HAL
open science

Gestion optimisée d'un modèle d'agrégation de flexibilités diffuses

Thomas Prelle

To cite this version:

Thomas Prelle. Gestion optimisée d'un modèle d'agrégation de flexibilités diffuses. Automatique. Ecole des Mines de Nantes, 2014. Français. NNT : 2014EMNA0177 . tel-01077960

HAL Id: tel-01077960

https://theses.hal.science/tel-01077960

Submitted on 27 Oct 2014

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 de Doctorat

Thomas PRELLE

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'École nationale supérieure des mines de Nantes
sous le label de l'Université de Nantes Angers Le Mans

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

Discipline : Informatique et application, section CNU 27
Spécialité : Recherche Opérationnelle
Unité de recherche : Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN)

Soutenue le 22 septembre 2014
Thèse n° : 2014 EMNA 0177

Gestion optimisée d'un modèle d'agrégation de flexibilités diffuses

JURY

Président :	**M. Bernard BOURGES**, Professeur de l'institut Mines-Télécom, École des mines de Nantes
Rapporteurs :	**M. Damien ENRST**, Professor, Université de Liège
	M. Michel GENDREAU, Professor, École Polytechnique de Montréal
Examinateurs :	**M. Kengy BARTY**, Ingénieur de recherche, EDF R&D OSIRIS
	M. Dominique FEILLET, Professeur de l'institut Mines-Télécom, École des mines de Saint-Etienne
Invités :	**M. Pierre-Etienne DELON**, Ingénieur de recherche, EDF R&D STEP
	M. Marc LE DU, Ingénieur de recherche, EDF R&D OSIRIS
Directrice de thèse :	**Mme Christelle GUÉRET**, Professeur des universités, Université d'Angers

Remerciements

À l'aboutissement de la rédaction de ce manuscrit de thèse, je suis intimement convaincu que la thèse est loin d'être le travail d'une seule personne. Je n'aurai en effet jamais pu achever ce doctorat sans le soutien d'un grand nombre de personnes.

Je tiens tout d'abord à remercier ma directrice de thèse, madame Christelle Guéret, pour ses conseils avisés, pour sa rigueur scientifique, pour sa curiosité pour un nouveau domaine d'application et aussi pour avoir toujours su se rendre disponible.

Je remercie également Pierre-Étienne Delon et Côme Bissuel du département STEP d'EDF R&D ainsi que Marc Le Du, Kengy Barty et Anes Dallagi du département OSIRIS d'EDF R&D pour avoir suivi l'évolution de mes travaux, pour leurs remarques et leurs conseils. Je tiens tout particulièrement à remercier Côme qui a fait en plus de son imputation.

Je remercie l'ensemble des membres de mon jury de thèse pour avoir accepté d'évaluer mon travail et particulièrement Michel Gendreau et Damien Enrst pour avoir accepté d'être rapporteurs de ma thèse ainsi que Bernard Bourges et Dominique Feillet pour avoir consenti d'être examinateurs de mon jury.

J'aimerai remercier l'ensemble des personnes (chercheurs, thésards, prestataires, stagiaires) du groupe P1B du département STEP d'EDF R&D pour m'avoir accueilli pendant 3 ans merci pour vos conseils, pour la bonne ambiance de travail, pour les pauses cafés, pauses goutée, etc, pour les afterworks programmés ou improvisés. Je tiens tout particulièrement à remercier Édouard et Romain avec lesquelles j'ai traversé ces trois années de thèse et avec qui j'ai pu partager les difficultés de réaliser une thèse. Un grand merci également à mes co-bureaux qui vont me manquer, tout particulièrement les discussions culinaires avec Jessica dans le RER de retour qui m'ont souvent ouvert l'appétit, les chansons de Manon dépaysantes (même si l'Allemagne n'est pas si loin de la France) et les "contestations" de Bruno qui m'ont fait sursauter plus d'une fois. J'espère trouver une ambiance de travail semblable pour la suite de ma carrière.

J'adresse mes remerciements à tous les membres de l'équipe SLP du laboratoire de l'IRCCyN et du département DAP de l'école des Mines de Nantes pour leur accueil lors de mes quelques venues à Nantes ainsi qu'à Delphine Turlier qui m'a permis de gérer administrativement ma soutenance de thèse.

Je tiens également à remercier mes amis pour les discussions par mails, les soirées, les week-ends, les ponts et les vacances qui m'ont permis de m'aérer l'esprit au cours de ces trois années.

Je termine en remerciant ma famille pour sa présence à ma soutenance et plus particulièrement ma mère pour ses relectures tout au long de la rédaction de cette thèse.

Table des matières

Remerciements **3**

Table des matières **5**

Table des figures **7**

Liste des tableaux **11**

Liste des algorithmes **13**

Glossaire **15**

Liste des acronymes **17**

Liste des symboles **19**

1 Introduction **23**

2 Contexte **27**
2.1 L'équilibre offre/demande : un enjeu essentiel 27
2.2 L'électricité de la production à la consommation 27
2.3 Les opérateurs du réseau électrique 30
2.4 Le marché de l'électricité aujourd'hui 30
2.5 Les évolutions du marché de l'électricité 31
2.6 Vers plus de flexibilité dans le réseau 33
2.7 Objectif général de cette thèse 46

3 Architecture et mode de gestion **49**
3.1 État de l'art 49
3.2 Architecture proposée 50
3.3 Mode de gestion proposé 52
3.4 Communication entre les acteurs de la centrale d'agrégation 56
3.5 Conclusion 58

4 Calcul du plan de production des procédés **61**
4.1 État de l'art 61
4.2 Modélisation du procédé 63
4.3 Description détaillée du problème 68
4.4 Instances 72
4.5 Complexité 73
4.6 Programmation linéaire en nombres entiers 86
4.7 Programmation par contraintes 95

4.8 Comparaison des différentes approches 110
4.9 Conclusion 112

5 Agrégation des plans de production 115
5.1 Description du problème 115
5.2 Caractérisation scientifique et complexité 124
5.3 Instances 124
5.4 Programmation dynamique 125
5.5 Programmation linéaire en nombres entiers 129
5.6 Méthode de Monte Carlo sur les chaines de Markov 130
5.7 Impact du signal de prix sur la solution 132
5.8 Conclusion 133

6 Conclusions et perspectives 137
6.1 Conclusions 137
6.2 Perspectives 139

Annexes 141

A Version désagrégée des contraintes des procédés 141

B Proposition d'un algorithme de modification d'un signal de prix linéaire 143

Bibliographie 145

Table des figures

1.1 Intégration des énergies renouvelables par couplage 25
1.2 Intégration des énergies renouvelables via des moyens de production flexible 25
1.3 Intégration des énergies renouvelables via des moyens de consommation flexible 25

2.1 Consommation moyenne hebdomadaire française de l'année 2012 28
2.2 Consommation moyenne journalière française entre le lundi 4 juin 2012 et le dimanche 1 juillet 2012 28
2.3 Consommation française du 20 février 2012 29
2.4 Consommation française du 27 juin 2012 29
2.5 Structure du réseau électrique d'aujourd'hui 32
2.6 Moyen d'action numéro 1 : augmenter de la production 32
2.7 Moyen d'action numéro 2 : décharger le stock 32
2.8 Répartition de la production électrique française pour la journée du 7 février 2012 34
2.9 Structure du réseau électrique de demain 35
2.10 Schéma de la centrale virtuelle d'agrégation de flexibilité 35
2.11 Exemple de courbe de charge d'un procédé de production pilotable 37
2.12 Exemple de courbe de flexibilité d'un procédé de production pilotable 37
2.13 Exemple de courbe de charge d'un procédé de consommation pilotable 37
2.14 Exemple de courbe de flexibilité d'un procédé de consommation pilotable 37
2.15 Exemple de courbe de charge d'un procédé de déplacement de la production fatale 39
2.16 Exemple de courbe de flexibilité d'un procédé de déplacement de la production fatale . . . 39
2.17 Fonctionnement du système PULSSI 39
2.18 Fonctionnement du système PAC 40
2.19 Exemple de courbe de charge d'un procédé de déplacement de la consommation fatale . . 40
2.20 Exemple de courbe de flexibilité d'un procédé de déplacement de la consommation fatale . 40
2.21 Exemple de courbe de charge d'un procédé de déplacement d'une consommation 41
2.22 Exemple de courbe de flexibilité d'un procédé de déplacement d'une consommation 41
2.23 Fonctionnement du système MICROSCOPE 41
2.24 Exemple de courbe de charge d'un procédé de stockage 42
2.25 Exemple de courbe de flexibilité d'un procédé de stockage 42
2.26 Moyen d'action numéro 3 : effacement de la consommation 44
2.27 Moyen d'action numéro 4 : reporter la consommation 44
2.28 Moyen d'action numéro 5 : lisser la consommation 44
2.29 Courbe de charge sans lissage 47
2.30 Courbe de charge avec lissage 47

3.1 Architecture d'un ensemble de procédés situés dans un secteur géographique précis 51
3.2 Périodes d'alerte 51
3.3 Phase 1 de PREMIO 51
3.4 Phase 2 de PREMIO 51
3.5 Fonctionnement du système ADDRESS 53

3.6 Architecture multi-niveaux . . . 53
3.7 Architecture d'un nœud . . . 53
3.8 Architecture du cas test . . . 54
3.9 Fonctionnement retenu . . . 55
3.10 Signal de prix induisant une augmentation de la production à un pas de temps donné . . . 57
3.11 Signal de prix induisant une augmentation de la consommation à un pas de temps donné . 57
3.12 Signal de prix induisant une augmentation de la production limitée à un pas de temps donné 57
3.13 Signal de prix induisant un lissage . . . 60
3.14 Signal de prix induisant un îlotage à un pas de temps donné . . . 60

4.1 Modélisation de Delon et Normand . . . 65
4.2 Modélisation de Heussen *et al* . . . 65
4.3 Modélisation générique . . . 65
4.4 Modélisation d'un procédé de production pilotable . . . 65
4.5 Modélisation d'un procédé de consommation pilotable . . . 65
4.6 Modélisation d'un procédé de déplacement de la production fatale . . . 66
4.7 Modélisation d'un procédé de déplacement de la consommation fatale . . . 66
4.8 Modélisation d'un procédé de déplacement d'une consommation . . . 66
4.9 Modélisation d'un procédé de stockage . . . 67
4.10 Exemple de plan de production pour les procédés sans stockage . . . 70
4.11 Graphe pour l'exemple de PSS_1 . . . 77
4.12 Automate $MaxOn^{p^+}$ fini déterministe pour la contrainte de durée maximale d'injection . . 101
4.13 Automate $MaxOn^{p^-}$ fini déterministe pour la contrainte de durée maximale de soutirage . 101
4.14 Automate $MaxOn^{x^+}$ fini déterministe pour la contrainte de durée maximale de charge . . 102
4.15 Automate $MaxOn^{x^-}$ fini déterministe pour la contrainte de durée maximale de décharge . 102
4.16 Automate $MinOn^{p^+}$ fini déterministe pour la contrainte de durée minimale d'injection . . 102
4.17 Automate $MinOff^{p^+}$ fini déterministe pour la contrainte de durée minimale entre un arrêt et le prochain départ d'une injection . . . 103
4.18 Automate $Inject$ fini déterministe concaténant les contraintes de durées minimale et maximale d'injection, et de durée minimale entre un arrêt et le prochain départ d'une injection . 104
4.19 Automate $Extract$ fini déterministe concaténant les contraintes de durées minimale et maximale de soutirage, et de durée minimale entre un arrêt et le prochain départ d'un soutirage 105
4.20 Automate $Charge$ fini déterministe concaténant les contraintes de durées minimale et maximale de charge, et de durée minimale entre un arrêt et le prochain départ d'une décharge . 106
4.21 Automate $Decharge$ fini déterministe concaténant les contraintes de durées minimale et maximale de décharge, et de durée minimale entre un arrêt et le prochain départ d'une décharge . . . 107
4.22 Automate $Call^{p^+}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre d'injections . . . 107
4.23 Automate $Call^{p^-}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre de soutirages . . . 108
4.24 Automate $Call^{x^+}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre de charges . . . 108
4.25 Automate $Call^{x^-}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre de décharges . . . 108
4.26 Automate $Cost$ fini déterministe avec un coût sur ses transitions pour la contrainte de coût de production . . . 108

5.1 Prix reçu par le coordinateur . . . 117
5.2 Plan de production du système piloté numéro 1 . . . 117
5.3 Plan de production du système piloté numéro 2 . . . 117

5.4 Plan de production du système piloté numéro 3 118
5.5 Plan de production du système piloté numéro 4 118
5.6 Plan de production du système piloté numéro 5 118
5.7 Plan de production du système piloté numéro 6 119
5.8 Puissances proposées par le coordinateur si aucun système piloté n'est sélectionné 119
5.9 Puissance proposée par le coordinateur si tous les systèmes pilotés sont sélectionnés 120
5.10 Combinaison optimale des puissances 120
5.11 Puissances proposées optimales pour le coordinateur 122
5.12 Combinaison optimale des variations de puissances 122
5.13 Variations de puissances optimales 123
5.14 Schéma illustrant la déformabilité des objets 124
5.15 Prix modifié pour le système piloté numéro 2 134
5.16 Plan de production modifié du système piloté numéro 2 134
5.17 Nouvelle combinaison optimale des variations de puissance 135
5.18 Nouvellles puissances proposées par le coordinateur 135

Liste des tableaux

4.1 Données pour la génération des instances de procédés sans stockage 73
4.2 Données pour la génération des instances de procédés avec stockage 73
4.3 Résultats du modèle de programmation linéaire en nombres entiers de base sur S_2 93
4.4 Résultats du modèle de programmation linéaire en nombres entiers de base sur S_{2a} (PSS) . 93
4.5 Résultats du modèle de programmation linéaire en nombres entiers de base sur S_{2b} (PAS) . 93
4.6 Comparaison des apports des améliorations sur S_3 96
4.7 Résultats sur S_2 96
4.8 Comparaison de solveurs sur S_3 96
4.9 Nombre d'instances résolues optimalement par le modèle de programmation par contraintes 111
4.10 Écart des solutions obtenues par le modèle de programmation par contraintes par rapport à l'optimum ou à la meilleure borne connue pour les instances S_2 111
4.11 Écart des solutions obtenues par le modèle de programmation par contraintes par rapport à l'optimum ou à la meilleure borne connue pour les instances S_{3a} 111
4.12 Écart des solutions obtenues par le modèle de programmation par contraintes par rapport à l'optimum ou à la meilleure borne connue pour les instances S_{3b} 111
4.13 Écarts moyens obtenus au bout d'une seconde par la programmation par contraintes et la programmation linéaire en nombres entiers sur les instances de S_3 113
4.14 Écarts moyens obtenus au bout d'une minute par la programmation par contraintes et la programmation linéaire en nombres entiers sur les instances de S_3 113
4.15 Écarts moyens obtenus au bout de 10 secondes par la programmation linéaire en nombres entiers en lui fournissant ou non la solution de la programmation par contraintes comme solution initiale sur les instances de S_3 113
4.16 Écarts moyens obtenus au bout d'une minute par la programmation linéaire en nombres entiers en lui fournissant ou non la solution de la programmation par contraintes comme solution initiale sur les instances de S_3 113
4.17 Avantages et inconvénients des méthodes pour le problème de calcul du plan de production des procédés pilotés directement 114

5.1 Résultats de la programmation linéaire en nombres entiers sur les 100 instances 130
5.2 Résultats de la méthode de Monte-Carlo sur les chaines de Markov sur les 100 instances . 134
5.3 Comparaison de la programmation linéaire en nombres entiers et la méthode de Monte-Carlo sur les chaines de Markov sur les 100 instances 134
5.4 Avantages et inconvénients des méthodes pour le problème d'agrégation des plans de production 136

Liste des algorithmes

1 Algorithme optimal pour les procédés de déplacement d'une consommation 74
2 Algorithme optimal pour les PSS_2 79

Glossaire

base Un moyen de production de la base constitue le socle de la production d'électricité. 14, 22, 23, 26, 38, 39

baseline La baseline d'un procédé est la puissance consommée/produite par défaut. 14, 28

blackout Un blackout est une panne généralisée du réseau électrique (coupure de courant électrique à grande échelle). 14, 15, 19, 38

centrale combinée gaz hybride Un cycle combiné gaz est composé d'une turbine à combustion (TAC) et d'une turbine à vapeur, chacune équipée de son propre alternateur. Dans une centrale combinée gaz hybride, la vapeur peut soit venir de la TAC soit d'une source renouvelable comme le solaire. 14, 15, 25, 41

centrale d'agrégation Une centrale virtuelle d'agrégation de flexibilité est une centrale qui agrège tous les procédés de modulation et qui réagit comme un moyen de production classique. 14, 16, 25, 35, 38, 41

contrainte dynamique Une contrainte dynamique est une contrainte physique d'un moyen de production. 14, 37

courbe de charge Série de valeurs horodatées de puissance, sur un pas de temps. 7, 14, 39, 122

délestage Un délestage est un effacement imposé à une partie de la consommation. 14, 23

effacement Un effacement est une diminution de la consommation. 7, 14, 19, 23, 25, 36, 42

effacement jour de pointe Le tarif effacement jour de pointe (EJP) d'EDF en France permet de payer l'électricité moins cher en contrepartie de tarif plus élevés pendant 22 jours dans l'année, décidés par EDF. 14, 19

energie primaire L'énergie primaire est l'énergie disponible dans l'environnement et directement exploitable sans transformation. Par exemple, le pétrole, le charbon ou le rayonnement solaire sont des énergies primaires alors que l'électricité n'en est pas une. 14, 23

fatal Un moyen de production est dit fatal quand le producteur ne peut agir sur la quantité d'électricité produite. Par exemple, une éolienne est un moyen de production fatal. 7, 14, 22, 23, 25, 26, 28, 30–32, 35, 37, 41, 107, 108, 130

heure creuse Un moment de la journée où l'électricité est moins chère. 14, 19, 30

heure pleine Un moment de la journée où l'électricité est plus chère. 14, 30

injection Un procédé est en injection quand il consomme moins ou produit plus que sa puissance par défaut (baseline). 14, 60–62, 78, 89, 90, 114, 115, 127

marché spot Le marché spot est le marché d'échange de l'électricité à court terme. 14, 64, 117

moyen de cogénération Un moyen de cogénération est un moyen de production qui produit deux énergies différentes dans le même processus, comme une centrale thermique produisant de l'électricité qui valorise la chaleur résiduelle en chauffant des habitations. 14, 15, 22, 41

opérateur d'effacement Un opérateur d'effacement est un acteur du marché de l'électricité qui agrège et valorise les effacements. 14, 23, 25

pas de temps Période de temps en heure, minute ou seconde. 14

pointe Un moyen de production de la pointe permettant de maintenir l'équilibre entre l'offre et la demande comme les centrales hydroélectriques. 14, 22, 23, 25, 26, 39

poste HTA/BT Un poste HTA/BT est un poste électrique (partie du réseau électrique qui permet de modifié la tension) situé à l'interface entre le réseau moyenne tension et le réseau basse tension. 14, 22, 24, 44, 46

poste source Un poste source est un poste électrique situé à l'interface entre le réseau de transport et le réseau de distribution. 14, 22, 24, 42, 44, 46

responsable d'équilibre Un responsable d'équilibre est un acteur du marché de l'électricité regroupant des fournisseurs, des producteurs et des opérateurs d'effacement qui permet de limiter les erreurs de l'ensemble des acteurs qu'il regroupe. 14, 23, 25, 35, 38, 42–44, 46, 130

réseau de distribution Le réseau de distribution permet de diffuser l'électricité au sein d'une région de consommation. 14, 22

réseau de transport Le réseau de transport permet d'acheminer l'électricité depuis les lieux de production vers les régions de consommation. 14, 19, 22

soutirage Un procédé est en soutirage quand il consomme plus ou produit moins que sa puissance par défaut (baseline). 14, 60–62, 114, 115, 127

tempo Il y a six prix du kWh différents dans le tarif Tempo d'EDF : il existe trois type de jours (Bleu, Blanc et Rouge). Pour chacun de ces types de jour, il existe un tarif heures pleines- heures creuses. 14, 19

valeur d'usage Une valeur d'usage désigne le coût de production de la quantité d'électricité. 14, 50

écarts Différence entre le total des quantités d'énergies injectées et le total des quantités d'énergie soutirées sur un pas de temps de 30 minutes. 14, 23

îlotage Une partie du réseau réalise un îlotage quand elle n'échange par d'électricité avec l'extérieur. Elle est auto-suffisante, elle réalise son propre équilibre offre/demande. 8, 14, 38, 48, 50, 52, 130–132

Liste des acronymes

CEA commissariat à l'énergie atomique et aux énergies renouvelables. 14, 15

CRE conseil supérieur de l'électricité. 14

DEL diodes électroluminescentes. 14, 28, 35, 37, 61, 129

DFA automate fini déterministe. 14, 87, 89, 90

DFAWC automate fini déterministe avec des coûts sur ses transitions. 14, 87, 90, 91

EJP effacement jour de pointe. 14, 19

ENR énergies renouvelables. 7, 14–17, 25, 26, 41, 54

EWG Energy Watch Group. 14, 15

PAS procédés avec stockage. 11, 14, 37, 53, 55, 62–65, 75, 77, 78, 84–86, 92, 101, 102

PDC procédés de déplacement d'une consommation. 13, 14, 37, 53, 55, 60, 64–66, 78

PF procédés fatals. 14, 37, 53, 55

PLNE programmation linéaire en nombres entiers. 11, 14, 53–55, 85, 102, 104–106, 124–126

PPC programmation par contraintes. 11, 14, 102–106

PSO particle swarm optimisation. 14, 53–55

PSS procédés sans stockage. 8, 11, 13, 14, 37, 53, 55, 61–65, 67, 69–71, 73, 77, 78, 84–86, 92, 102

STEP stations de transfert d'énergie par pompage. 14, 19, 22, 25, 41, 53, 54

TAC turbine à combustion. 14

Liste des symboles

$\delta_p(t)$ La différence au pas de temps t entre la puissance réellement consommée/produite $p_{gi}(t)$ et la puissance consommée/produite par défaut $P_{def}(t)$.

$\delta_{coor}(t)$ La variation de puissance du coordinateur au pas de temps t.

$\delta_{di}(t)$ La puissance échangée entre la partie processus internes du procédé et la partie interface réseau du procédé au pas de temps t.

$\delta_{gi}(t)$ La puissance échangée entre la partie interface réseau du procédé et le stockage au pas de temps t.

$\delta_{ip}(t)$ La puissance échangée entre la partie processus internes du procédé et le stockage au pas de temps t.

$\Delta_{max}(t)$ La puissance minimale modifiée du coordinateur au pas de temps t.

$\Delta_{min}(t)$ La puissance minimale modifiée du coordinateur au pas de temps t.

$\Delta_{sp}(t)$ La différence entre la puissance proposée par le système piloté sp et sa puissance par défaut au pas de temps t.

$\mathcal{T}$ L'horizon de temps.

Θ La tâche à satisfaire pour les procédés de déplacement d'une consommation sur la journée.

C La capacité de stockage du procédé.

C_{Loss} La constante de perte constante du stock.

C_{marg} Le coût marginal du procédé.

C_{on} Le coût de maintien du procédé.

C_{start} Le coût de départ du procédé.

