **8602**, 165–176 (2014)
- [74] Joncour, C. : Problèmes de placement 2D et application à l’ordonnancement : modélisation par la théorie des graphes et approches de programmation mathématique. Ph.D. thesis (2010, pp 147-166)
- [75] Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., Vanderbeck, F. : Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics* **36**, 695–702 (2010)
- [76] Jong, Y.Y., Lee, M. : Acceleration of Benders decomposition for Mixed Integer Linear Programming. 4th International Symposium on Advanced Control of Industrial Processes pp. 222–227 (2011)
- [77] Jost, V., Savourey, D. : A 0-1 integer linear programming approach to schedule outages of nuclear power plants. *Journal of Scheduling* **16**(6), 551–566 (2013)
- [78] Kall, P., Wallace, S.W. : Stochastic programming. John Wiley & Sons, Chichester (1994)
- [79] Kawas, B., Koc, A., Laumanns, M., Lee, C., Marinescu, R., Mevissen, M., Taheri, N., van den Heever, S., Verago, R. : Unified framework and toolkit for commerce optimization under uncertainty. *IBM Journal of Research and Development* **58**(5) (2014)
- [80] Khemmoudj, M. : Modélisation et résolution de systèmes de contraintes : application au problème de placement des arrêts et de la production des réacteurs nucléaires d’EDF. Ph.D. thesis (2007)
- [81] Kirkpatrick, S., Gelatt, C., Vecchi, M. : Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
- [82] Kouvelis, P., Daniels, R., Vairaktarakis, G. : Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions* **32**, 421–432 (2000)
- [83] Laporte, G., Louveaux, F. : The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters* **13**, 1133–142 (1993)
- [84] Lazic, J., Hanafi, S., Mladenovic, N., Urozevic, D. : Variable neighbourhood decomposition search for 0-1 mixed integer programs. *Computers&Operations Research* **37**(6), 1055–1067 (2010)
- [85] Lee, J., Leung, J., Margot, F. : Min-up/min-down polytopes. *Discrete Optimization* **1**, 77–85 (2004)
- [86] Lemaréchal, C., Nemirovski, A., Nesterov, Y. : New variants of bundle methods. *Mathematical Programming* **69**(1-3), 111–147 (1995)
- [87] Lemaréchal, C., Sagastizábal, C., Pellegrino, F., Renaud, A. : Bundle methods applied to the unit-commitment problem. *System Modelling and Optimization IFIP — The International Federation for Information Processing* pp. 395–402 (1996)
- [88] Linderoth, J., Ralphs, T. : Noncommercial Software for Mixed-Integer Linear Programming : Theory and Practice. *CRC Press Operations Research Series* pp. 253–303 (2005)
- [89] Louat, C. : Etude et mise en oeuvre de stratégies de coupes efficaces pour des problèmes entiers mixtes 0-1. Ph.D. thesis (2009)
- [90] Lucas, J.Y., Sommer, T.L. : TANNAT : modélisation en Programmation Linéaire en Nombres Entiers du problème de la recherche de plannings d’arrêts de tranches nucléaires avec prise en compte des contraintes intra-sites et inter-sites. *Tech Report EDF R&D* (2008)
- [91] Lusby, R., Muller, L., Petersen, B. : A solution approach based on benders decomposition for the preventive maintenance scheduling problem of a stochastic large-scale energy system. *Journal of Scheduling* **16**(6), 605–628 (2013)
- [92] Mantawy, A., Abdel-Magid, Y., Selim, S. : A new genetic-based tabu search algorithm for unit commitment problem. *Electric Power Systems Research* **49**(2), 71–78 (1999)
- [93] Marsten, R., Hogan, W., Blankenship, J. : The boxstep method for large-scale optimization. *Operations Research* **23**(3), 389–405 (1975)
- [94] du Merle, O., Villeneuve, D., Desrosiers, J., Hansen, P. : Stabilized column generation. *Discrete Mathematics* **94**, 229–237 (1999)

- [95] Minoux, M. : Duality, Robustness, and 2-stage robust LP decision models. Application to Robust PERT Scheduling (2007)
- [96] Minoux, M. : Robust linear programming with right hand side uncertainty, duality and application. *Encyclopedia of Optimization* pp. 3317–3327 (2009)
- [97] Mladenović, N., Hansen, P. : Variable neighborhood search. *Computers & Operations Research* **24**(11), 1097–1100 (1997)
- [98] Mladenović, N., Hansen, P. : Variable neighborhood search : Principles and applications. *European Journal of Operational Research* **130**, 449–467 (2001)