$D_{iscr}(t)$ L'ensemble des puissances que le procédé peut atteindre au pas de temps t.

G_{max} Le gradient maximal entre deux puissances consécutives du procédé.

$I_{ndispo}(t)$ L'indisponibilité du procédé au pas de temps t.

$l(t)$ La quantité d'énergie perdue par le stockage du procédé au pas de temps t.

$L_{engthTask}$ La durée de la tâche Θ pour les procédés de déplacement d'une consommation.

$M^{p^+}_{axCall}$ Le nombre maximal d'injections dans le réseau pendant $\mathcal{T}$.

$M^{p^-}_{axCall}$ Le nombre maximal de soutirages depuis le réseau pendant $\mathcal{T}$.

$M_{axCall}^{x^+}$ Le nombre maximal de charges dans le réseau pendant $\mathcal{T}$.

$M_{axCall}^{x^-}$ Le nombre maximal de décharges depuis le réseau pendant $\mathcal{T}$.

$M_{axVol}^{p^+}$ La quantité maximale d'énergie injectée dans le réseau pendant $\mathcal{T}$.

$M_{axVol}^{p^-}$ La quantité maximale d'énergie soutirée depuis le réseau pendant $\mathcal{T}$.

$P_{coor}^{def}(t)$ La puissance par défaut du coordinateur au pas de temps t.

$P_{coor}^{pro}(t)$ La puissance proposée par le coordinateur à son client au pas de temps t.

$P_r(t)$ Le prix reçu par le procédé au pas de temps t.

$P_r^{coor}(t)$ Le prix envoyé par le client du coordinateur pour le pas de temps t.

$P_r^{sp}(t)$ Le prix envoyé par le coordinateur au système piloté sp au pas de temps t.

$P_{def}(t)$ La puissance par défaut du procédé au pas de temps t.

$P_{def}^{sp}(t)$ La puissance par défaut du système piloté sp au pas de temps t.

$p_{gi}(t)$ La puissance échangée entre le réseau et le procédé au pas de temps t.

$p_{ip}(t)$ La puissance produite ou consommée par la partie processus internes du procédé au pas de temps t.

P_{Loss} La constante de perte proportionnelle du stock.

$P_{max}(t)$ La variation de puissance maximale du procédé au pas de temps t.

P_{max}^{coor} La puissance maximale (valeur positive ou nulle) échangeable entre un coordinateur et son client.

P_{max}^{prod} La puissance maximale du procédé.

$P_{min}(t)$ La variation de puissance minimale du procédé au pas de temps t.

P_{min}^{conso} La puissance minimale du procédé.

P_{min}^{coor} La puissance minimale (valeur négative ou nulle) échangeable entre un coordinateur et son client.

$P_{pro}^{sp}(t)$ La puissance proposée par le système piloté sp au pas de temps t.

S_{power} Un tableau qui définit les puissances consommées pendant la durée de la tâche Θ pour les procédés de déplacement d'une consommation.

T L'horizon $\mathcal{T}$ est décomposé en T pas de temps.

$T_{maxOn}^{p^+}$ La durée maximale d'injection dans le réseau.

$T_{maxOn}^{p^-}$ La durée maximale de soutirage depuis le réseau.

$T_{maxOn}^{x^+}$ La durée maximale de charge dans le réseau.

$T_{maxOn}^{x^-}$ La durée maximale de décharge depuis le réseau.

$T_{MaxStart}$ La période de démarrage au plus tard de la tâche Θ pour les procédés de déplacement d'une consommation.

$T_{minOff}^{p^+}$ La durée minimale entre deux injections dans le réseau.

$T^{p^-}_{minOff}$ La durée minimale entre deux soutirages depuis le réseau.

$T^{x^+}_{minOff}$ La durée minimale entre deux charges dans le réseau.

$T^{x^-}_{minOff}$ La durée minimale entre deux décharges depuis le réseau.

$T^{p^+}_{minOn}$ La durée minimale d'injection dans le réseau.

$T^{p^-}_{minOn}$ La durée minimale de soutirage depuis le réseau.

$T^{x^+}_{minOn}$ La durée minimale de charge dans le réseau.

$T^{x^-}_{minOn}$ La durée minimale de décharge depuis le réseau.

$T_{MinStart}$ La période de démarrage au plus tôt de la tâche Θ pour les procédés de déplacement d'une consommation.

$x(t)$ Le niveau du stock au pas de temps t.

$X_{max}(t)$ Le niveau de stock maximal au pas de temps t.

$X_{min}(t)$ Le niveau de stock minimal au pas de temps t.

X^{unload}_{min} Le niveau de stock minimal avant de pouvoir déstocker.

1

Introduction

L'énergie fait partie des enjeux majeurs du XXIème siècle. Grâce à l'avènement de techniques de valorisation des pétroles non-conventionnels comme l'exploitation des sables bitumineux, l'extraction du pétrole de schiste ou la construction de puits offshore profonds, le pic pétrolier a pu être repoussé, mais son avènement reste inéluctable. Les acteurs de la production d'électricité ont donc actuellement pour objectifs d'agir en faveur de l'environnement et de la santé et de faire face à l'amenuisement des ressources disponibles en énergies fossiles. En effet d'après une étude du commissariat à l'énergie atomique et aux énergies renouvelables (CEA) [CEA, 2012], il restait en réserve en 2012 : 40 années de consommation de pétrole, 70 années de consommation de gaz naturel et 230 années de consommation de charbon. Si nous tenons le raisonnement suivant, à savoir que lorsqu'une source d'énergie fossile s'épuisera, elle sera progressivement remplacée par une autre jusqu'à épuisement de toutes les énergies fossiles, il ne resterait en fait qu'une centaine d'années de consommation de charbon d'après Energy Watch Group (EWG) [EWG, 2007]. Ils se doivent donc de développer un mix de production de plus en plus décarbonné.

Un producteur peut chercher pour cela à améliorer ses capacités à capter le carbone. Même si cette méthode est utile pour l'environnement, le dioxyde de carbone n'étant pas relâché dans l'atmosphère, elle n'est pourtant pas pérenne à cause de la diminution des réserves d'énergies fossiles.

Dans ce contexte social, économique et environnemental de plus en plus tendu, la solution qui semble la plus attractive afin de diminuer la part des énergies fossiles est d'augmenter la part des énergies renouvelables (ENR) (éolien, solaire, moyen de cogénération, etc.) dans le mix énergétique. Toutefois, la production de ces ENR est intermittente et non pilotable, pouvant de ce fait générer des situations de tension sur le réseau électrique entraînant un risque de blackout. Pour cette raison si ce type de production se développe, des solutions doivent être mises en place pour compenser la volatilité de ces énergies.

Une façon d'intégrer ces moyens de production dans le réseau est de les rendre moins intermittentes en les couplant avec un autre système stabilisant l'ensemble (cf. Figure 1.1). Un producteur peut ainsi associer ces ENR à un stockage comme une batterie [PEGASE, 2013]. Cependant cette solution n'est pas idéale car il est très difficile de stocker efficacement l'électricité. Il peut également les coupler à un autre moyen de production comme dans les centrales combinées gaz hybrides. Si une telle solution est utilisable par exemple pour des champs entiers de panneaux solaires thermodynamiques, elle n'est pas applicable pour toutes les sources ENR comme les panneaux photovoltaïques par exemple.

Une autre manière d'intégrer ces moyens de production dans le réseau est de trouver la façon de diminuer suffisamment la production pendant la durée des pics de production des ENR, et d'augmenter suffisamment la production pendant la durée de leurs creux de production. Un producteur peut utiliser ses centrales flexibles pour réaliser cette modulation (cf. Figure 1.2). Cependant, actuellement, le nombre de centrales flexibles décarbonées est limité pour des raisons d'économie du système.

Enfin un producteur peut intégrer ses énergies renouvelables aux réseaux en augmentant suffisamment la consommation pendant la durée des pics de production des ENR et en diminuant suffisamment la consommation pendant la durée de leurs creux de production (cf. Figure 1.3). Les fluctuations restantes seront compensées par les centrales flexibles des producteurs. Cependant ces sources de flexibilité (par exemple l'eau chaude sanitaire, le chauffage électrique, ...) sont par nature petites, hétéroclites et diffuses (i.e. réparties sur l'ensemble du territoire). Afin de disposer de suffisamment de puissance, il est nécessaire de les regrouper au sein d'une "centrale virtuelle d'agrégation de flexibilité". Celle-ci devra réagir comme une centrale classique, c'est-à-dire s'intégrer à la gestion d'ensemble du système électrique.

L'objectif général de cette thèse est de proposer un mode de gestion optimisé d'une centrale d'agrégation de flexibilités dans le but de minimiser le coût global du système électrique.

Cette thèse est constituée de quatre chapitres.

Le premier chapitre sera consacré à la présentation du contexte de cette thèse. Il introduira les enjeux du marché de l'électricité actuel puis en décrira en détail le concept de la centrale d'agrégation. Nous verrons dans ce chapitre que grâce à divers moyens de production, de stockage et de gestion de la consommation, un producteur peut disposer de nouveaux leviers d'action afin de gérer son équilibre offre/demande. Nous verrons également que pour que ces procédés aient un impact suffisant sur le réseau, ils doivent être regroupés au sein d'une centrale d'agrégation, qui devra réagir comme les autres moyens de production (nucléaire, thermique à flamme ou hydraulique) tout en prenant en compte les contraintes éventuelles sur le réseau.

Dans un second chapitre, nous définirons le système de pilotage retenu pour la centrale d'agrégation après avoir présenté un état de l'art des systèmes existants. Nous verrons que l'utilisation du mode de gestion que nous proposons impose de résoudre deux problèmes d'optimisation : le calcul des plans de production optimaux des procédés et l'agrégation de ces plans de production.

La résolution du premier problème sera traitée dans le troisième chapitre. Nous commencerons par détailler le problème puis nous réaliserons un état de l'art. Suite à cet état de l'art, nous exposerons un modèle générique de procédé. Comme nous disposons de peu d'instances réelles, nous générerons des instances aléatoirement basées sur des procédés réels. Ensuite, nous nous intéresserons à la complexité du problème général et de plusieurs de ses cas particuliers. Nous exposerons et testerons ensuite différentes méthodes de résolution.

Le second problème d'optimisation (agrégation des plans de production) sera traité dans le chapitre quatre. Ce chapitre débutera par une description détaillée du problème. Nous présenterons ensuite les différentes méthodes de résolution que nous avons développées puis nous les testerons sur des jeux de données issus de la résolution du premier problème d'optimisation.

Nous terminerons cette thèse par des conclusions et perspectives.

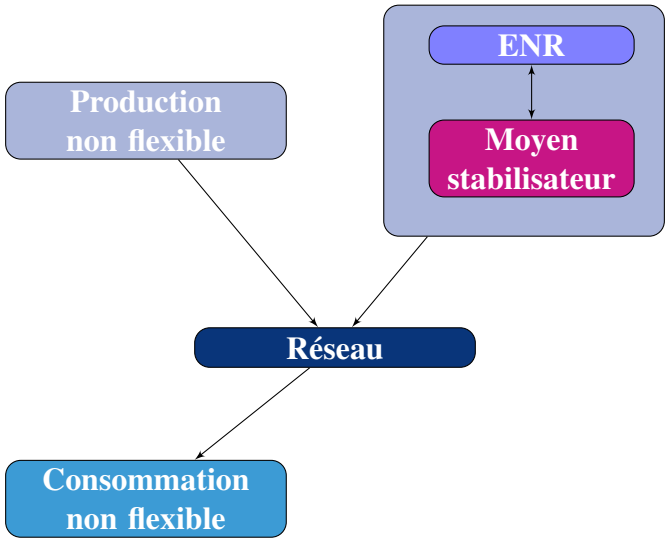

FIGURE 1.1 – Intégration des énergies renouvelables par couplage

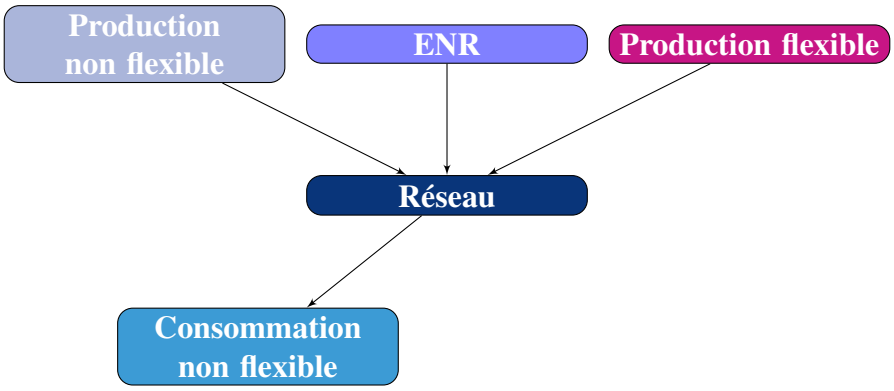

FIGURE 1.2 – Intégration des énergies renouvelables via des moyens de production flexible

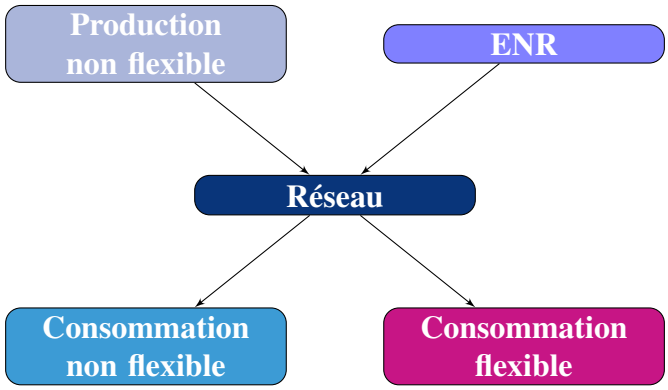

FIGURE 1.3 – Intégration des énergies renouvelables via des moyens de consommation flexible

2

Contexte

2.1 L'équilibre offre/demande : un enjeu essentiel

Un impératif du système électrique est de respecter l'équilibre offre/demande. En effet, l'électricité ne se stocke pas ou très peu et elle doit être disponible en temps réel, en toute saison et à tout moment de la journée pour les consommateurs. La production doit donc être égale à chaque seconde à la consommation. En effet, par exemple, si la consommation est supérieure à la production alors les moyens de production se déconnectent du réseau automatiquement pour éviter une casse. Ce découplage amplifie alors le phénomène et entraine une rupture de l'approvisionnement de l'électricité, il y a donc un blackout.

Il est donc essentiel de prévoir la consommation à chaque instant. En France, la consommation varie en fonction de deux paramètres : le comportement des consommateurs et l'époque de l'année. Elle est cependant plus au moins prévisible car cyclique. Nous pouvons identifier trois cycles : un cycle annuel (la figure 2.1 montre une pointe en février et un creux le 15 août [RTE, 2012]), un cycle hebdomadaire (la figure 2.2 montre que les jours de la semaine sont équivalents et que la consommation diminue le weekend [RTE, 2012]) et un cycle journalier (la figure 2.3 illustre qu'en hiver il y a une pointe le matin vers 9h et le soir vers 19h, et la figure 2.4 illustre qu'en été il y a une pointe vers 12h et une autre vers 19h [RTE, 2012]). Mais la consommation dépend aussi fortement de la température. En effet, les variations de température induisent une augmentation de la consommation du chauffage électrique en hiver et de la climatisation en été. Une erreur de précision d'un degré Celsius fait varier la puissance appelée de 1000 MW en hiver et de 700 MW en été (sachant qu'une tranche nucléaire a une puissance maximum de 1300 MW).

À court terme, l'équilibre offre/demande est réalisé par des systèmes de réglages automatiques installés directement sur les moyens de production. Afin de rétablir l'équilibre offre/demande, actuellement, ce sont les moyens de production démarrés et qui ont suffisamment de réserve qui s'adaptent généralement à la consommation, à trois exceptions près : l'asservissement des usages électriques comme le démarrage automatique des ballons d'eau chaude sanitaire pendant les heures creuses, l'effacement volontaire avec soit les tarifs EJP et tempo ou le pilotage direct, et le stockage de l'électricité via des stations de transfert d'énergie par pompage (STEP) dont le fonctionnement consiste à remonter de l'eau dans un barrage quand le prix de l'électricité est bas et à la turbiner quand ce prix est élevé.

2.2 L'électricité de la production à la consommation

La figure 2.5 illustre le cheminement de l'électricité. Une fois produite par un moyen de production, l'électricité emprunte un réseau de lignes comparable au réseau routier [RTE, 2014b]. Le réseau de transport

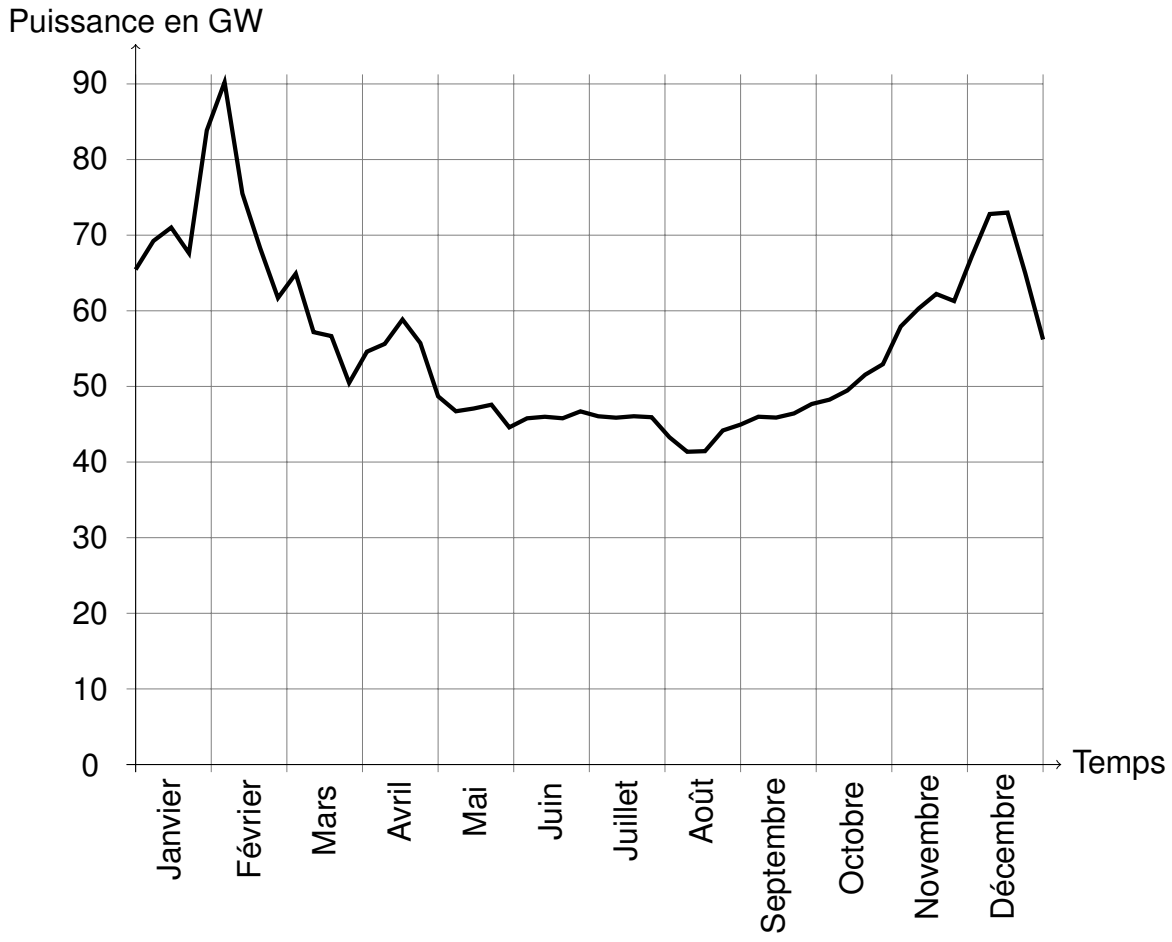

FIGURE 2.1 – Consommation moyenne hebdomadaire française de l'année 2012

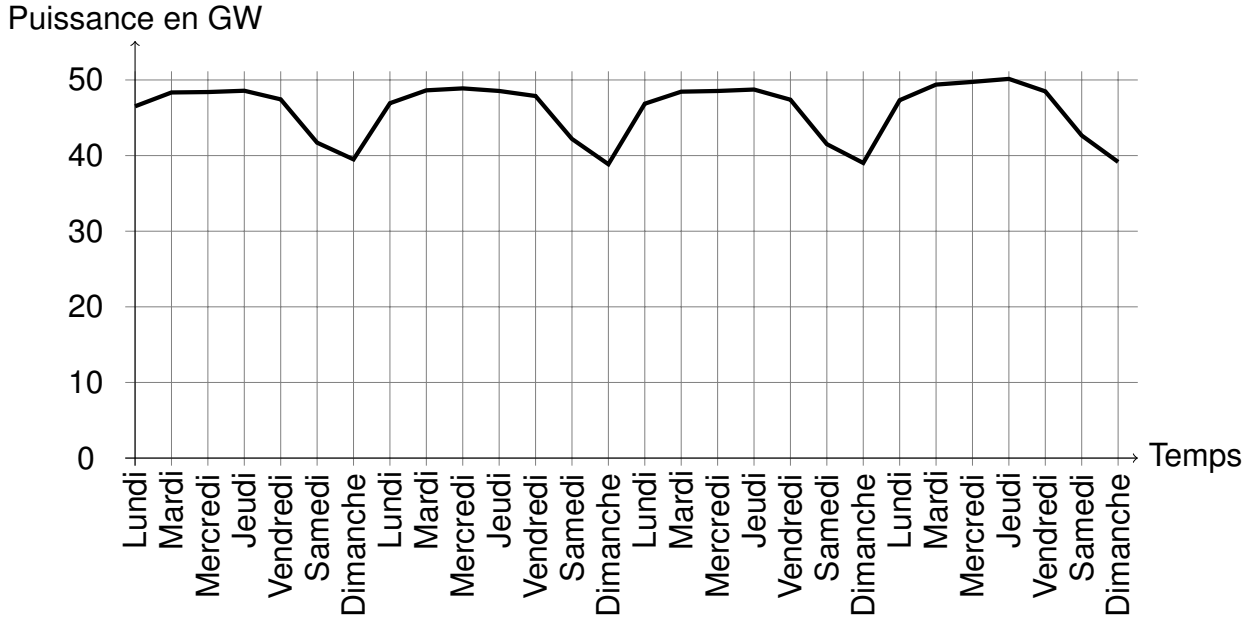

FIGURE 2.2 – Consommation moyenne journalière française entre le lundi 4 juin 2012 et le dimanche 1 juillet 2012

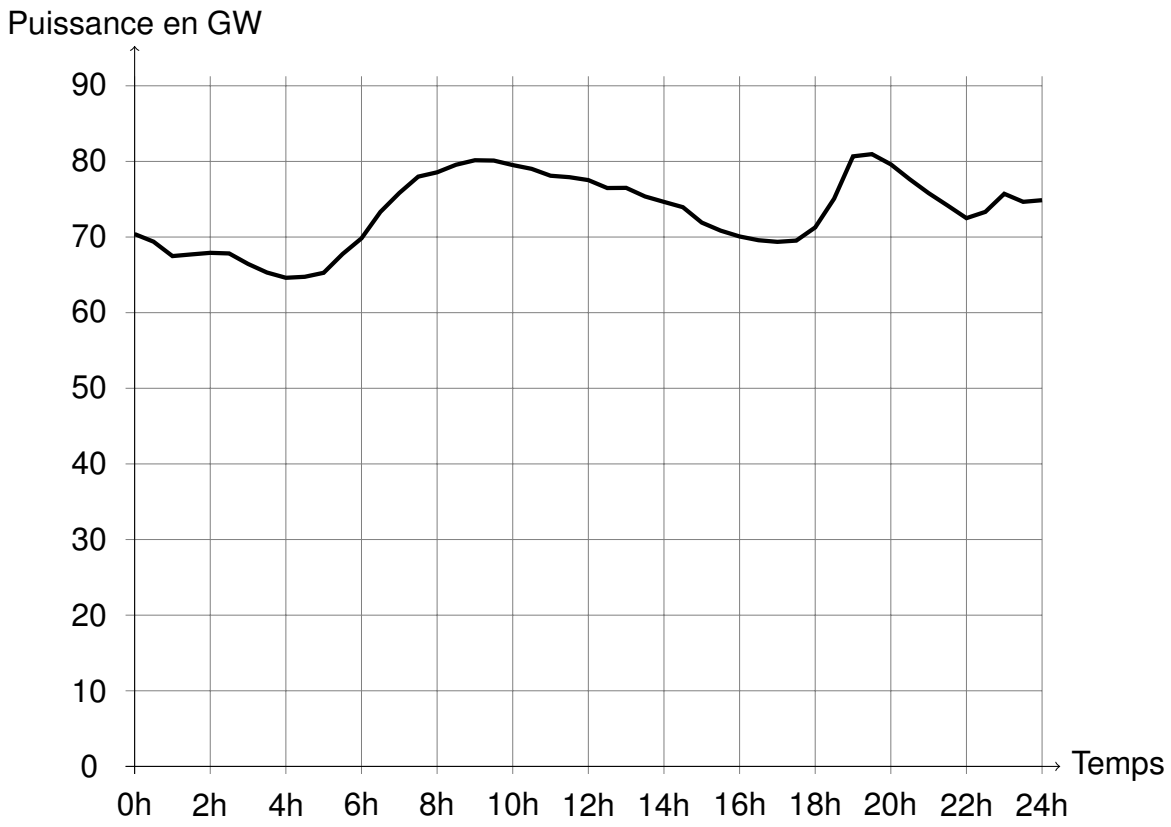

FIGURE 2.3 – Consommation française du 20 février 2012

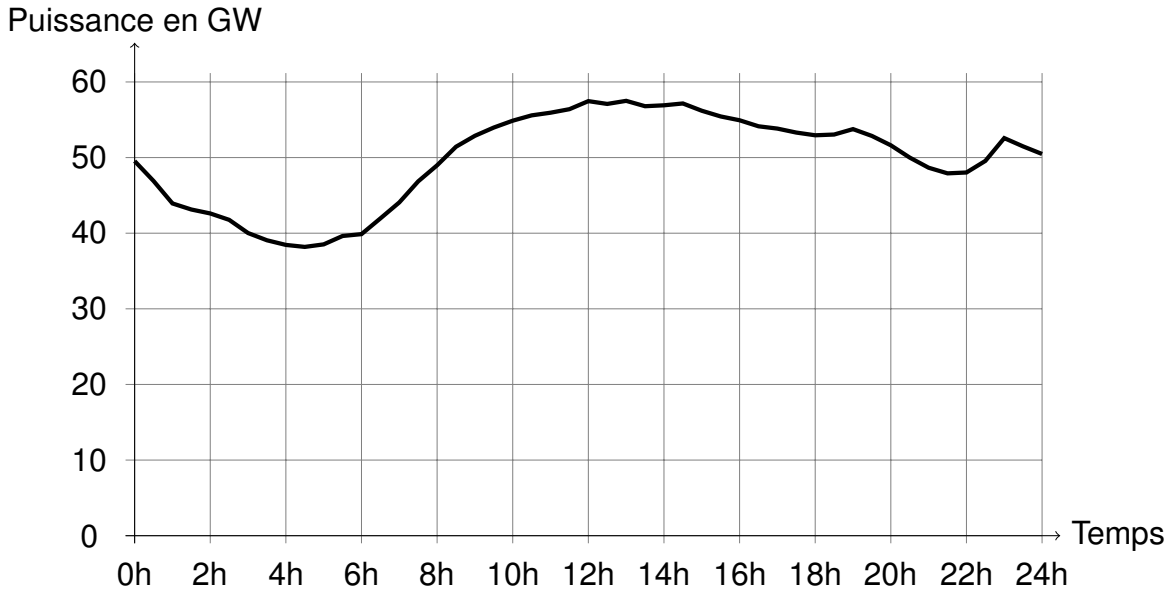

FIGURE 2.4 – Consommation française du 27 juin 2012

composé des grands axes, permet d'acheminer l'électricité depuis les lieux de production vers les régions de consommation. Le réseau de distribution composé des axes secondaires, permet d'acheminer l'électricité qui vient du réseau de transport vers les consommateurs. À l'intersection des voies, nous trouvons des postes électriques, l'équivalent des échangeurs sur les autoroutes. Leurs fonctions principales sont de répartir l'énergie entre les différentes lignes issues du poste, de raccorder un producteur ou un consommateur au réseau électrique et de modifier la tension de l'électricité acheminée. En effet en France, le courant produit est porté à 400 000 Volts ce qui permet de le transporter sur de longues distances avec le minimum de pertes d'électricité vers des régions consommatrices de grandes quantités d'énergie. Ces lignes sont comparables aux autoroutes. La tension de l'électricité est ensuite abaissée au niveau d'un poste afin de la transporter au niveau d'une région. Cette partie du réseau permet d'acheminer l'électricité jusqu'aux grands consommateurs industriels et aux réseaux de distribution. Elles sont comparables aux routes nationales. L'électricité passe du réseau de transport aux réseaux de distribution via un poste source. Elle est ensuite acheminée depuis un poste source vers un poste HTA/BT situé dans un quartier. Puis depuis ce poste, elle est distribuée jusqu'au compteur du client.

Pour résumer, l'électricité ne se stocke pas et elle doit être acheminée physiquement du lieu de production vers le lieu de consommation. Les deux enjeux essentiels sont donc : l'équilibre offre/demande à chaque seconde et le respect des contraintes réseaux : les contraintes d'intensité (la capacité maximale des câbles et des transformateurs) et les contraintes de tension dans les lignes électriques.

2.3 Les opérateurs du réseau électrique

Trois types d'opérateurs interagissent avec ce réseau afin de maintenir l'équilibre offre/demande : les fournisseurs d'électricité, les producteurs d'électricité et les gestionnaires du réseau. Les fournisseurs d'électricité comme EDF, GDF-Suez ou Direct Énergie commercialisent l'électricité en alimentant les consommateurs finaux en électricité soit à partir de l'électricité qu'ils ont produite eux-même, soit à partir d'une énergie qu'ils ont achetée en France ou à l'étranger.

Les producteurs d'électricité (EDF, GDF-Suez, la compagnie nationale du Rhône, ...) disposent d'un certain nombre de moyens de production qui peuvent être regroupés en trois classes : la base, la pointe et le fatal. La base regroupe des centrales thermiques, nucléaires et à flamme. La pointe est composée des centrales hydrauliques et des turbines à combustion. Le fatal est, quant à lui, composé entre autres de toutes les énergies renouvelables comme l'énergie solaire, éolienne, hydraulique fatal ou les moyens de cogénération. Leur production dépendant des conditions météorologiques, ces énergies ne sont pas toujours disponibles. Il est à noter que selon sa taille, un moyen de production peut être relié soit directement au réseau de transport (par exemple une centrale nucléaire), soit à un réseau de distribution (par exemple les panneaux photovoltaïques chez un particulier). En plus de ces moyens de production, il existe quelques procédés permettant de stocker de grandes quantités d'électricité comme les STEP.

Les gestionnaires du réseau doivent assurer l'acheminement de l'électricité des producteurs vers les consommateurs. Les gestionnaires doivent également réagir en temps réel en cas d'erreur de prévision de consommation ou de production.

Lorsqu'un déséquilibre se produit, des systèmes de réglage automatiques installés directement sur les moyens de production se déclenchent et permettent de maintenir l'équilibre offre/demande pendant 15 minutes. Pendant ce temps, les gestionnaires de réseaux font appel aux producteurs et aux consommateurs connectés au réseau afin qu'ils modifient très rapidement leur programme de fonctionnement prévu.

2.4 Le marché de l'électricité aujourd'hui

Depuis la libéralisation du marché de l'électricité, la France doit réussir à faire cohabiter plusieurs acteurs dans le système électrique. Afin que l'équilibre offre/demande puisse être satisfait à chaque seconde, le

rôle de responsable d'équilibre a été créé. Tous les acteurs, qu'ils soient fournisseurs, producteurs, consommateurs ou tradeurs doivent être rattachés à un responsable d'équilibre. Ce dernier doit respecter l'équilibre offre/demande dans son périmètre d'équilibre. C'est à dire que la somme de sa production et de ses achats doit être égale à tout instant à la somme de sa consommation et de ses ventes tout en respectant les contraintes de ses acteurs et de façon à maximiser son gain global. Il est à noter que le responsable d'équilibre ne doit pas prendre en compte les contraintes réseaux pendant le calcul de son équilibre. Cependant, en France, il envoie son plan d'injection (quantité d'électricité injectée par le responsable d'équilibre en chaque poste, à chaque instant) aux gestionnaires de réseau de transport.

Pendant la journée, le responsable d'équilibre a pour rôle de coordonner l'offre et la demande des opérateurs qu'il gère en compensant les erreurs des uns grâce aux autres. Actuellement, en cas de pic de consommation, le responsable d'équilibre peut :

- augmenter la production :
  - en agissant sur ses moyens de production de la pointe (cf. Figure 2.6), ou en achetant sur les marchés ;
  - en déchargeant les procédés de stockage qui ont été chargés auparavant (cf. Figure 2.7) ;
- diminuer la consommation grâce à l'asservissement des usages électriques, l'effacement volontaire et le pilotage direct.

Inversement, pendant les pics de production fatale, le responsable d'équilibre peut :

- diminuer la production pilotable ;
- stocker grâce aux procédés de stockage ;
- augmenter la consommation grâce à l'asservissement des usages électriques.

Si le responsable d'équilibre ne peut pas compenser les écarts, le gestionnaire de réseau utilise un mécanisme d'ajustement afin de rétablir l'équilibre offre/demande. Il fait appel alors aux responsables d'équilibre connectés au réseau pour qu'ils modifient très rapidement leur programme de fonctionnement prévu. Cet ajustement est payé par le responsable d'équilibre qui est à l'origine de l'écarts. Dans les cas extrêmes, le gestionnaire de réseau réalise un délestage (effacement imposé à une partie de la consommation) afin de sécuriser son réseau.

2.5 Les évolutions du marché de l'électricité

En 2013, la part de la base dans la production d'électricité en France représentait 80,4 %, celle de la pointe 14,8 % et celle du fatal 4,8 % [RTE, 2014a] (cf. Figure 2.8 [RTE, 2012]). La volonté d'augmenter la part des énergies renouvelables dans le mix énergétique français implique une augmentation significative de la part des moyens de production fatals. Les responsables d'équilibre devront donc les intégrer dans leur périmètre d'équilibre.

De plus l'électrification de nouveaux usages comme la voiture électrique et l'évolution des usages de l'électricité à cause par exemple des changements de réglementation pour la construction des nouveaux bâtiments vont faire évoluer la courbe de consommation. Par exemple, les ballons d'eau chaude sanitaire classiques (qui chauffent l'eau grâce à une résistance) coûteux en energie primaire vont disparaitre au profit de ballons d'eau chaude thermodynamique (qui chauffent l'eau grâce à une pompe à chaleur) plus économes en energie primaire. La démocratisation des voitures électriques pourrait aussi intensifier le pic de consommation du soir.

Et enfin, l'article 14 de la loi Brottes [JORF, 2013] a introduit tout récemment un cadre juridique autorisant l'implication d'un nouvel acteur sur le marché de l'électricité : l'opérateur d'effacement. Celui-ci

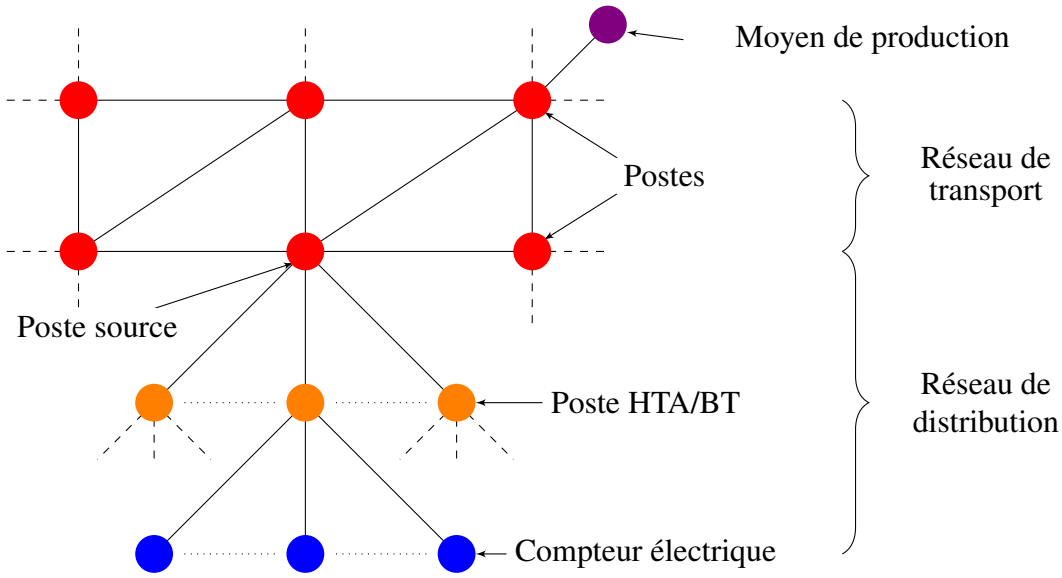

FIGURE 2.5 – Structure du réseau électrique d'aujourd'hui

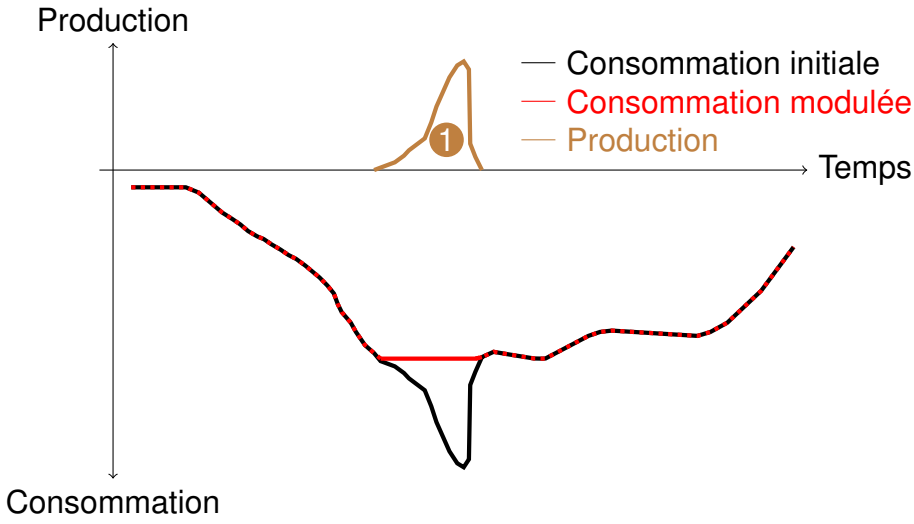

FIGURE 2.6 – Moyen d'action numéro 1 : augmenter de la production

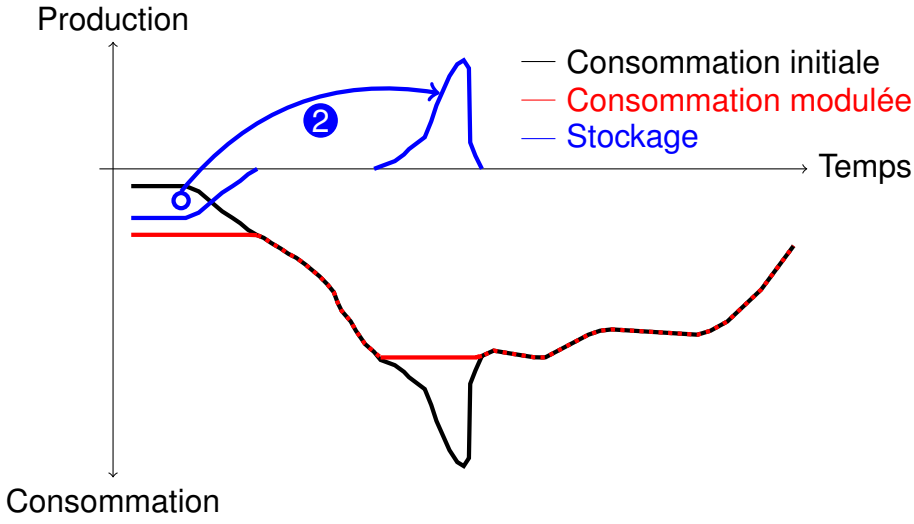

FIGURE 2.7 – Moyen d'action numéro 2 : décharger le stock

contractualise avec des consommateurs pour qu'ils acceptent de moduler leur consommation. Il agrège alors ces effacements pour les revendre à un autre acteur. L'opérateur d'effacement peut être différent du fournisseur qui approvisionne le consommateur, cependant il intéragit de la même façon que les autres acteurs avec les gestionnaires de réseau. Il n'est pas obligatoirement relié à un responsable d'équilibre mais il doit prévenir les responsables d'équilibre des fournisseurs impactés. Un responsable d'équilibre peut donc regrouper des fournisseurs, des producteurs, des consommateurs, des tradeurs et des opérateurs d'effacement.

2.6 Vers plus de flexibilité dans le réseau

Du fait du caractère intermittent des ENR, leur insertion dans le réseau va rendre l'équilibre offre/demande encore plus difficile à maintenir pour les responsables d'équilibre et pour les gestionnaires de réseau. Une solution, pour intégrer ces nouvelles variables dans l'équation de l'équilibre offre/demande serait d'augmenter les capacités du réseau afin de pouvoir acheminer l'électricité à tout instant, mais aussi de pouvoir diminuer la production adaptable pendant la durée de leurs pics de production, et l'augmenter suffisamment pendant la durée de leurs creux de production. Les gestionnaires de réseau devront investir dans leur réseau. Les responsables d'équilibre devront avoir recours aux moyens de production de la pointe afin d'adapter la production à la consommation. Cependant, le nombre de centrales flexibles économiques est limité pour des raisons d'économie du système.

Une autre voie afin d'intégrer les nouveaux moyens de production fatals est d'être capable de rendre ces procédés non fatals en les couplant à un moyen de stockage comme une STEP. Cependant, les moyens de stockage d'électricité industrielle sont limités. Une seconde façon de les rendre non fatals est de les coupler à un autre moyen de production comme dans les centrales combinées gaz hybrides. Une telle solution est utilisable par exemple pour des champs entiers de panneaux solaires, mais elle n'est pas rentable dans le cas de petits dispositifs tels que des panneaux solaires situés sur une maison.

Un responsable d'équilibre peut aussi intégrer ses moyens de production fatals en augmentant suffisamment la consommation pendant la durée de leurs pics de production, et en diminuant suffisamment la consommation pendant la durée de leurs creux de production. Ainsi, au lieu d'adapter la production à la consommation, le responsable d'équilibre réalise son équilibre offre/demande en modifiant la production et la consommation. C'est cette dernière piste qui va nous intéresser.

Afin de mettre cette solution en œuvre, le responsable d'équilibre peut développer le stockage et utiliser des procédés de modulation de la production et de la consommation. Ces sources de flexibilité sont hétéroclites : nous pouvons citer par exemple l'eau chaude sanitaire, les radiateurs, les véhicules électriques, le solaire thermique, l'électroménager. Les procédés de modulation sont eux aussi hétéroclites : par exemple un système regroupant une pompe à chaleur, des radiateurs auxquels nous ajoutons un ballon d'eau chaude ou un système qui permet de baisser la luminosité des lampadaires dans une rue. Les procédés sont aussi par nature petits (par rapport aux autres moyens de production) et répartis sur l'ensemble du territoire (procédés dits diffus). Il est à noter que de part leur taille, ces procédés seront reliés à un réseau de distribution (cf. Figure 2.9).

Afin d'intégrer facilement ces procédés au processus de répartition du responsable d'équilibre, l'idée est de les regrouper au sein d'un réseau électrique "intelligent" (smart grid) qui devra pouvoir gérer tous ces procédés et réagir comme un moyen de production classique. Ce smart grid sera appelé : "centrale virtuelle d'agrégation de flexibilité" (cf. Figure 2.10). Comme les autres moyens de production, cette centrale d'agrégation devra transmettre à son responsable d'équilibre les données qui caractérisent sa production. Cette centrale d'agrégation regroupant des procédés hétéroclites devra pouvoir gérer toutes les consommations (modulables et non-modulables) et toutes les productions que ce soit des moyens de production fatals (comme les ENR) ou pilotables (comme les turbines à combustion). Elle devra aussi respecter les contraintes liées aux réseaux de distribution (la capacité maximale des câbles et la puissance minimale et maximale des postes électriques).

Un grand nombre de procédés intégrables dans ces centrales d'agrégation existent à l'heure actuelle.