- [99] Mladenović, N., Hansen, P., Urozević, D. : Variable neighborhood search and local branching. *Computers and Operations Research* **33**(10), 3034–3045 (2006)
- [100] Nemhauser, G., Savelsberg, M., Sigismondi, G. : MINTO, a Mixed INTEger Optimizer. *Operations Research Letters* **15**(1), 47–58 (1994)
- [101] Pacqueau, R., Soumis, F., Hoang, L.N. : A fast and accurate algorithm for stochastic integer programming, applied to stochastic shift scheduling
- [102] Pessoa, A., Sadykov, R., Uchoa, E., Vanderbeck, F. : In-out separation and column generation stabilization by dual price smoothing. *Lecture Notes in Computer Science* **7933**, 354–365 (2013)
- [103] Porcher, M., Gorge, A., Juan, O., Simovic, T., Dereu, G. : Challenge ROADEF/EURO 2010 : a large-scale energy management problem with varied constraints. EDF R&D Technical Report (2010)
- [104] Raidl, G.R., Felzl, H. : An improved hybrid genetic algorithm for the generalized assignment problem. *Proceedings of the 2003 ACM Symposium on Applied Computing* pp. 990–995 (2004)
- [105] Rajan, D., Takriti, S. : Min-Up/Down Polytopes of the Unit Commitment Problem with Start-Up Costs. Tech. rep., IBM Research Report (2005)
- [106] Remli, N. : Robustesse en programmation linéaire. Ph.D. thesis (2011)
- [107] Renaud, A. : Daily Generation Management at Electricité de France : From Planning Towards Real Time. *IEEE Trans. on Automatic Control* **38**(7) (1993)
- [108] Roussel-Ragot, P., Siarry, P., Dreyfus, G. : La méthode du "recuit simulé" en électronique : principe et parallélisation. colloque national sur la conception de circuits à la demande pp. 1–10 (1986)
- [109] Rozenknopf, A., Calvo, R.W., Alfandari, L., Chemla, D., Létocart, L. : Solving the electricity production planning problem by a column generation based heuristic. *Journal of Scheduling* (2012)
- [110] Sadykov, R. : Integer programming-based decomposition approaches for solving machine scheduling problems. Ph.D. thesis (2006)
- [111] Sagaspe, L. : Allocation sûre dans les systèmes aéronautiques : Modélisation, vérification et génération. Ph.D. thesis (2008)
- [112] Savelsbergh, M., Vanderbeck, F. : A generic view of Dantzig-Wolfe Decomposition in Mixed Integer Programming. *Operations Research Letters* **34**(3), 296–306 (2006)
- [113] Savelsbergh, M.W. : Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing* **6**(4), 445–454 (1994)
- [114] Schmid, V., Doerner, K.F., Hartl, R.F., Salazar-González, J.J. : Hybridization of very large neighborhood search for ready-mixed concrete delivery problems. *Computers & operations research* **37**(3), 559–574 (2010)
- [115] Soyster, A. : Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research* **21**, 1154–1157 (1973)
- [116] Taillard, E., Voss, S. : Popmusic- Partial Optimization Metaheuristic under Special Intensification Conditions. *Operations Research/Computer Science Interfaces Series* **15**, 613–629 (2002)
- [117] Takriti, S., Krasenbrink, B., Wu, L. : Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research* **48**, 268–280 (2000)
- [118] Talbi, E.G. : Hybrid metaheuristics. Springer (2013)

- [119] Talbi, E.G., Roux, O., Fonlupt, C., Robillard, D. : Parallel ant colonies for combinatorial optimization problems. *Lecture Notes in Computer Science* **1586**, 239–247 (1999)
- [120] Thiele, A., Terry, T., Epelman, M. : Robust linear optimization with recourse. Tech. rep. (2009)
- [121] Todosijević, R., Mladenović, M., Hanafi, S., Crévits, I. : VNS based heuristic for solving the Unit Commitment problem. *Electronic Notes in Discrete Mathematics* **39**, 153–160 (2012.)
- [122] Vanderbeck, F. : Branching in branch-and-price : a generic scheme. *Mathematical Programming Ser. A* **130**, 249–294 (2011)
- [123] Wolsey, L. : Solving Multi-Item Lot-Sizing problems with an MIP Solver using classification and reformulation. *Management Science* **48**(12) (2002)
- [124] Yıldız, S., Vielma, J.P. : Incremental and encoding formulations for mixed integer programming. *Operations Research Letters* **41**(6), 654–658 (2013)