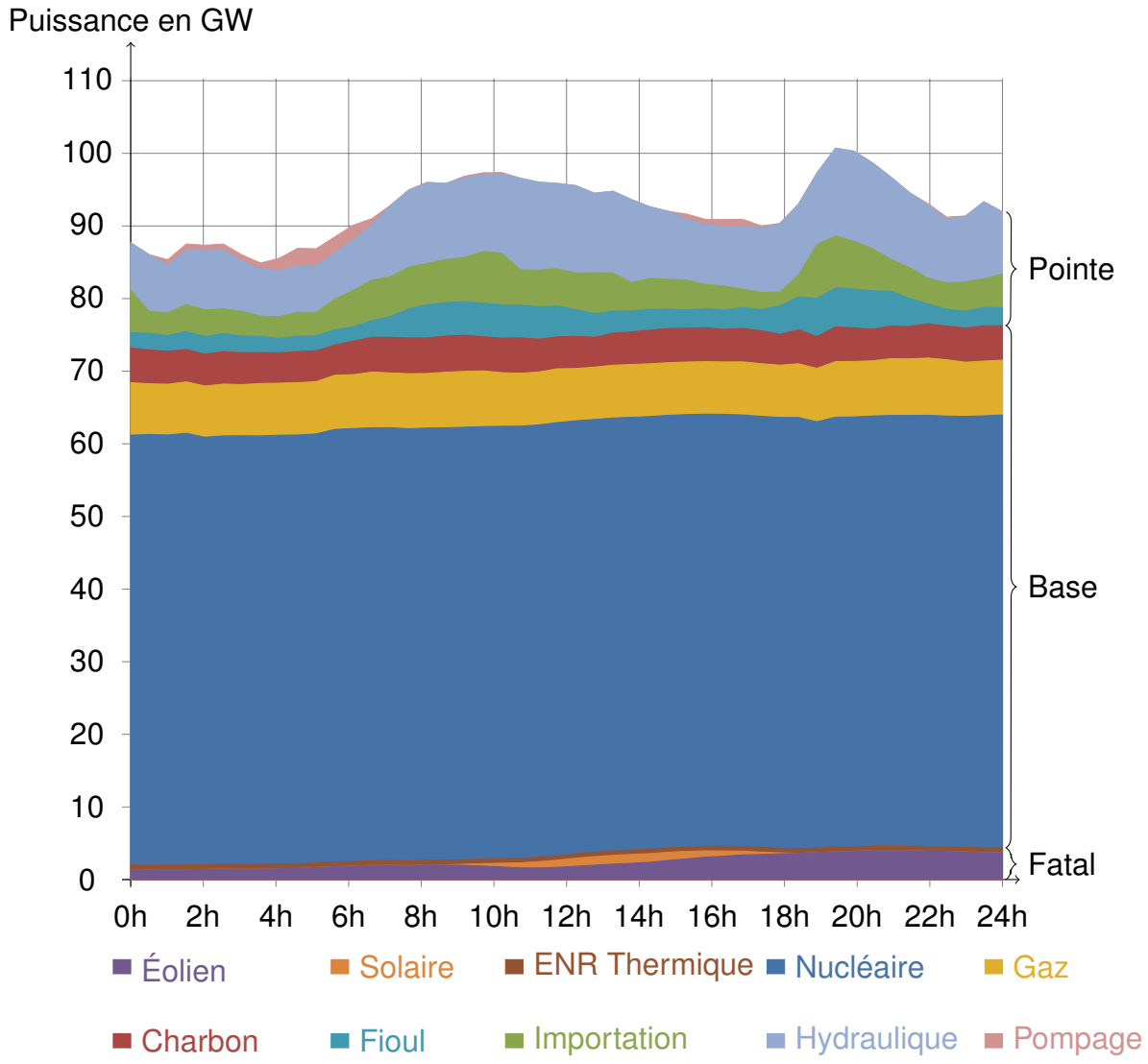

FIGURE 2.8 – Répartition de la production électrique française pour la journée du 7 février 2012

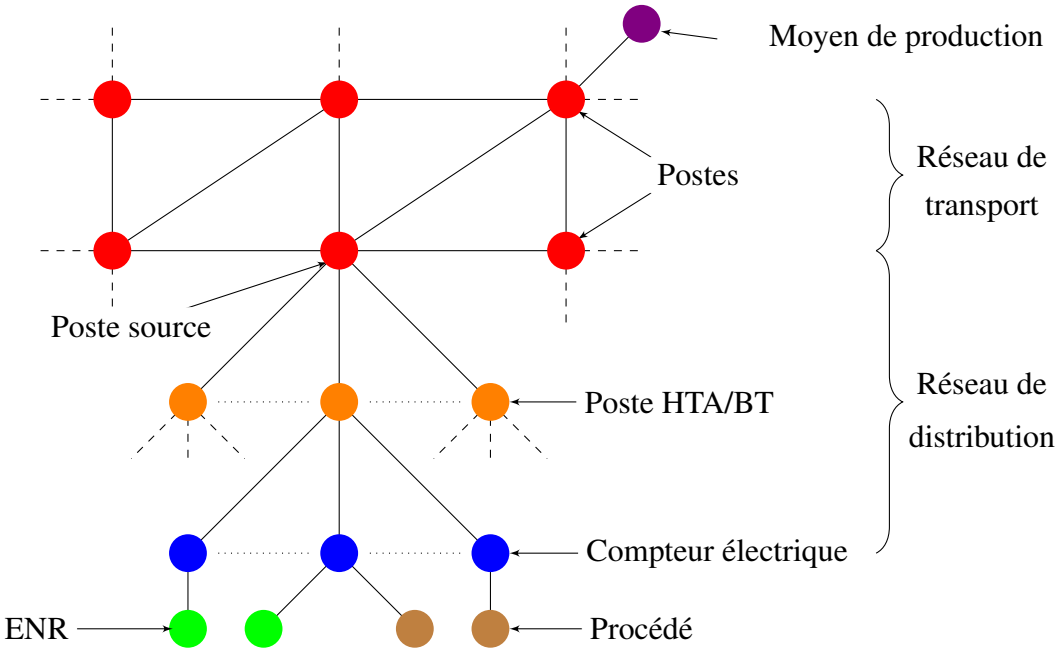

FIGURE 2.9 – Structure du réseau électrique de demain

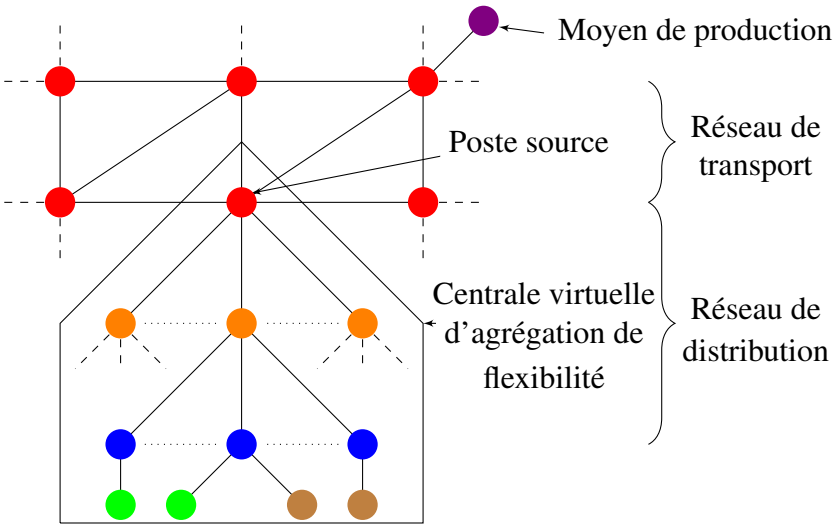

FIGURE 2.10 – Schéma de la centrale virtuelle d'agrégation de flexibilité

A contrario d'une centrale classique, tous ces procédés ont une puissance consommée/produite par défaut ("baseline") non nulle que nous supposerons connue. Par exemple, un radiateur en hiver a une consommation de base pour maintenir une pièce à sa température de confort. Les flexibilités apportées par ces procédés sont en réalité une modulation de leur puissance consommée/produite par rapport à leur baseline. Ces différents procédés comportent des caractéristiques très variées. Nous proposons de les regrouper en plusieurs catégories :

- **Les procédés de consommation fatale :** il s'agit de procédés de consommation qui ne peuvent être contrôlés comme une chaine hifi (nous ne savons pas quand elle sera allumée ou éteinte). Pour ce type de procédés, la puissance par défaut est la prévision de consommation du procédé (supposée connue).

- **Les procédés de production fatale :** il s'agit des procédés de production qui ne peuvent être contrôlés comme les panneaux photovoltaïques ou les éoliennes. Pour ce type de procédés, la puissance par défaut est la prévision de production du procédé (supposée connue).

- **Les procédés de production pilotable :** par opposition aux procédés de production fatale, la production de ces procédés peut être contrôlée. Un tel procédé est par exemple une turbine à combustion qui pourra être démarrée ou arrêtée suivant la demande. Pour ce type de procédés, la puissance par défaut est la production "normale". La figure 2.11 représente l'exemple d'une courbe de charge d'un procédé de production pilotable avec en bleu la courbe de la puissance par défaut et en rouge la courbe de la puissance modulée. Nous voyons que la modulation peut être faite à la hausse comme à la baisse. La flexibilité (différence entre la puissance modulée et la puissance par défaut) correspond à la hauteur des rectangles hachurés : cette valeur est positive si la puissance proposée (trait rouge) est supérieure à la puissance par défaut (trait bleu), et négative sinon. La figure 2.12 montre la courbe de flexibilité associée à la figure 2.11.

- **Les procédés de consommation pilotable :** par opposition aux procédés de consommation fatale, la consommation de ces procédés peut être contrôlée. La modulation de consommation d'un tel procédé est dite pure. Cela signifie que si une baisse de la consommation est réalisée à un instant t, elle ne devra pas être compensée par la suite. Un tel procédé est par exemple un système de pilotage de l'éclairage public utilisant des diodes électroluminescentes (DEL) à la place des ampoules : il est possible de faire varier le niveau d'éclairage des lampadaires à distance. Pour ce type de procédés, la puissance par défaut est la consommation "normale". Pour un système de gestion de l'éclairage public qui contrôle $400\ W$ de DELs, la puissance par défaut est généralement de $400\ W$ de 20h à 7h. La figure 2.13 représente un exemple de courbe de charge pour un procédé de consommation pilotable avec en bleu la courbe de la puissance par défaut et en rouge la courbe de la puissance modulée. Nous voyons que la modulation peut être faite à la hausse comme à la baisse et qu'elle est bien pure. La figure 2.14 montre la courbe de flexibilité associée à la figure 2.13. Il est à noter que la courbe de flexibilité (cf. Figure 2.14) du procédé de consommation pilotable est similaire à celle (cf. Figure 2.12) du procédé de production pilotable.

- **Les procédés de déplacement de la production fatale :** ce type de procédés est à mi-chemin entre les procédés de production fatale et les procédés de production pilotable. La production n'est pas pilotable mais elle peut être stockée localement afin d'être utilisée ultérieurement. Pour ce type de procédés, la puissance par défaut est l'apport reçu par le procédé. Un tel procédé est par exemple un système de production solaire thermique qui peut stocker le liquide chauffé dans un ballon et qui peut ultérieurement produire de l'électricité via une turbine à vapeur à partir de cette eau. Son apport par défaut est alors l'apport solaire. La figure 2.15 représente l'exemple d'une courbe de charge d'un procédé de déplacement de la production fatale avec en bleu la courbe de la puissance par défaut et en rouge la courbe de la puissance modulée. Nous voyons sur cette figure que la charge est lissée. La figure 2.16 montre la courbe de flexibilité associée à la figure 2.15.

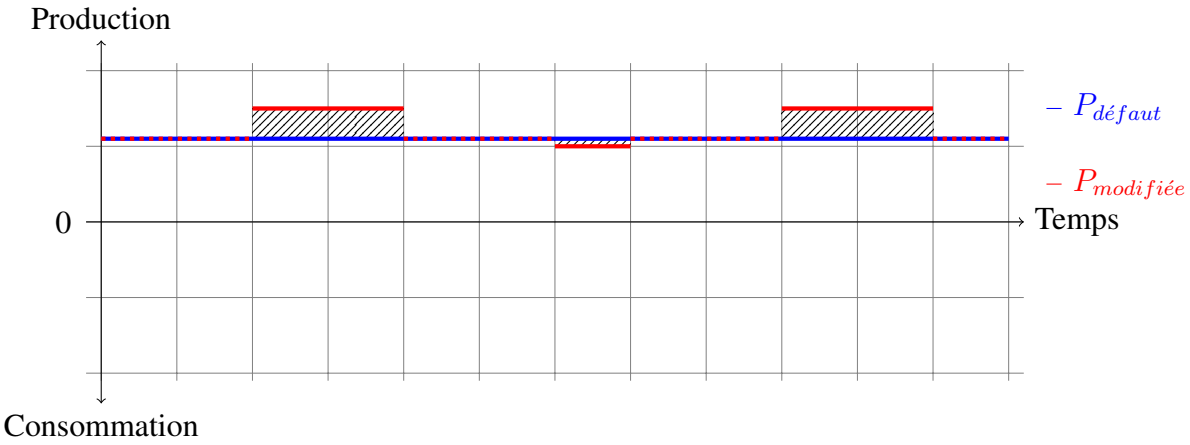

FIGURE 2.11 – Exemple de courbe de charge d'un procédé de production pilotable

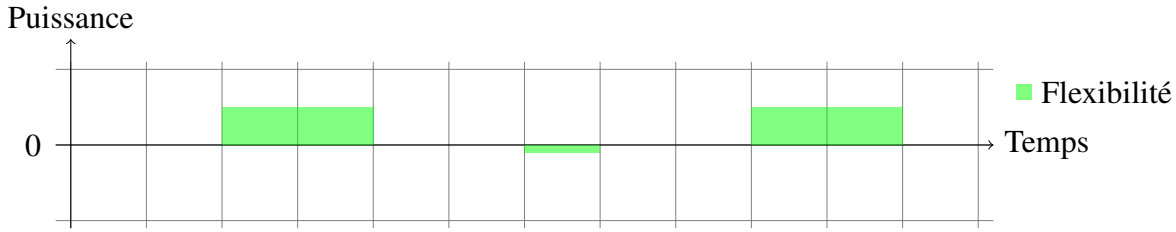

FIGURE 2.12 – Exemple de courbe de flexibilité d'un procédé de production pilotable

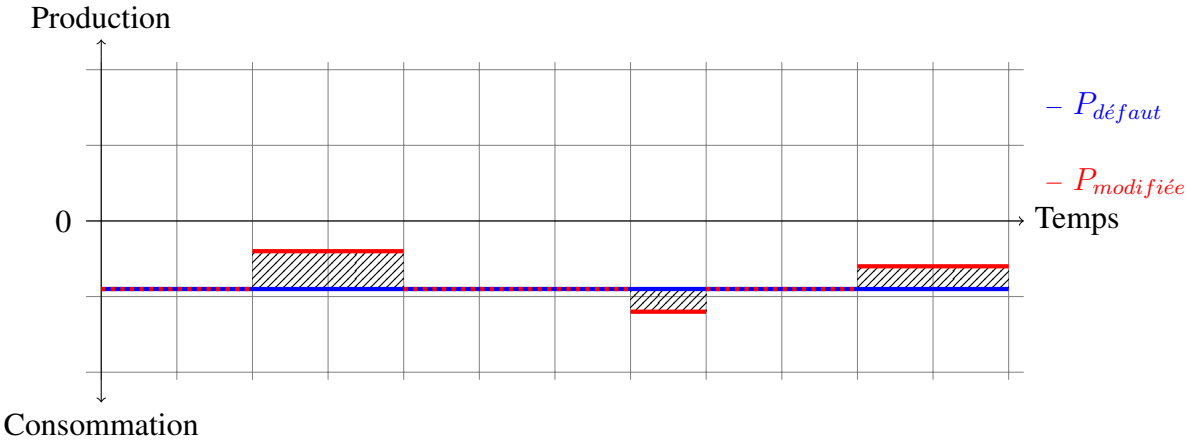

FIGURE 2.13 – Exemple de courbe de charge d'un procédé de consommation pilotable

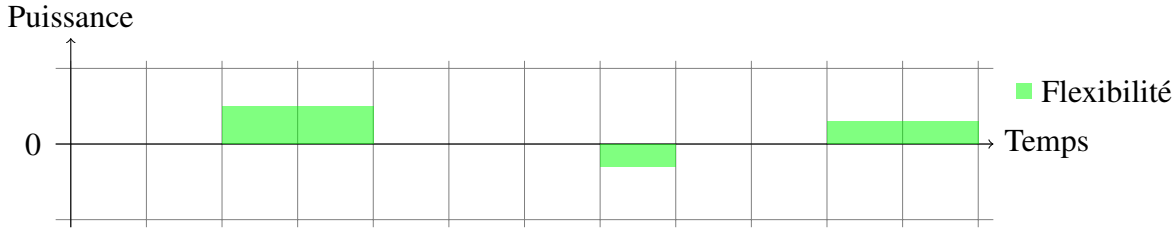

FIGURE 2.14 – Exemple de courbe de flexibilité d'un procédé de consommation pilotable

- **Les procédés de déplacement de la consommation fatale :** ces procédés sont à mi-chemin entre les procédés de consommation fatale et les procédés de consommation pilotable. La consommation n'est pas pilotable mais elle peut être stockée localement a priori afin d'être utilisée pendant les pointes de consommation. ces procédés permettent par exemple de consommer moins sur une période à condition de compenser par la suite (déplacement), ou de consommer peu sur une longue période plutôt que beaucoup sur une courte période (lissage). Un radiateur peut par exemple être éteint ou baissé quelques temps sans que l'occupant du bâtiment soit gêné. Il faudra par contre compenser cette baisse ultérieurement pour retrouver la température de confort souhaitée. Plusieurs procédés de déplacement de la consommation fatale existent actuellement :

  - PULSSI (de la société Watteco) : il s'agit d'un système de gestion de l'énergie électrique. Il permet de délester en cas de besoin un ou plusieurs appareils électriques (radiateur, ballon d'eau chaude,...) en utilisant leur inertie thermique (cf. Figure 2.17) ;
  - le système PAC : il s'agit d'un système de pilotage d'une pompe à chaleur qui permet de lisser la courbe de consommation d'électricité. Le principe consiste à utiliser un ballon de stockage d'eau chaude. Le dimensionnement de ce stockage thermique doit permettre de couper la pompe à chaleur pendant les périodes d'extrême pointe en hiver, sans dégradation du confort (cf. Figure 2.18) ;
  - un système de recharge intelligente de la batterie d'un véhicule : un tel système permet de répartir la recharge de la batterie sur l'ensemble du temps d'arrêt du véhicule.

  Pour ce type de procédés, la puissance par défaut est le besoin "normal" en énergie du procédé. Par exemple pour le système PAC, c'est la quantité d'énergie nécessaire pour maintenir une température de confort dans l'habitation. La figure 2.19 représente l'exemple d'une courbe de charge d'un procédé de déplacement de la consommation fatale avec en bleu la courbe de la puissance par défaut et en rouge la courbe de la puissance modulée. Nous voyons sur cette figure que la charge est déplacée. La figure 2.20 montre la courbe de flexibilité associée à la figure 2.19.

- **Les procédés de déplacement de la consommation :** ces procédés permettent de positionner dans le temps une consommation prévue. Par exemple, un système de pilotage d'une machine à laver pourra permettre le démarrage de la machine à un instant précis, par exemple pendant un moment de surproduction. Pour ce type de procédés, la puissance par défaut est la consommation pour un départ à l'instant "normal". La figure 2.21 représente l'exemple d'une courbe de charge d'un procédé de déplacement d'une consommation avec en bleu la courbe de la puissance par défaut et en rouge la courbe de la puissance modulée. La figure 2.22 montre la courbe de flexibilité associée à la figure 2.21.

- **Les procédés de stockage :** ces procédés permettent de charger et décharger un stock. Par exemple le procédé MICROSCOPE (Micro-Injecteur pour Optimiser par le Stockage la Courbe de Production Électrique) permet de stocker de l'énergie provenant du réseau, et de la restituer (quelques kilowatt-heures) en fonction des besoins du système électrique. Il est composé d'une batterie et d'un panneau photovoltaïque connectés au réseau électrique (cf. Figure 2.23). Pour ce type de procédés, la puissance par défaut est la puissance normale respectant le plan de charge/décharge normal du procédé qui peut consister par exemple à charger pendant les heures creuses et à décharger pendant les heures pleines. La figure 2.24 représente l'exemple d'une courbe de charge d'un procédé de stockage avec en bleu la courbe de la puissance par défaut et en rouge la courbe de la puissance modulée. La figure 2.25 montre la courbe de flexibilité associée à la figure 2.24.

Pour résumer, nous avons identifié huit catégories :

1. **les procédés de consommation fatale :** par exemple une chaine hifi ;

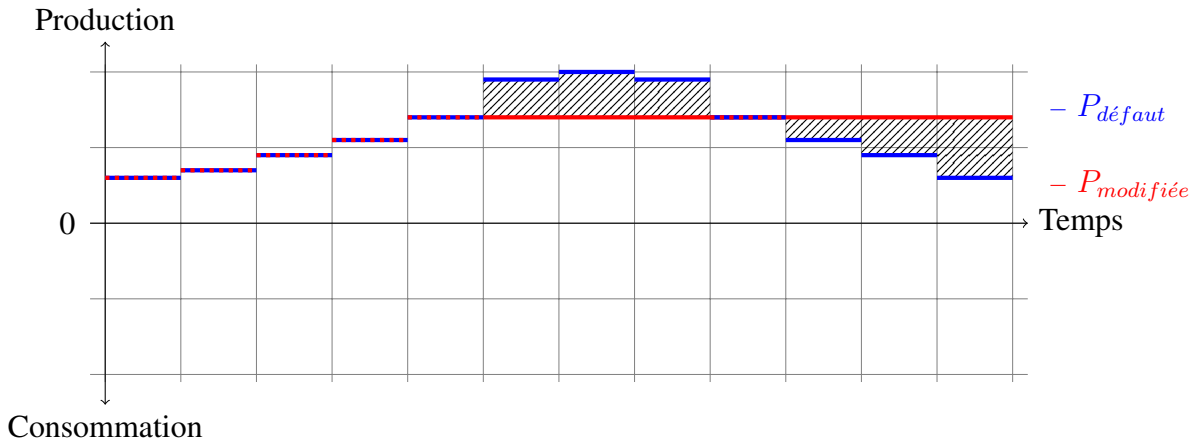

FIGURE 2.15 – Exemple de courbe de charge d'un procédé de déplacement de la production fatale

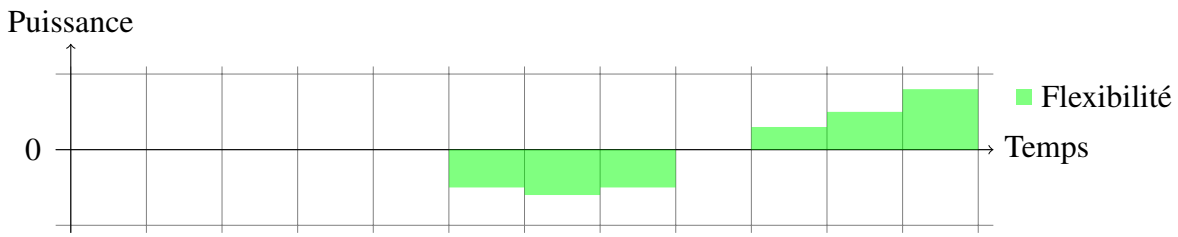

FIGURE 2.16 – Exemple de courbe de flexibilité d'un procédé de déplacement de la production fatale

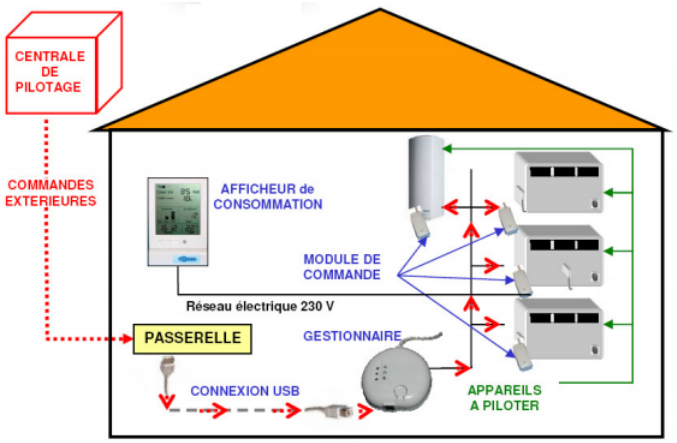

FIGURE 2.17 – Fonctionnement du système PULSSI

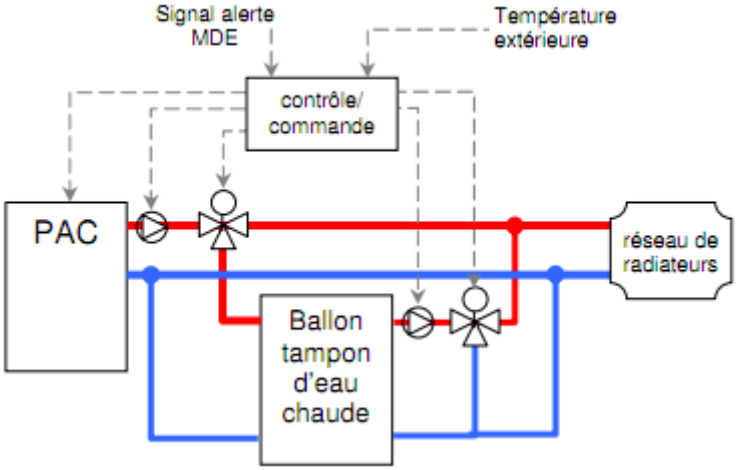

FIGURE 2.18 – Fonctionnement du système PAC

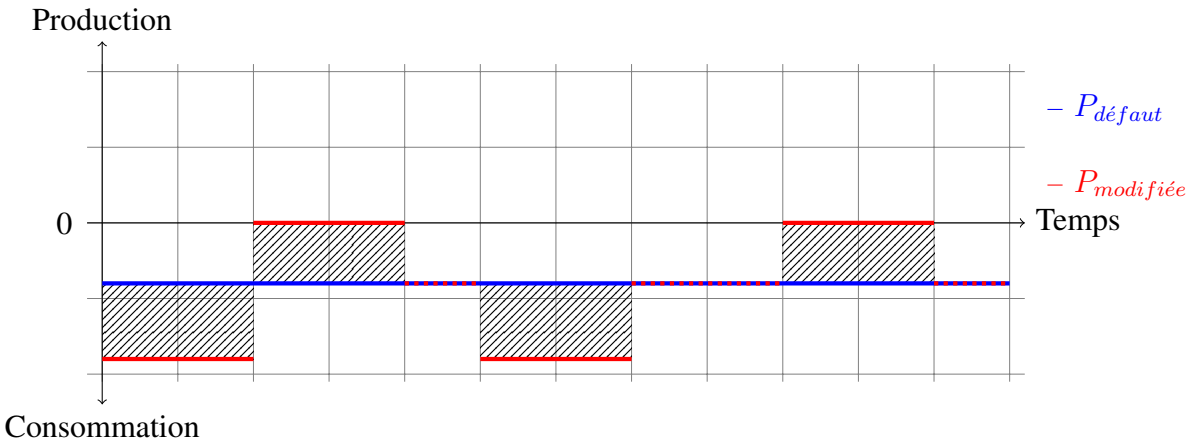

FIGURE 2.19 – Exemple de courbe de charge d'un procédé de déplacement de la consommation fatale

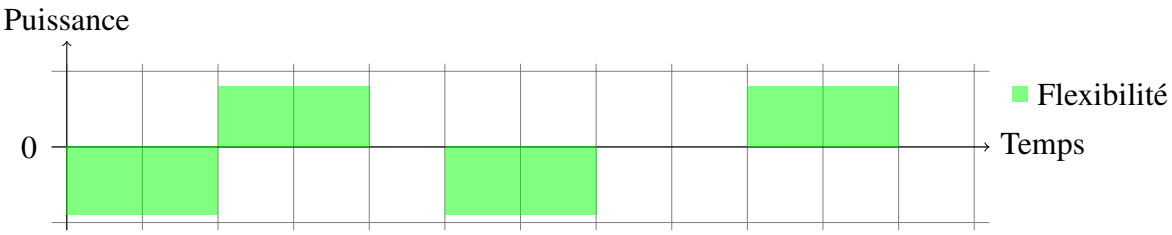

FIGURE 2.20 – Exemple de courbe de flexibilité d'un procédé de déplacement de la consommation fatale

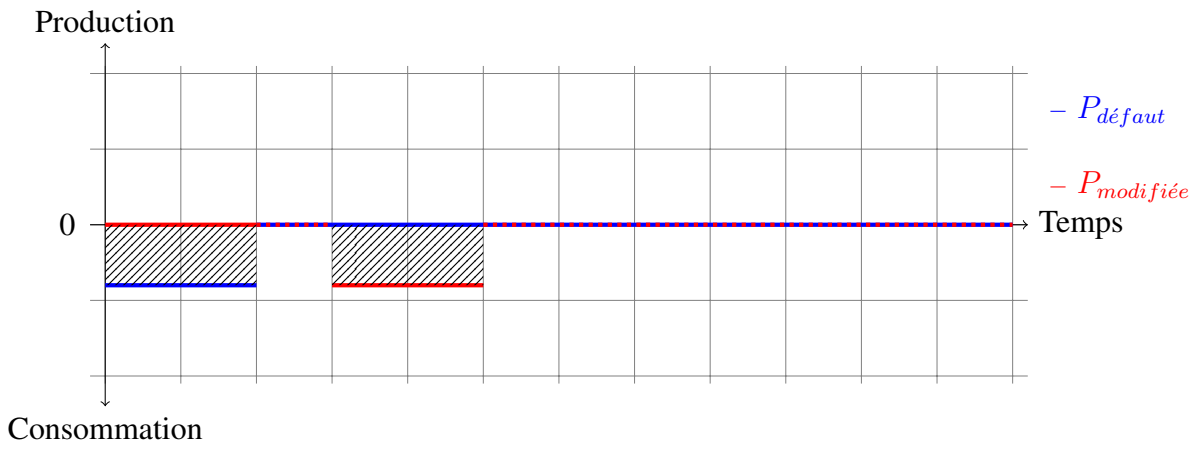

FIGURE 2.21 – Exemple de courbe de charge d'un procédé de déplacement d'une consommation

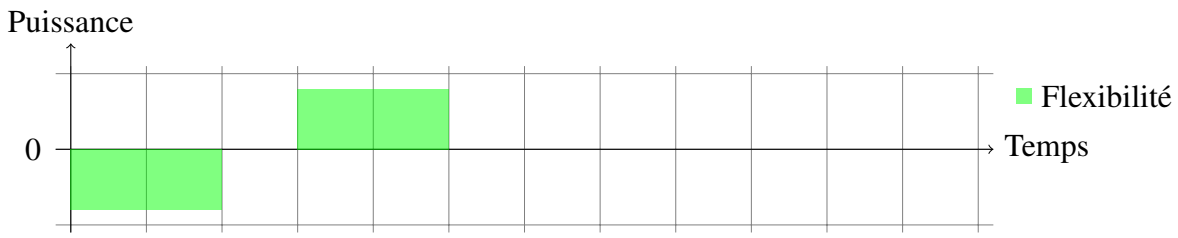

FIGURE 2.22 – Exemple de courbe de flexibilité d'un procédé de déplacement d'une consommation

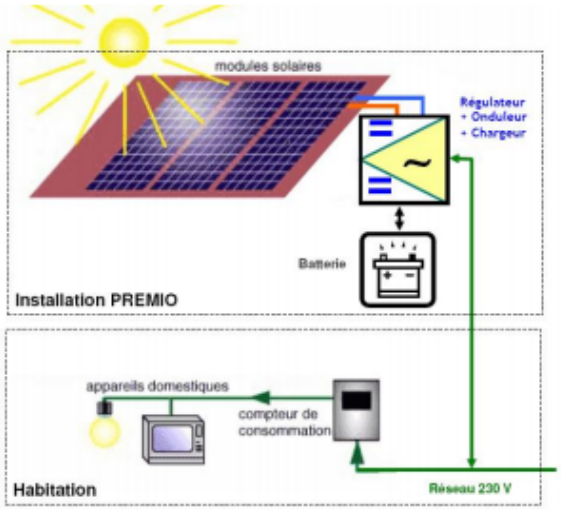

FIGURE 2.23 – Fonctionnement du système MICROSCOPE

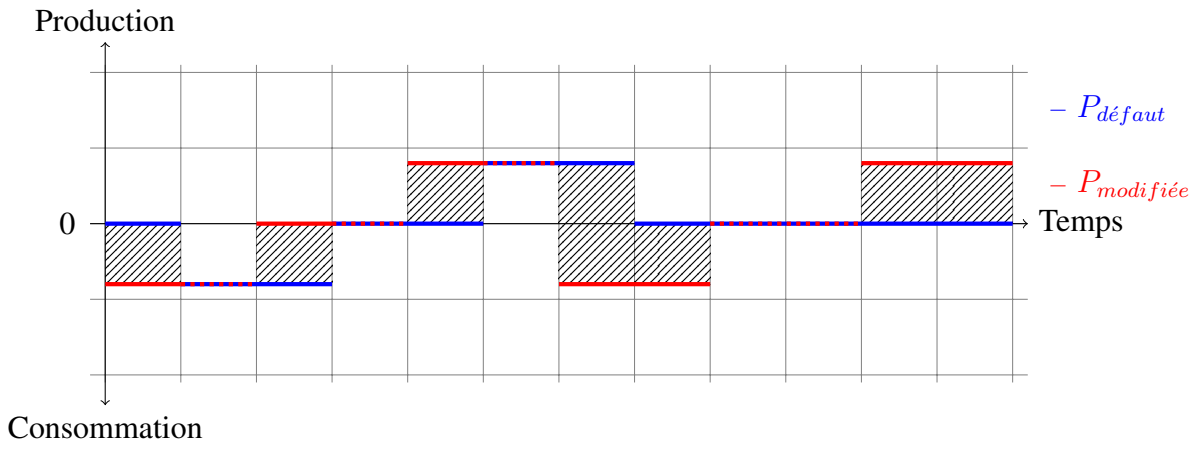

FIGURE 2.24 – Exemple de courbe de charge d'un procédé de stockage

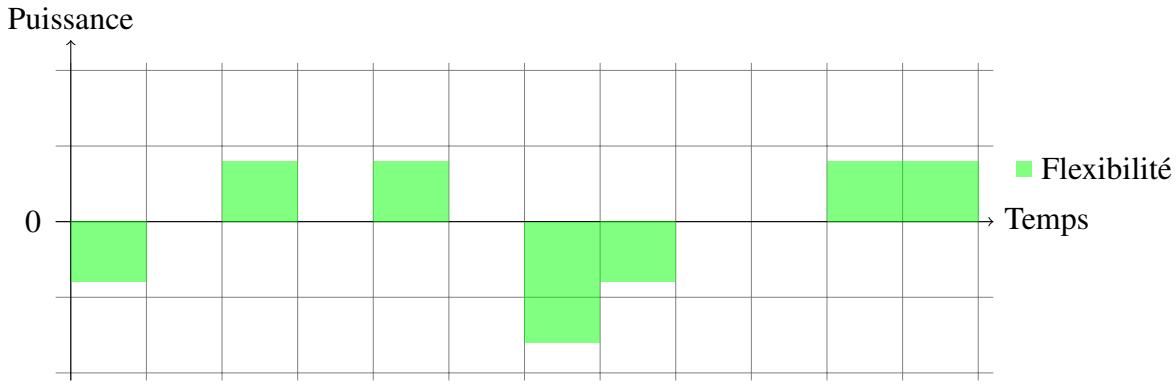

FIGURE 2.25 – Exemple de courbe de flexibilité d'un procédé de stockage

2. **les procédés de production fatale :** par exemple une éolienne ;
3. **les procédés de production pilotable :** par exemple une turbine à combustion ;
4. **les procédés de consommation pilotable :** par exemple un pilotage de DEL ;
5. **les procédés de déplacement de la consommation fatale :** par exemple une gestion du chauffage ;
6. **les procédés de déplacement de la production fatale :** par exemple un stockage de l'énergie solaire thermique ;
7. **les procédés de déplacement de la consommation :** par exemple un pilotage de l'électroménager ;
8. **les procédés de stockage :** par exemple une batterie.

Ces procédés peuvent être situés chez des particuliers, dans des entreprises, dans des collectivités territoriales ou dans des écoles. De plus ces procédés peuvent être pilotés soit directement, soit par incitation. Dans le cas du pilotage direct, le fonctionnement du procédé est connu. Il est donc possible d'agir sur sa puissance. Dans le cas du pilotage par incitation, le procédé réagit de façon autonome à un signal qui lui est envoyé. Il consiste par exemple à proposer à un consommateur un prix bas à une période donnée pour l'inciter à consommer plus à ce moment là (mais le consommateur reste libre de ses décisions).

Comparativement à un responsable d'équilibre qui gère des procédés "classiques", un responsable d'équilibre qui gère, entre autre, une centrale d'agrégation a des moyens d'actions supplémentaires pendant un pic de consommation. Afin d'augmenter la production, il peut en plus des actions classiques :

- augmenter la production des procédés de production pilotable ;
- déplacer une partie de la production en utilisant les procédés de déplacement de la production fatale (cf. Figure 2.27) ;
- lisser la production fatale d'un appareil en utilisant les procédés de déplacement de la production fatale (cf. Figure 2.28).

Et afin de diminuer la consommation, il peut aussi :

- effacer (i.e. réduire) la consommation en agissant sur les procédés de consommation pilotable (cf. Figure 2.26) ;
- reporter une partie de la consommation en utilisant soit les procédés de déplacement de la consommation fatale soit les procédés de déplacement d'une consommation ;
- lisser la consommation d'un appareil en utilisant les procédés de déplacement de la consommation fatale.

Inversement, pendant les pics de production fatal, le responsable d'équilibre peut en plus des actions classiques :

- diminuer la production des procédés de production pilotable ;
- reporter une partie de la production en utilisant les procédés de déplacement de la production fatale ;
- lisser la production fatale d'un appareil en utilisant les procédés de déplacement de la production fatale ;
- augmenter la consommation en agissant sur les procédés de consommation pilotable ;

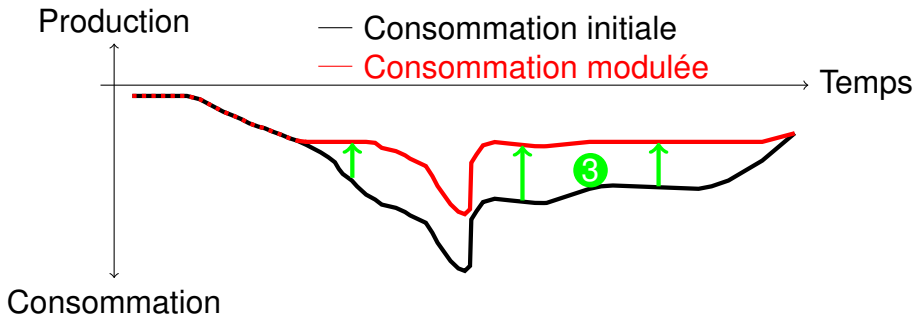

FIGURE 2.26 – Moyen d'action numéro 3 : effacement de la consommation

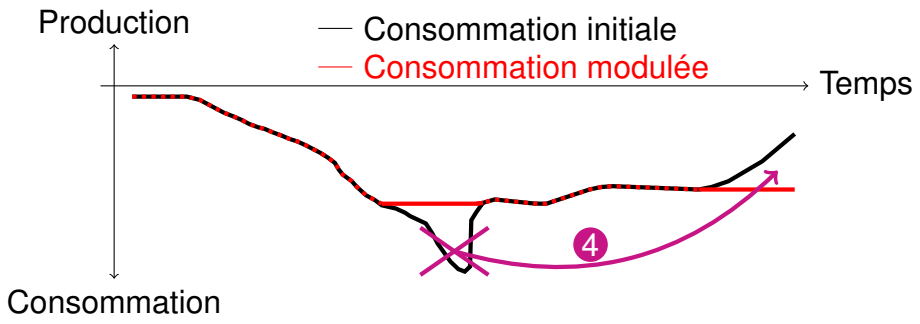

FIGURE 2.27 – Moyen d'action numéro 4 : reporter la consommation

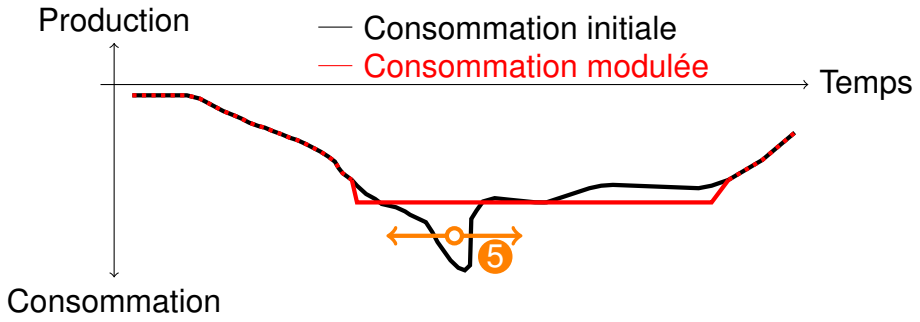

FIGURE 2.28 – Moyen d'action numéro 5 : lisser la consommation

- déplacer une partie de la consommation en utilisant soit les procédés de déplacement de la consommation fatale soit les procédés de déplacement d'une consommation ;
- de lisser la consommation d'un appareil en utilisant les procédés de déplacement de la consommation fatale.

Étant donné qu'une baisse de consommation a le même impact sur le réseau qu'une augmentation de production (nous avons pu le constater sur les figures 2.12 et 2.14), tous les procédés de consommation peuvent être vus comme des procédés de production "inversés". Ainsi, les procédés de consommation fatale et les procédés de production fatale sont similaires et peuvent être regroupés dans une même classe. De la même façon, les procédés de production pilotable et les procédés de consommation pilotable peuvent également être regroupés dans une même classe, tout comme les procédés de déplacement de la production fatale et de déplacement de la consommation fatale.

De plus, comme nous l'avons vu, les procédés de déplacement de la consommation fatale et de déplacement de la production fatale comportent un stock et fonctionnent de façon similaire à un procédé de stockage. En effet, un procédé de stockage va stocker de l'électricité à certains moment (et donc soutirer de l'électricité au réseau), et se décharger à d'autres moments (et donc injecter de l'électricité dans le réseau). De la même façon, un procédé de déplacement de la consommation fatale pourra stocker (et donc consommer plus que par défaut, i.e. soutirer plus d'électricité) à une période donnée, puis se décharger (et donc consommer moins, i.e. injecter de l'électricité dans le réseau) à une autre période. Un procédé de déplacement de la production, lui, pourra stocker (et donc produire moins, i.e. soutirer de l'électricité au réseau) à certains moments, puis décharger son stock (et donc produire plus, i.e. injecter) à d'autres moments. Ces trois types de procédés ont donc des comportements identiques et peuvent être regroupés.

Nous distinguerons finalement par la suite quatre groupes de procédés :

- les procédés fatals (PF) qui incluent les procédés de production et de consommation fatales ;
- les procédés de déplacement d'une consommation (PDC) ;
- les procédés sans stockage (PSS) qui incluent les procédés de production et de consommation pilotable ;
- les procédés avec stockage (PAS) qui incluent les procédés de déplacement de la production et de la consommation fatale et ceux de stockage.

Par la suite, nous utiliserons ces acronymes pour désigner les différents types de procédés.

Ces procédés sont soumis à diverses contraintes : des contraintes techniques (contraintes dynamiques), et des contraintes "de confort". Les contraintes techniques sont intrinsèques aux procédés, elles doivent être respectées obligatoirement. Par exemple, dans un système de gestion de l'éclairage public qui contrôle $400\ W$ de DEL, les DEL sont contrôlées $100\ W$ par $100\ W$. Ainsi, la puissance peut être égale à :

- $400\ W$ quand toutes les DEL sont allumées ;
- $300\ W$ quand un quart des DEL sont éteintes ;
- $200\ W$ quand la moitié des DEL sont éteintes ;
- $100\ W$ quand trois-quart des DEL sont éteintes ;
- $0\ W$ quand toutes les DEL sont éteintes.

La puissance de ce système est donc discrétisée.

Les contraintes dites de confort modélisent un souhait de confort du procédé (pour éviter par exemple une usure prématurée du procédé) ou du confort de ses utilisateurs. Par exemple, la pompe à chaleur du système PAC ne doit pas être allumée puis éteinte immédiatement afin de limiter des casses matérielles. Une

durée minimale d'utilisation de ce système est donc à respecter. Un autre exemple est le cas des radiateurs contrôlés par le système PULSSI qui ne peuvent pas être éteints trop longtemps sous peine de générer trop d'inconfort pour l'utilisateur. Il faut donc contraindre la durée maximale d'utilisation de ce système.

Nous pouvons noter qu'outre le fait qu'elle permette de rendre le réseau plus flexible, la centrale d'agrégation offre au responsable d'équilibre la possibilité de lisser la consommation. Il peut ainsi augmenter la part de production de la base, ce qui revient moins cher (cf. Figures 2.29 et 2.30). De plus, lors d'un blackout, le responsable d'équilibre a la possibilité, grâce à cette centrale d'agrégation, de préserver une partie du réseau en réalisant un îlotage : cette partie du réseau n'échange alors pas d'électricité avec l'extérieur, elle est auto-suffisante. Elle réalise donc son propre équilibre offre/demande.

2.7 Objectif général de cette thèse

L'objectif général de cette thèse est de proposer un système de pilotage pour une centrale virtuelle d'agrégation de flexibilité afin qu'elle réponde aux exigences d'un responsable d'équilibre.

Nous allons considérer dans cette thèse comme cas d'utilisation d'une centrale virtuelle d'agrégation de flexibilité, une centrale d'agrégation "reliée" virtuellement à un poste source du réseau de transport (comme tous les autres moyens de production) et pouvant gérer tout type de procédé relié à un réseau de distribution. Cette centrale d'agrégation devra donc pouvoir contenir n'importe quel type de procédés, y compris d'autres centrales d'agrégation. Elle devra de plus pouvoir être gérée comme n'importe quel autre moyen de production. Ceci implique qu'il faudra être capable de déterminer les données qui caractérisent sa production, et de répartir les demandes de production sur ses procédés. Elle devra enfin satisfaire les contraintes liées aux réseaux de distribution, c'est-à-dire sa puissance devra respecter la capacité maximale des câbles et la puissance minimale et maximale des postes électriques.

La proposition d'un tel système de pilotage implique tout d'abord de déterminer une architecture et un mode de gestion de la centrale d'agrégation. Il s'agira ensuite de calculer les données qui caractérisent sa production avec au moins une méthode n'utilisant que des logiciels gratuits car il est difficilement envisageable de déployer chez tous les acteurs de la centrale d'agrégation une application utilisant des logiciels payants. Dans cette thèse nous nous intéresserons à l'utilisation de la centrale virtuelle d'agrégation de flexibilité pour le processus de répartition de l'offre sur la demande de la veille pour le lendemain, mais aussi pour le mécanisme d'ajustement. Ceci nécessitera des algorithmes ayant un temps de réponse court (moins de une minute).

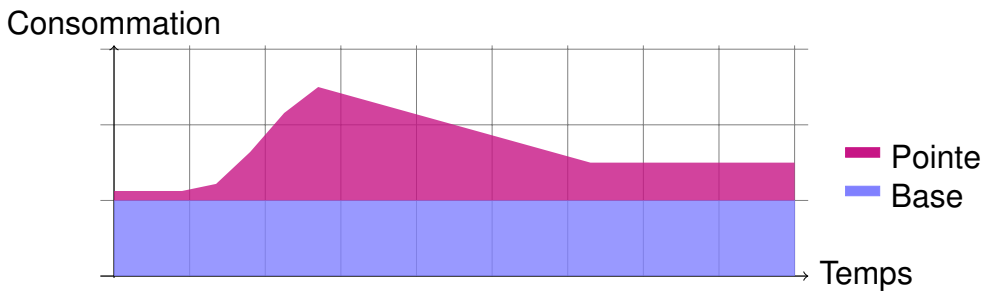

FIGURE 2.29 – Courbe de charge sans lissage

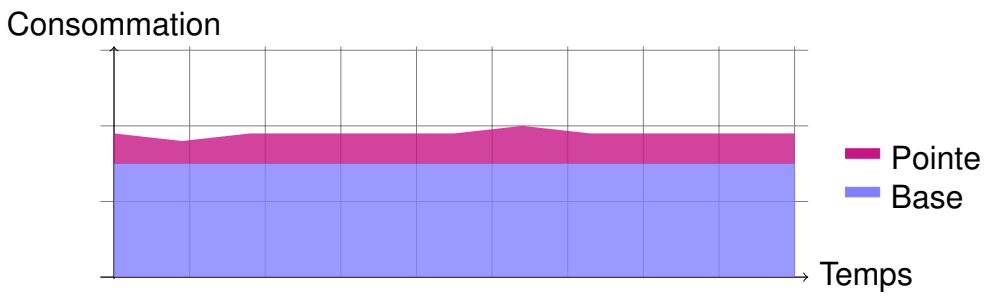

FIGURE 2.30 – Courbe de charge avec lissage

3

Architecture et mode de gestion

3.1 État de l'art

Dans le chapitre précédent, nous avons vu qu'il existe deux façons d'intégrer des ENR dans un réseau : les rendre non fatales en les associant à d'autres procédés, ou rendre le réseau plus flexible en y intégrant des procédés permettant de moduler la production et la consommation.

Les premiers chercheurs qui ont travaillé sur les centrales d'agrégation se sont intéressés à des centrales d'agrégation rendant les ENR non fatales en les regroupant avec des procédés avec lesquels ils échangent de l'énergie. Ainsi des chercheurs ont proposé de coupler des ENR avec des moyens de stockage : un moyen de stockage sous forme thermique dans des moyens de cogénération [Wille-Haussmann *et al.*, 2010], des moyens de cogénération, des ENR et des batteries [Gu *et al.*, 2010], une turbine à gaz, des ENR et une batterie de grande capacité [PEGASE, 2013] et [Sobu and Wu, 2012b], une centrale thermique à flamme et une STEP [Cohen and Wan, 1985] et [Aoki *et al.*, 1987], des éoliennes et des procédés de stockage [Sanchez de la Nieta *et al.*, 2013], [Green and Sanduk, 2013] et [Kahrobaee *et al.*, 2013], de l'énergie solaire thermique avec un procédé de stockage [Tian and Zhao, 2013], des éoliennes et des panneaux solaires avec un procédé de stockage [Sobu and Wu, 2012a] et [Wang and Yang, 2013], ou encore des éoliennes avec des véhicules électriques [Vachirasricirikul and Ngamroo, 2014]. D'autres chercheurs ont étudié des centrales combinées gaz hybrides solaires [Buck *et al.*, 2002], [Kane *et al.*, 2003] et [Schwarzbözl *et al.*, 2006]. Tous ces travaux traitent de centrales regroupant peu de procédés qui interagissent entre eux de façon très spécifiques.

D'autres travaux proposent des architectures pour un ensemble de procédés qui n'interagissent pas obligatoirement ensemble. [Gudi *et al.*, 2010], [Pedrasa *et al.*, 2010], [Fernandes *et al.*, 2011], [Oliveira *et al.*, 2011], [Xiong *et al.*, 2011], [Goya *et al.*, 2011a], [Georgievski *et al.*, 2012], [Tanaka *et al.*, 2012], [Agnetis *et al.*, 2013], [Kim and Lavrova, 2013], [Zhao *et al.*, 2013] et [Yoon *et al.*, 2014] se sont intéressés à des procédés de consommation dans un même bâtiment. L'objectif est de diminuer la facture énergétique de l'ensemble. [Barbato *et al.*, 2011] ont travaillé sur le même problème en ajoutant un moyen de stockage au bâtiment. Dans [Nguyen and Le, 2014] et [Tushar *et al.*, 2014] les auteurs proposent une architecture pour un ensemble de véhicules électriques couplés à un bâtiment. [Bozchalui *et al.*, 2012] et [Zhao and Wu, 2014] ont couplé des ENR et un bâtiment et [Chen *et al.*, 2013] des data centers. Plus généralement [Oshiro *et al.*, 2012] ont couplé des demandes modulables avec du stockage. Au sein des projets PREMIO [PREMIO, 2007], NiceGrid [NiceGrid, 2011] et ADDRESS [ADDRESS, 2008], la centrale d'agrégation étudiée pilote tous les types de procédés situés dans une même région. L'architecture proposée dans tous ces travaux est à un seul niveau et composée d'un coordinateur communiquant avec ces procédés (cf. Figure

3.1).

Ces architectures utilisent deux types de mode de gestion de procédés différents : un pilotage direct comme dans le projet PREMIO [PREMIO, 2007], ou un pilotage incitatif comme dans le projet ADDRESS [ADDRESS, 2008]. Au sein de PREMIO, le pilotage direct proposé consiste dans un premier temps à l'envoi par un client à la centrale d'agrégation de périodes d'alerte utilisant un code couleur (niveau d'alerte blanc, orange ou rouge) correspondant à des périodes où la consommation est plus ou moins élevée ou à des périodes de pollution plus ou moins forte. La figure 3.2 représente un exemple de périodes d'alerte envoyées par le client. Plus le niveau de la période d'alerte est haut, plus il est profitable pour la centrale d'agrégation de proposer une grande puissance. La centrale de pilotage de PREMIO calcule à partir d'une modélisation et des données de chaque procédé qu'elle pilote, sa puissance maximale pour chaque période de temps en essayant de maximiser sa puissance sur les plages d'alerte de niveau élevé (cf. Figure 3.3). Ensuite le client centralise les puissances de tous les moyens de production et décide des quantités qu'il va demander à la centrale d'agrégation. Dans une seconde phase, la centrale décide de la répartition de la puissance demandée par le client sur ses moyens de production tout en minimisant son coût de production (cf. Figure 3.4). Cette gestion est limitée car PREMIO ne peut pas par exemple piloter des procédés par incitation. Le projet NiceGrid propose un mode de gestion semblable.

Le projet ADDRESS, quant à lui, ne regroupe que des procédés d'effacement de la consommation. La modélisation exacte de ces procédés n'est pas connue (nous pouvons imaginer qu'un procédé est un consommateur dont nous ne connaissons pas le comportement). Cependant, à chaque procédé est associé un profil de consommation, c'est à dire une représentation statistique de sa consommation (ou production). Lors d'une sollicitation de la centrale de pilotage par le client, un signal de prix est déterminé pour chacun des profils. Ce signal de prix est une courbe qui représente le prix proposé aux procédés du profil considéré pour une puissance produite et pour un pas de temps donné. Ce signal est ensuite envoyé par la centrale de pilotage aux procédés qui vont optimiser automatiquement leur consommation ou leur production (cf. Figure 3.5). Il existe une limite à ce type de gestion car il est très difficile de prévoir le comportement de la centrale de pilotage suite à la réception du signal de prix, et le seul moyen d'action du client pour modifier la consommation ou la production est de modifier le signal de prix envoyé.

Concernant l'architecture utilisée dans tous ces travaux, elle ne permet pas de tenir compte de toutes les contraintes liées au réseau. En effet, il serait facile d'intégrer les contraintes de puissance minimale et maximale du poste source au niveau du coordinateur. Cependant, cette architecture à un seul niveau ne permet pas de tenir compte des contraintes réseau des postes intermédiaires (postes HTA/BT, compteurs), ni des capacités des câbles les reliant. Elle n'est donc pas adaptée à notre contexte.

Actuellement, il n'y a donc aucune architecture et aucun mode de gestion qui réponde totalement aux exigences d'un responsable d'équilibre.

3.2 Architecture proposée

L'architecture que nous proposons est présentée Figure 3.6. Elle est basée sur l'architecture de la Figure 3.1 mais elle est multi-niveaux. Le nœud racine correspond au client (poste source du réseau de distribution). Ce nœud est relié au nœud associé au coordinateur de la centrale. Les nœuds des niveaux inférieurs sont soit des procédés pilotés directement, soit des procédés pilotés par incitation, soit des centrales d'agrégation qui elles-mêmes regroupent des procédés ou des centrales.

Étant donné que les procédés pilotés par incitation et les centrales d'agrégation peuvent être considérés comme des "boites noires" qui remontent des informations au coordinateur du niveau supérieur sans que ce dernier n'ait la possibilité d'agir, nous regrouperons ces deux types de nœuds sous l'appellation "coordinateur". À chaque nœud de l'architecture sera donc associé un client et des procédés pilotés directement et des coordinateurs (cf. Figure 3.7).

Cette architecture multi-niveaux permet de modéliser l'arborescence ainsi que les contraintes du réseau de distribution en "plaçant" un coordinateur au niveau d'un nœud du réseau de distribution. Elle permet

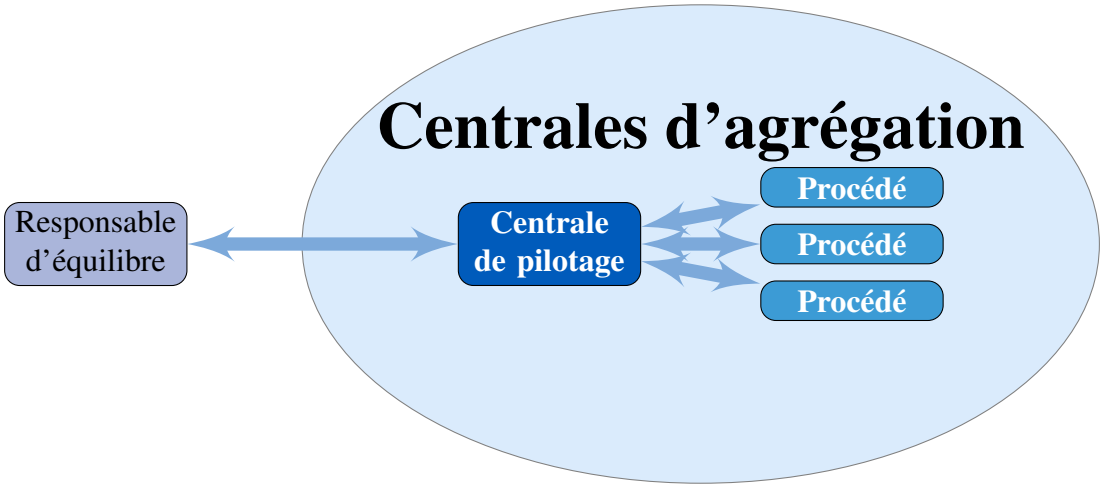

FIGURE 3.1 – Architecture d'un ensemble de procédés situés dans un secteur géographique précis

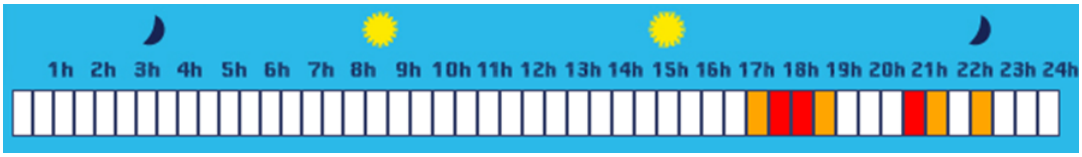

FIGURE 3.2 – Périodes d'alerte

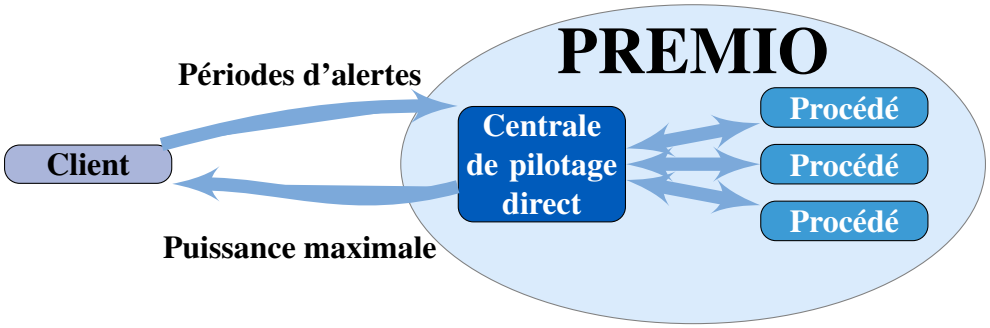

FIGURE 3.3 – Phase 1 de PREMIO

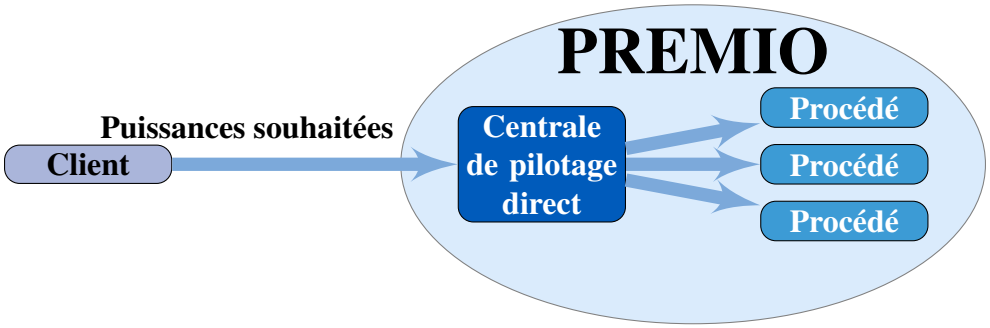

FIGURE 3.4 – Phase 2 de PREMIO

aussi d'intégrer tous les types de procédés y compris d'autres centrales d'agrégation.

Il est à noter que nous supposons que chaque procédé piloté directement peut être considéré indépendamment des autres. En effet, des procédés qui interagissent entre eux seront modélisés comme un unique procédé. De plus, la figure 3.8 représente l'architecture du cas test que nous allons utiliser tout au cours de cette thèse. Un coordinateur sera placé au niveau du poste source ainsi qu'au niveau de chaque poste HTA/BT et au niveau de chaque compteur d'un consommateur/producteur. Dans cette configuration, nous aurons :

- environ 50 coordinateurs de poste HTA/BT pour chaque poste source ;
- environ 350 coordinateurs de compteur pour chaque poste HTA/BT ;
- entre 1 et 10 procédés par compteur.

Il s'agit du nombre de niveaux minimal permettant de représenter un réseau de distribution. Des niveaux supplémentaires peuvent être ajoutés. Par exemple nous pouvons ajouter un coordinateur au dessus de celui du poste source qui coordonnerait plusieurs réseaux de distribution. Et nous allons considérer dans le cas test pour simplifier qu'il n'y a pas de procédés reliés directement aux postes HTA/BT et aux postes sources.

3.3 Mode de gestion proposé

Nous allons maintenant expliquer comment chaque coordinateur va interagir au travers de cette architecture avec ses procédés afin de déterminer, comme pour tout autre moyen de production, les données qui caractérisent la production de sa centrale d'agrégation pour ensuite les transmettre au responsable d'équilibre. Le fonctionnement résumé dans la figure 3.9 est le même pour chaque coordinateur, quel que soit sont niveau dans l'architecture :

Phase 1 : **Envoi des souhaits du client (coordinateur du niveau supérieur) au coordinateur.**

Nous verrons dans la section suivante, quelle forme prennent ces souhaits.

Phase 2 : **Envoi des souhaits du coordinateur à ses procédés pilotés directement et à ses coordinateurs pilotés.**

Le coordinateur transmet à son tour ses souhaits aux procédés qu'il pilote directement et à ses coordinateurs pilotés. Ces souhaits peuvent être différents de ceux qui lui ont été transmis par le client. Ils peuvent également être différents pour chacun des procédés et des coordinateurs.

Phase 3 : **Calcul puis envoi du plan de production des procédés pilotés directement et des coordinateurs pilotés.**

En fonction des souhaits reçus, chaque procédé et chaque coordinateur calcule son plan de production en maximisant son profit, puis l'envoie au coordinateur.

Phase 4 : **Agrégation des plans de production des procédés pilotés directement et des coordinateurs pilotés.**

Après avoir reçu tous les plans de production des procédés, le coordinateur calcule puis transmet au client son plan de production en maximisant son profit.

Phase 5 : **Décision du client.**

Le client centralise les plans de production de tous les coordinateurs ainsi que des autres moyens de production, décide quelles quantités il va demander à chacun, puis envoie ses ordres de production.

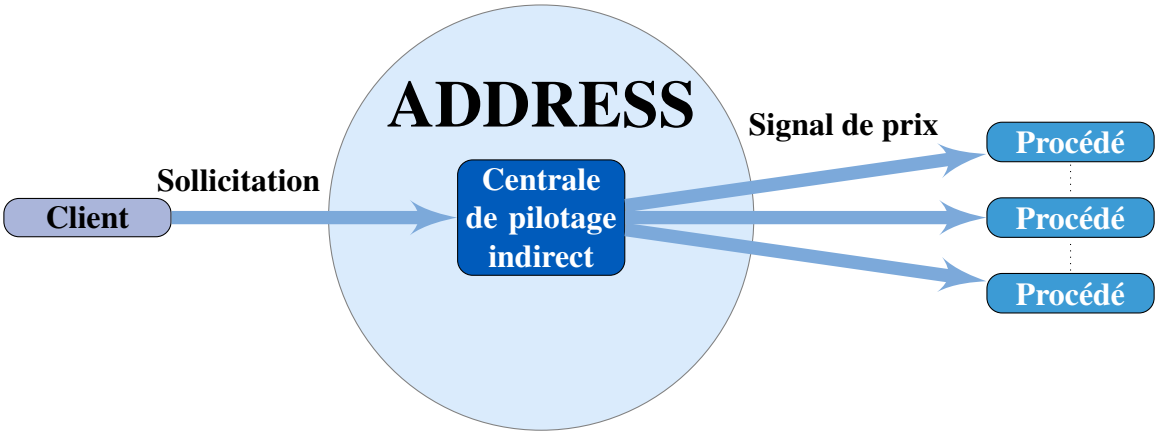

FIGURE 3.5 – Fonctionnement du système ADDRESS

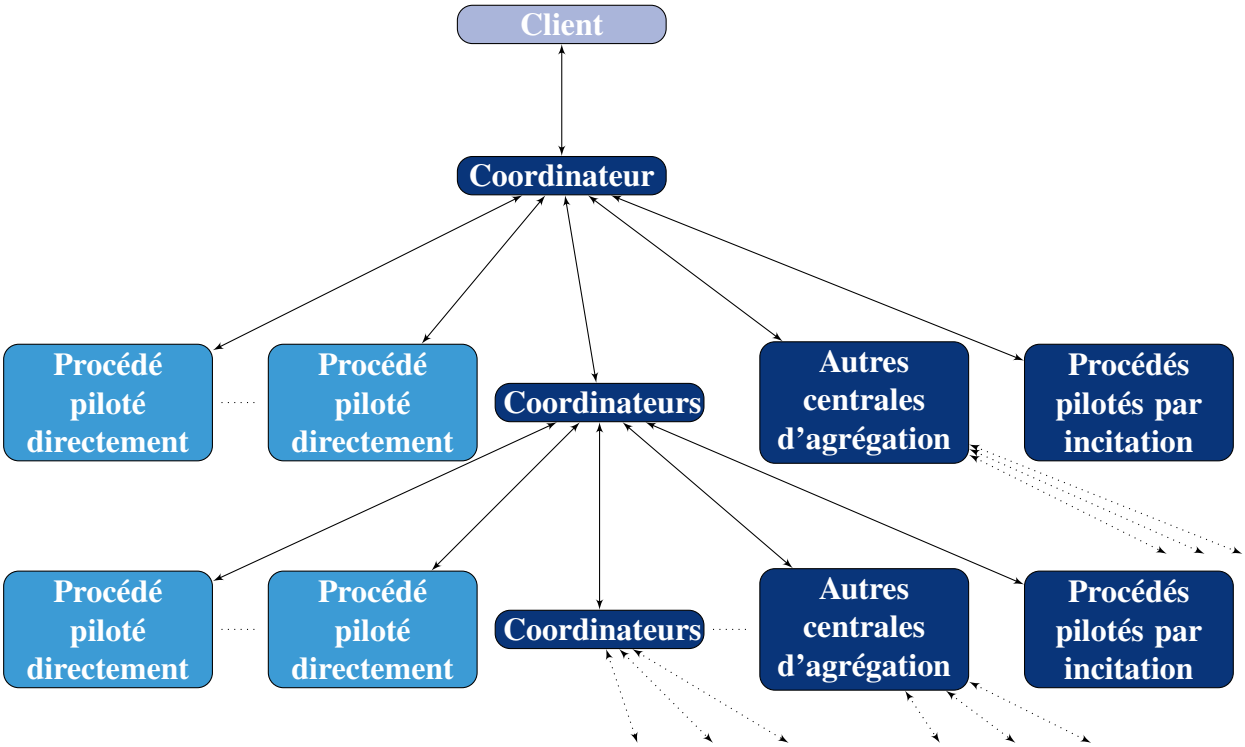

FIGURE 3.6 – Architecture multi-niveaux

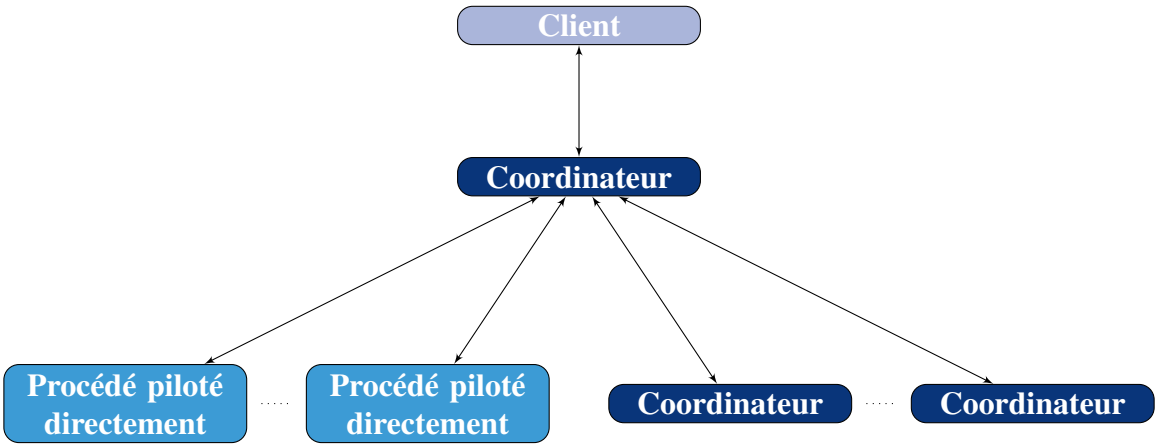

FIGURE 3.7 – Architecture d'un nœud

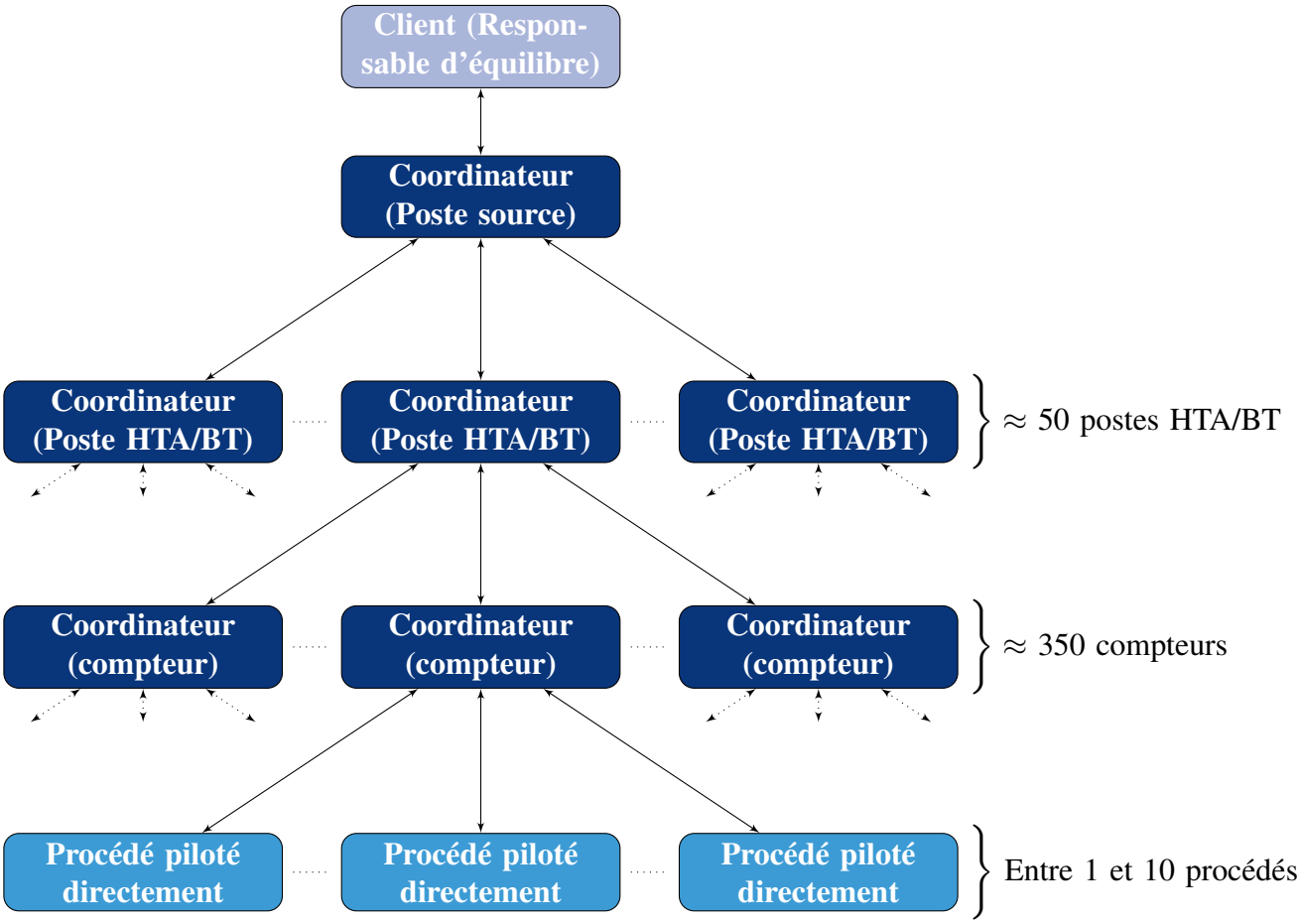

FIGURE 3.8 – Architecture du cas test

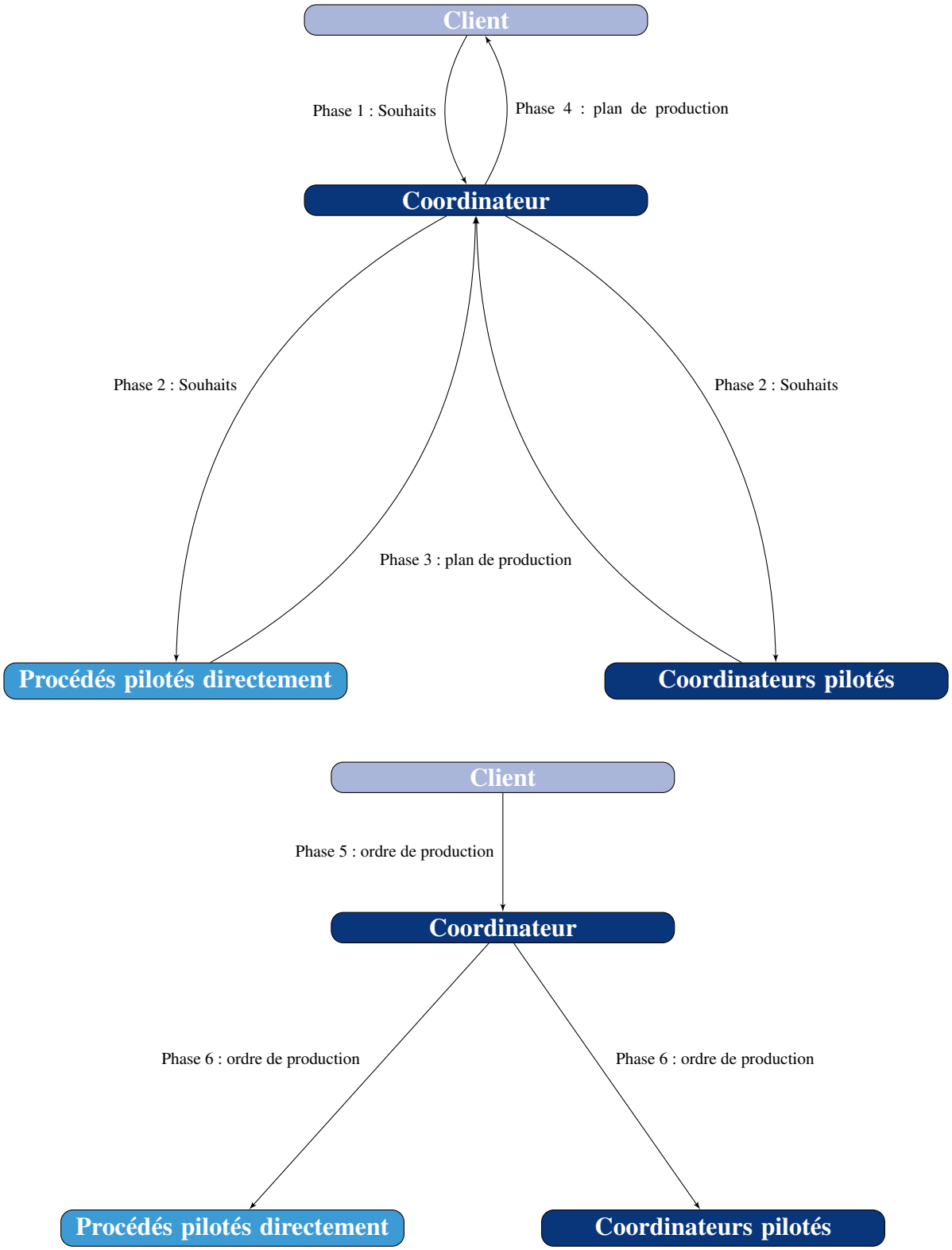

FIGURE 3.9 – Fonctionnement retenu

Phase 6 : **Répartition de la demande du client sur les procédés pilotés directement et les coordinateurs pilotés.**

Si le client choisit d'utiliser la production du coordinateur, le coordinateur envoie des ordres de production à ses procédés pilotés et à ses coordinateurs pilotés.

3.4 Communication entre les acteurs de la centrale d'agrégation

Nous allons dans cette section préciser sous quelle forme les informations qui transitent entre les acteurs de la centrale d'agrégation (souhaits, plan de production) sont envoyées.

Un client peut souhaiter par exemple augmenter la production, diminuer la production, faire un îlotage, lisser la courbe de charge ou satisfaire une demande précise.

Plusieurs options sont possibles : dans le projet PREMIO, les souhaits sont transmis sous forme de périodes d'alerte. Cependant, ce code couleur est peu précis. Une autre possibilité est de transmettre ces souhaits sous forme d'une courbe de demande à satisfaire. Cependant, la centrale d'agrégation peut avoir des difficultés à satisfaire une courbe de demande sans marge. Une autre option est l'envoi d'un signal de prix (comme dans le projet ADDRESS). Un tel signal consiste, pour un pas de temps donné, en une fonction indiquant pour chaque puissance le prix associé. Un tel signal a l'avantage d'être plus souple qu'une courbe de demande, tout en donnant plus d'information qu'une période d'alerte. Cependant, dans le projet ADDRESS, le signal de prix d'un pas de temps donné n'est connu qu'au pas de temps précédent, ce qui ne permet pas de planifier la production sur l'horizon d'une journée.

Nous proposons donc d'utiliser un signal de prix similaire à celui utilisé dans le projet ADDRESS, mais qui serait connu la veille pour tous les pas de temps du lendemain. Ce prix représenterait le prix d'achat par le client de la production du coordinateur. Un tel signal de prix peut permettre au client de répondre à différents objectifs. Par exemple :

- s'il souhaite une augmentation de la production sur un pas de temps, le client va envoyer pour ce pas de temps un signal de prix ressemblant à la figure 3.10. Le coordinateur est alors incité à augmenter sa production car plus il produit (plus la puissance échangée est positive) plus sa rémunération est élevée ;

- s'il souhaite une augmentation de la consommation sur un pas de temps, le client va envoyer pour ce pas de temps un signal de prix ressemblant à la figure 3.11. Le coordinateur est alors incité à augmenter sa consommation car plus il consomme (plus la puissance échangée est négative) plus sa rémunération est élevée ;

- s'il souhaite une augmentation de la production limitée sur un pas de temps, le client envoie pour ce pas de temps un signal de prix ressemblant à la figure 3.12. Le coordinateur est alors incité à augmenter sa production jusqu'à 2 ;

- s'il souhaite un lissage de la courbe d'échange sur un ensemble de pas de temps, le client doit envoyer pour chaque pas de l'horizon de temps un signal de prix ressemblant à la figure 3.13. Le coordinateur est alors incité à échanger $P_{echange}^{moy}$ (la moyenne des puissances échangées sur l'horizon de temps) ;

- s'il souhaite un îlotage sur un pas de temps, le client doit envoyer pour ce pas de temps un signal de prix ressemblant à la figure 3.14. Le coordinateur est alors incité à ne rien échanger avec le client.

La prise en compte d'un signal de prix quelconque pour chaque pas de temps étant assez complexe, nous supposerons dans cette thèse que le signal de prix sera linéaire à chaque pas de temps. Il consistera donc pour chaque pas de temps en un prix par kilowattheure. Avec un tel signal de prix, le client ne pourra avoir que deux objectifs : soit augmenter la production à un pas de temps donné en choisissant un prix positif (cf. Figure 3.10) ou soit augmenter la consommation à un pas de temps donné en choisissant un prix négatif

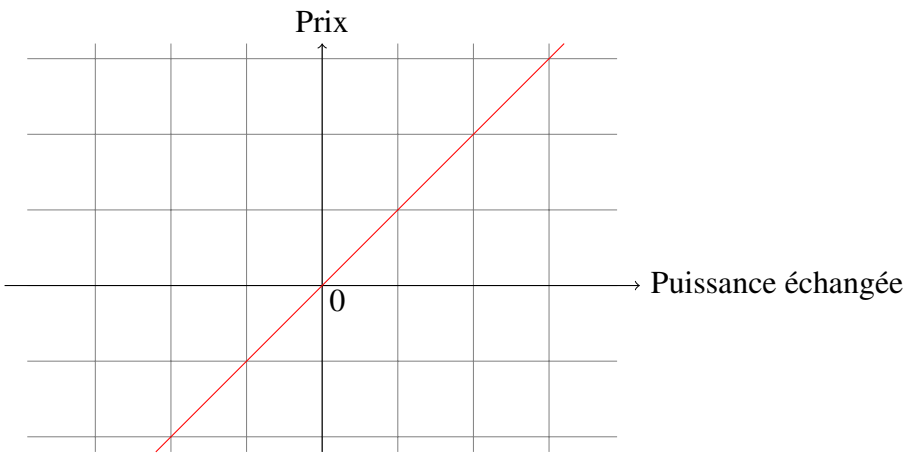

FIGURE 3.10 – Signal de prix induisant une augmentation de la production à un pas de temps donné

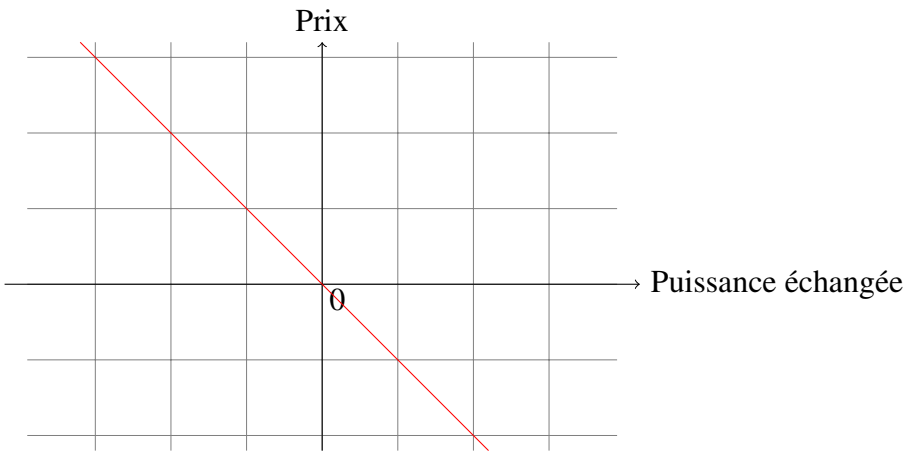

FIGURE 3.11 – Signal de prix induisant une augmentation de la consommation à un pas de temps donné

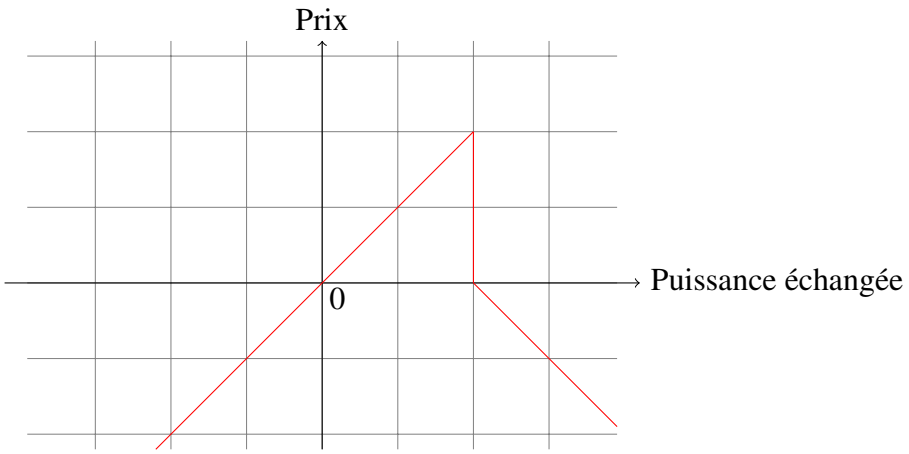

FIGURE 3.12 – Signal de prix induisant une augmentation de la production limitée à un pas de temps donné

(cf. Figure 3.11). Tous les autres objectifs comme par exemple une augmentation de la production limitée, un lissage ou un îlotage nécessitent d'avoir un signal de prix qui dépend du temps et de la puissance. Dans cette thèse, nous ne nous intéresserons pas à la fabrication des prix. Nous les supposerons connus.

Concernant les plans de production remontés au client, ils peuvent prendre la forme des volumes que le moyen de production peut produire sur une période de 24h, et leurs valeurs d'usage pour chaque pas de temps (type 1). Le principal obstacle à la détermination de ces volumes est que les décisions ne sont pas décorrélées du temps : par exemple un chauffage ne peut pas être arrêté sans prendre en compte le fait que le bâtiment devra être chauffé plus par la suite pour retrouver sa température de confort. Ainsi, à chaque volume possible pour un pas de temps t pourra correspondre des volumes proposés pour les pas de temps suivants et précédents différents. L'établissement des données de type 1 consiste alors à déterminer plusieurs plans de production, un pour chaque combinaison réalisable de volumes sur l'horizon, ce qui est extrêmement combinatoire. Une autre solution est que le procédé transmette les puissances minimale et maximale qu'il peut produire pour chaque pas de temps sur une période de 24h, accompagné des contraintes qu'il doit respecter, et du coût marginal de production de chaque pas de temps (type 2). Cependant, il très difficile d'agréger les contraintes de plusieurs procédés. Par exemple, à partir des durées minimales de production de différents procédés, il est très difficile, voire impossible, de déterminer la durée minimale de production de la centrale d'agrégation.

Nous avons donc fait le choix suivant : la centrale d'agrégation va envoyer au client des données de type 1 mais simplifiées (composées uniquement d'une combinaison de volumes pour l'horizon de temps). Il est à noter que le coût marginal sera le prix envoyé par le client. Il aura le choix soit d'accepter les puissances proposées par ses procédés, soit leurs puissances par défaut. Comme l'architecture se répète niveau par niveau, tous les coordinateurs auront aussi le choix soit de sélectionner les puissances proposées soit les puissances par défaut pour chaque procédé ou coordinateur qu'ils pilotent.

Pour résumer, tous les coordinateurs et tous les procédés pilotés directement devront être capables de communiquer à leur client :

- une puissance par défaut pour chaque pas de temps ;
- une puissance proposée pour chaque pas de temps.

Ces données seront par la suite le plan de production du procédé.

3.5 Conclusion

L'architecture et le mode de gestion proposés correspondent bien au cahier des charges puisqu'ils permettent :

- de piloter tous les types de procédé y compris d'autres centrales d'agrégation ;
- de respecter toutes les contraintes réseaux ;
- de réaliser plusieurs types d'objectif comme augmenter la production et diminuer la production. Et faire un îlotage, lisser la courbe de charge ou satisfaire une demande précise si nous considérons un signal de prix quelconque.

Dans ce mode de gestion, nous avons répertorié deux principaux problèmes d'optimisation. Tout d'abord dans la phase 3, il faut déterminer le plan de production des procédés de façon à maximiser leur gain tout en prenant en compte les contraintes qu'ils doivent respecter. Ensuite, dans la phase 4, le coordinateur doit agréger les plans de production de ses procédés et coordinateurs afin de maximiser le gain global en tenant compte des contraintes du réseau (limites minimale et maximale de quantité d'électricité échangée avec le réseau).

Ces problèmes d'optimisation sont à résoudre dans deux contextes : un contexte statique (résolution du problème la veille pour le lendemain) et un contexte dynamique (résolution du problème pendant la journée afin d'adapter le plan de production de la centrale d'agrégation pour répondre aux appels du mécanisme d'ajustement). Il est à noter que la cas dynamique correspond au même problème que le cas statique mais l'horizon de temps est plus court (reste de la journée). Dans les chapitres suivants, nous nous intéresserons donc aux deux cas.

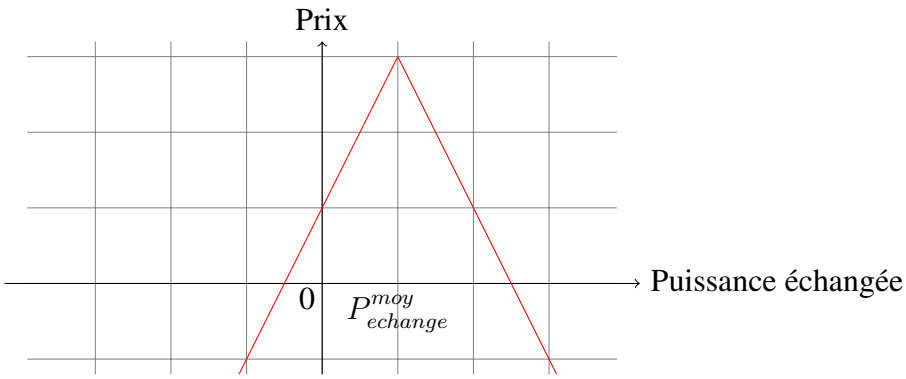

FIGURE 3.13 – Signal de prix induisant un lissage

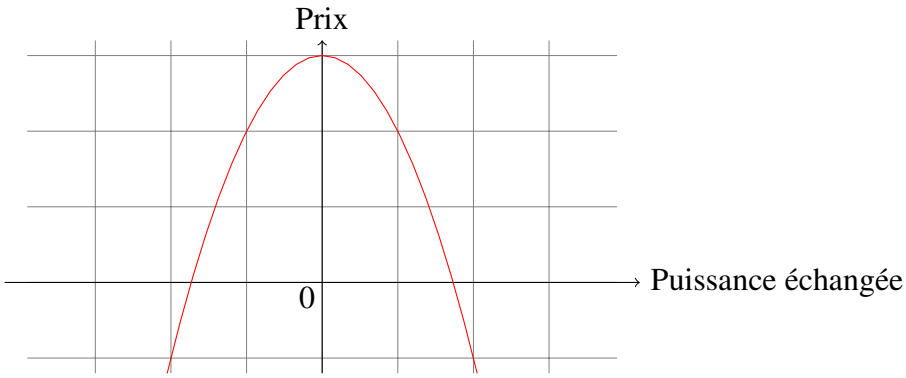

FIGURE 3.14 – Signal de prix induisant un îlotage à un pas de temps donné

4

Calcul du plan de production des procédés

Ce chapitre est consacré au problème d'optimisation consistant à calculer les puissances proposées par les procédés de déplacement d'une consommation (PDC), les procédés sans stockage (PSS) et procédés avec stockage (PAS) en fonction du signal de prix reçu. Ces puissances sont calculées de façon à maximiser le gain du procédé (différence entre le coût de vente et le coût de production). Il est à noter que nous optimiserons qu'un procédé à la fois. Nous rappellons que les puissances par défaut sont considérées connues pour tous les types de procédés (cf. Partie 2). Et que les puissances proposées des procédés fatals (PF) ne peuvent pas être optimisées car elles sont égales à leurs puissances par défaut.

Notre objectif est de développer des méthodes le plus génériques possible, pouvant ainsi résoudre le problème pour tout type de procédé. Ces méthodes devront être très rapides afin de pouvoir être utilisées dans un contexte temps réel.

4.1 État de l'art

Dans cette section, nous allons présenter un état de l'art des méthodes proposées pour calculer le plan de production le plus profitable pour un ou plusieurs procédés.

Les premiers travaux publiés dans la littérature sur ce sujet considèrent un unique procédé. Ainsi, la maximisation du gain d'une centrale hydraulique est étudiée depuis de nombreuses années : [Becker and Yeh, 1974] ont proposé une méthode de programmation linéaire, [Ölçer *et al.*, 1985] ont proposé une approche par programmation dynamique [Guangwen and Li, 1997], et [Wang *et al.*, 2006] ont travaillé sur un algorithme génétique.

Plusieurs méthodes ont également été proposées pour résoudre les problèmes liés aux recharges d'un procédé de stockage. Le problème d'optimisation d'une STEP a été résolu par un programmation linéaire en nombres entiers (PLNE) [Li *et al.*, 2011], une méthode basée sur des réseaux de neurones [Ruomei *et al.*, 1995], un algorithme de programmation dynamique [Wenqian, 2005] et une méthode de relaxation lagrangienne [Guan *et al.*, 1994]. [Soares *et al.*, 2011] ont comparé une méthode heuristique (particle swarm optimisation (PSO)) avec une méthode exacte (PLNE) pour résoudre le problème de la recharge intelligente de véhicules électriques. [Ahn *et al.*, 2011] ont utilisé un PLNE qui optimise la charge d'une batterie depuis le réseau. Puis plus tard, [Pang *et al.*, 2012] et [Jin *et al.*, 2013] ont proposé une approche similaire pour optimiser non seulement la charge mais aussi la décharge de la batterie.

En parallèle de ces travaux visant l'optimisation d'un seul procédé, des chercheurs se sont intéressés au calcul du plan de production le plus profitable pour un ensemble de procédés de différents types. Plusieurs articles concernent le couplage des moyens de production avec des moyens de stockage. Par

exemple, [Wille-Haussmann *et al.*, 2010] ont étudié l'incorporation d'un procédé de stockage sous forme thermique dans des moyens de cogénération. Les auteurs ont développé une approche PLNE pour résoudre ce problème. [Gu *et al.*, 2010] ont optimisé le plan de production d'un mix énergétique regroupant de la cogénération, des énergies renouvelables et des batteries en utilisant de la PLNE. Le problème d'une centrale thermique et d'une STEP a aussi été résolu par programmation dynamique [Cohen and Wan, 1985] et par une méthode de relaxation lagrangienne [Aoki *et al.*, 1987]. Plus récemment, plusieurs auteurs se sont intéressés à l'association d'ENR et de procédés de stockage : le plan de production d'une STEP couplée à des ENR a été résolu par de la programmation linéaire [Sanchez de la Nieta *et al.*, 2013]. Le couplage d'une turbine à gaz, d'ENR et d'une batterie de grande capacité a été optimisé au sein du projet [PEGASE, 2013] avec une approche de PLNE et par [Sobu and Wu, 2012b] avec une méthode de type PSO. [Kahrobaee *et al.*, 2013] ont résolu le problème de couplage d'éoliennes et de procédés de stockage avec un PSO. Le problème regroupant des éoliennes, des panneaux solaires et un procédé de stockage a été traité par [Sobu and Wu, 2012a] et [Wang and Yang, 2013] qui proposent respectivement un algorithme de programmation dynamique et un PSO. [Voice *et al.*, 2011] et [Wei *et al.*, 2010] ont étudié une agrégation de petits moyens de stockage en modélisant les interactions entre eux par un système multi-agents. Enfin, [Voice *et al.*, 2011] ont utilisé un mécanisme de pricing pour piloter le système alors que [Wei *et al.*, 2010] ont adopté une stratégie basée sur l'algorithme Zero-Intelligence-Plus utilisé en finance.

D'autres travaux proposent des méthodes pour optimiser un ensemble de procédés de consommation situés dans un même bâtiment : un PSO [Gudi *et al.*, 2010], un algorithme génétique [Fernandes *et al.*, 2011], [Zhao *et al.*, 2013] et [Kim and Lavrova, 2013], un PLNE [Oliveira *et al.*, 2011] et [Agnetis *et al.*, 2013], une recherche tabou [Tanaka *et al.*, 2012], un algorithme d'apprentissage [Xiong *et al.*, 2011], une heuristique [Goya *et al.*, 2011a], de la programmation dynamique [Georgievski *et al.*, 2012] et un PSO [Pedrasa *et al.*, 2010]. Pour essayer de réduire encore plus la facture, [Barbato *et al.*, 2011] ont ajouté un moyen de stockage au bâtiment et résolu le problème par un PLNE. [Tushar *et al.*, 2014] ont ajouté des véhicules électriques et résolvent aussi le problème par un PLNE. Le problème de calcul du plan de production optimal de panneaux photovoltaïques couplés à un bâtiment a été résolu par [Bozchalui *et al.*, 2012] par un PLNE, et [Zhao and Wu, 2014] ont proposé également un modèle de PLNE pour optimiser le problème associant des éoliennes et un bâtiment. La PLNE a également été utilisée par [Chen *et al.*, 2013] pour résoudre le problème d'un ensemble de data centers. Pour terminer, [Oshiro *et al.*, 2012] s'est intéressé au problème regroupant des demandes contrôlables et un moyen de stockage, problème que les auteurs ont résolu par une heuristique.

Certains chercheurs se sont intéressés à d'autres fonctions objectifs que la maximisation du gain. La plus étudiée est la minimisation du coût de production. Dans un problème de minimisation du coût de production, une puissance donnée doit être produite à chaque pas de temps, et l'on cherche à minimiser le coût de cette production. [Chen *et al.*, 2011] ont proposé un PLNE pour le problème regroupant plusieurs sources de production (intermittentes et conventionnelles). Un algorithme de programmation dynamique a été proposé pour un problème de stockage pur [Koutsopoulos *et al.*, 2011]. L'agrégation d'un ensemble de procédés composé de moyens de production, de moyens de stockage et de moyens de pilotage de la consommation a été résolue par une recherche tabou suivie d'un algorithme génétique [Asato *et al.*, 2010], un PSO et une approche par programmation non linéaire en nombres entiers [Faria *et al.*, 2011]. Un des inconvénients de toutes ces techniques d'optimisation est la nécessité d'avoir un ordinateur qui soit connecté avec l'ensemble des moyens de production. La récupération de toutes les informations nécessaires à l'optimisation auprès de tous les procédés implique une perte de temps. [Zhang and Chow, 2011] et [Faria *et al.*, 2013] ont donc imaginé une gestion décentralisée. Les auteurs obtiennent alors le meilleur coût pour l'ensemble des procédés grâce à un algorithme cost concensus [Zhang and Chow, 2011] et un PSO par [Faria *et al.*, 2013].

Quelques chercheurs ont cherché à minimiser, en plus du coût de production, les émissions de dioxyde de carbone et autres gaz à effet de serre lors de la production d'électricité. [Al-Awami *et al.*, 2009] ont utilisé un PSO pour résoudre le problème couplant plusieurs sources de production (intermittentes et conventionnelles). [Saber and Venayagamoorthy, 2012] se sont intéressés au même problème en y ajoutant des batteries de voitures électriques. Ils ont proposé un PSO pour le résoudre. [Hermans *et al.*, 2011] ont uniquement

cherché à minimiser les émissions de dioxyde de carbone de plusieurs sources de production (carbonée et décarbonée) et la recharge d'une flotte de véhicules électriques avec un PLNE.

D'autres recherches traitent des fonctions objectifs liées à la sécurisation du réseau :

- la minimisation des variations de la fréquence est abordée par [Han *et al.*, 2010] qui proposent un algorithme génétique, [Lu *et al.*, 2011] qui résolvent un problème de flot de puissance maximale, et [Vachirasricirikul and Ngamroo, 2014] qui ont développé un PSO ;
- la minimisation des variations de la tension est traitée par [Alonso *et al.*, 2012] (algorithme génétique), [Viehweider *et al.*, 2010] (PLNE), [Laaksonen *et al.*, 2011] (problème de flot de puissance maximale), et [Goya *et al.*, 2011b] (heuristique) ;
- la minimisation des pertes a été étudiée par [Xiwen *et al.*, 2010] qui ont proposé un algorithme génétique, [Qudaih and Mitani, 2011] qui propose une méthode basée sur les réseaux de neurones, [Ochoa and Harrison, 2011] qui modélisent le problème comme un problème de flot de puissance maximale, [Possemato *et al.*, 2013] qui ont développé deux algorithmes évolutionnistes (PSO et algorithme génétique), et enfin [Inoue *et al.*, 2014] qui a proposé une heuristique.

Enfin [Wang *et al.*, 2011] ont utilisé une PSO afin de minimiser l'inconfort d'un consommateur lorsqu'il utilise une centrale d'agrégation.

Chacun de ces travaux propose une modélisation spécifique à chaque procédé géré. Une approche générique pouvant modéliser à la fois des procédés sans stockage (PSS), des PDC, des procédés avec stockage (PAS) et PF a été proposée par [Delon and Normand, 2011] et par [Heussen *et al.*, 2010]. Ces approches sont basées sur la programmation linéaire en nombres entiers. Cependant, la modélisation proposée par Heussen *et al.* ne permet pas de modéliser un procédé qui injecte de l'énergie directement dans le réseau sans être préalablement stockée, comme dans le système PAC (cf. Figure 2.18). En ce qui concerne le modèle de Delon and Normand, il n'est pas entièrement satisfaisant car il approxime la contrainte de stock.

Dans la chapitre suivant, nous allons détailler ces deux modélisations et exposer notre solution afin de modéliser les quatre groupes de procédés.

4.2 Modélisation du procédé

Afin de modéliser tous les groupes de procédés, nous nous sommes principalement intéressés aux modèles de Delon and Normand [Delon and Normand, 2011] et de Heussen *et al.* [Heussen *et al.*, 2010].

La modélisation de Delon and Normand est composée d'une partie processus interne et d'une partie interface réseau (cf. Figure 4.1) qui échangent directement de l'énergie. Cette modélisation n'est pas satisfaisante car elle approxime la contrainte de stock. En effet, le stockage est pris en compte grâce à une contrainte de durée maximale d'effacement qui représente le temps maximal de décharge et une contrainte de durée minimale entre un arrêt et le prochain départ qui représente le temps minimal de charge. Ainsi, la valeur du stock n'est pas connue à tout instant.

La proposition de Heussen *et al.*, quant à elle, consiste à modéliser les procédés par des "power nodes". Un power node est un stockage sous une forme d'énergie (chimique, chaleur, etc..) placé entre un processus, qui convertit cette forme d'énergie en énergie électrique afin de l'échanger avec le réseau, et un processus de production ou de consommation interne de cette forme d'énergie. Le figure 4.2 représente cette modélisation. Le power node a une capacité de stockage C qui peut être nulle. Son stockage n'est pas parfait, il subit des pertes ($\mu \geq 0$). Le power node échange avec le réseau ("Grid side") via un processus de conversion. Il peut soutirer au réseau une quantité $uload$ mais il ne reçoit qu'une partie de cette énergie après la conversion ($\eta load * uload$). S'il veut injecter une puissance $ugen$ sur le réseau, il doit injecter $\eta gen^{-1} * ugen > ugen$ dans le processus de conversion à cause des pertes liées au processus de conversion. Le power node échange aussi avec le processus de production/consommation interne ("Demand/Supply side"). Il reçoit de l'énergie soit via des apports pilotables ("provided energy" $\epsilon > 0$) soit via des apports incontrôlables ("unserved load"

$w < 0$). Il injecte de l'énergie soit via des consommations pilotables ("demande energy" $\epsilon < 0$) soit via des demandes incontrôlables ("spilled energy" $w > 0$). Cette modélisation a l'avantage de couvrir la majorité des procédés. Par exemple dans le cas du système PAC (cf. Figure 2.18), le power node est le ballon d'eau chaude. Une pompe à chaleur permet de convertir l'électricité fournie par le réseau en eau chaude (*grid side*), et des radiateurs soutirent l'énergie fournie par le ballon d'eau chaude (*demand side*).

Cette modélisation est centrée sur le stockage (les processus de conversion vers le réseau et d'échange interne sont externalisés). Elle sous-entend donc que tout transite par le stockage. Or pour certains procédés l'énergie est transférée directement du convertisseur à l'échangeur interne ou vice versa sans passer par le stockage. C'est la cas, par exemple du système PAC : la pompe à chaleur peut envoyer de l'eau chaude dans les radiateurs sans passer par le ballon d'eau chaude.

Nous avons donc modifié la modélisation proposée par Heussen *et al.* pour y intégrer le processus d'échange avec le réseau et le processus de production/consommation interne.

Notre modélisation est composée de (cf. Figure 4.3) :

- une partie processus internes : elle représente la partie du procédé qui produit ou consomme une puissance p_{ip} (*power internal process*) : les radiateurs dans le cas du système PAC ;
- un stockage de capacité C (qui peut être nulle) avec des pertes l. Dans le cas d'un système PAC, ce stockage représente le ballon d'eau chaude. Il échange avec la partie processus internes une puissance δ_{ip}. Par convention δ_{ip} est positif si la partie processus internes fournie de l'énergie au stockage et négatif si elle reçoit de l'énergie ;
- un partie interface réseau qui permet l'échange d'une puissance p_{gi} (*power grid interface*) vers ou depuis le réseau et la conversion d'une forme d'énergie en énergie électrique (la pompe à chaleur dans le cas du système PAC). Elle échange avec le stockage une puissance δ_{gi} et avec la partie processus internes une puissance δ_{di} (*direct*). Par convention δ_{gi} est positif si le stockage fournit de l'énergie à l'interface réseau et négatif s'il reçoit de l'énergie. Par convention δ_{di} est positif si la partie processus internes fournit de l'énergie à l'interface réseau et négatif si elle reçoit de l'énergie.

Il est à noter que, contrairement à la modélisation proposée par Heussen *et al.*, les pertes liées au processus de conversion n'apparaissent pas dans notre modélisation. En effet, quelle que soit la quantité d'énergie produite ou consommée, les pertes de conversion représentent un pourcentage connu à l'avance et peuvent donc être prises en compte a priori.

Avec cette modélisation générique, nous pouvons modéliser les six types de procédés. En effet :

1. Les procédés de production pilotable ont un stockage de capacité nulle, et leurs δ_{di} seront positifs ou nuls (cf. Figure 4.4) ;
2. Les procédés de consommation pilotable ont un stockage de capacité nulle, et leurs δ_{di} seront négatifs ou nuls (cf. Figure 4.5) ;
3. Les procédés de déplacement de la production fatale ont une puissance interne p_{ip} positive ou nulle connue et leurs δ_{di}, δ_{ip} et δ_{gi} seront positifs ou nuls (cf. Figure 4.6) ;
4. Les procédés de déplacement de la consommation fatale ont une puissance interne p_{ip} négative ou nulle connue et leurs δ_{di}, δ_{ip} et δ_{gi} seront négatifs ou nuls (cf. Figure 4.7) ;
5. Les procédés de déplacement d'une consommation ont un stockage de capacité nulle, et leurs δ_{di} seront négatifs ou nuls (cf. Figure 4.8) ;
6. Les procédés de stockage ont une puissance interne p_{ip} nulle et les δ_{ip} seront positifs, quand le procédé se déchargera, ou négatifs, quand le procédé se chargera (cf Figure 4.9).

FIGURE 4.1 – Modélisation de Delon et Normand

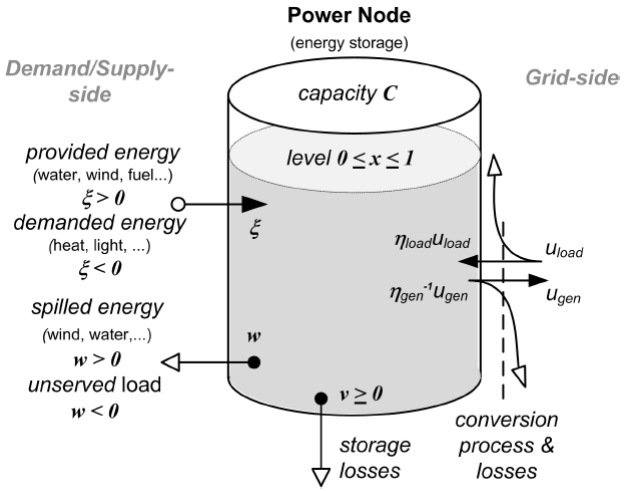

FIGURE 4.2 – Modélisation de Heussen *et al*

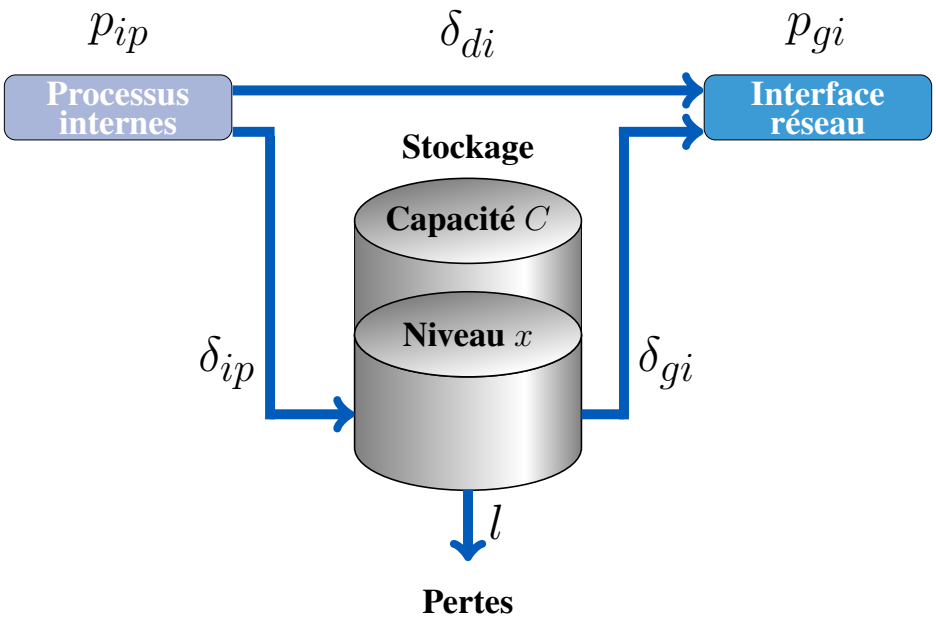

FIGURE 4.3 – Modélisation générique

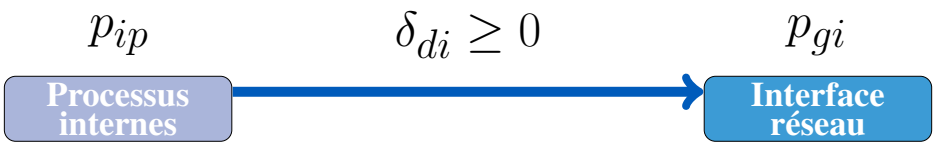

FIGURE 4.4 – Modélisation d'un procédé de production pilotable

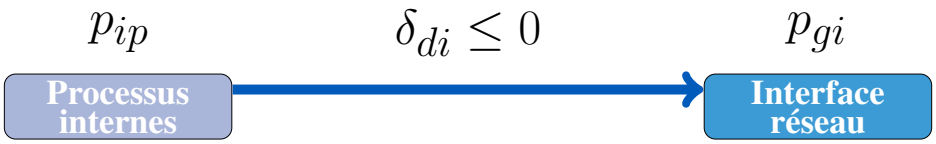

FIGURE 4.5 – Modélisation d'un procédé de consommation pilotable

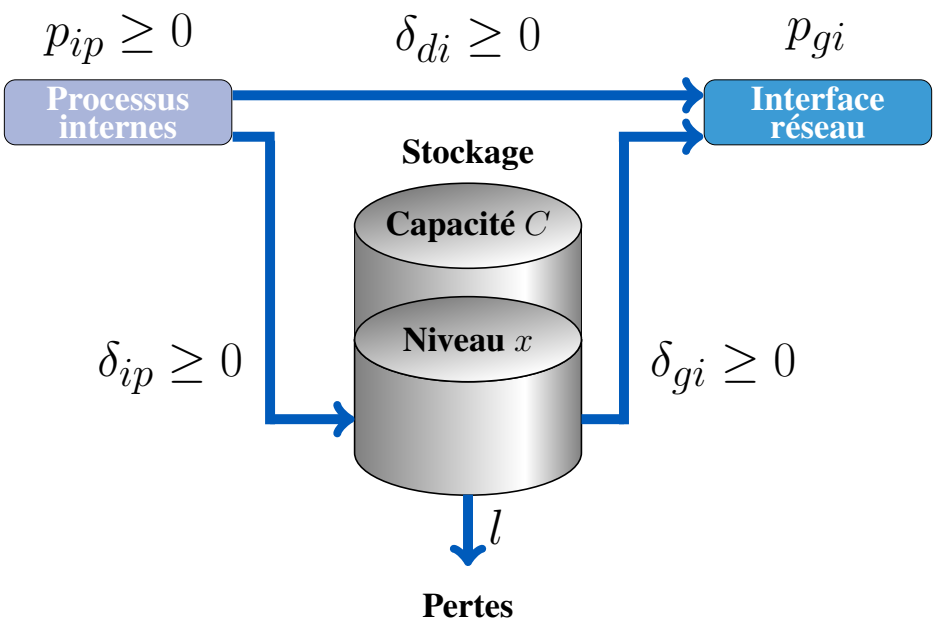

FIGURE 4.6 – Modélisation d'un procédé de déplacement de la production fatale

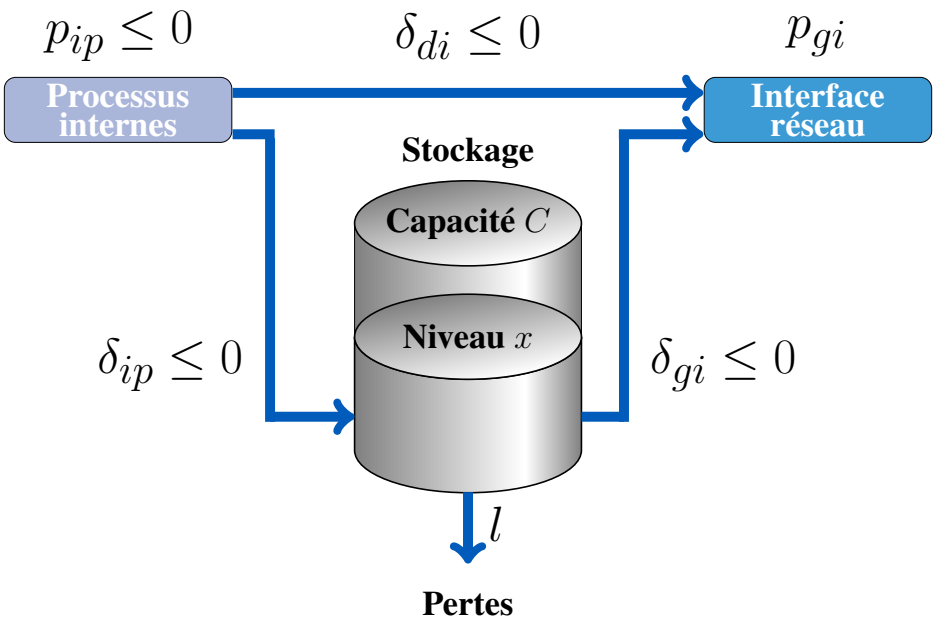

FIGURE 4.7 – Modélisation d'un procédé de déplacement de la consommation fatale

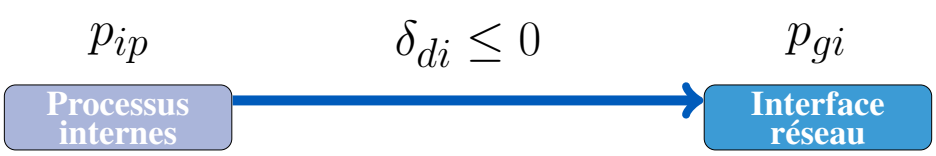

FIGURE 4.8 – Modélisation d'un procédé de déplacement d'une consommation

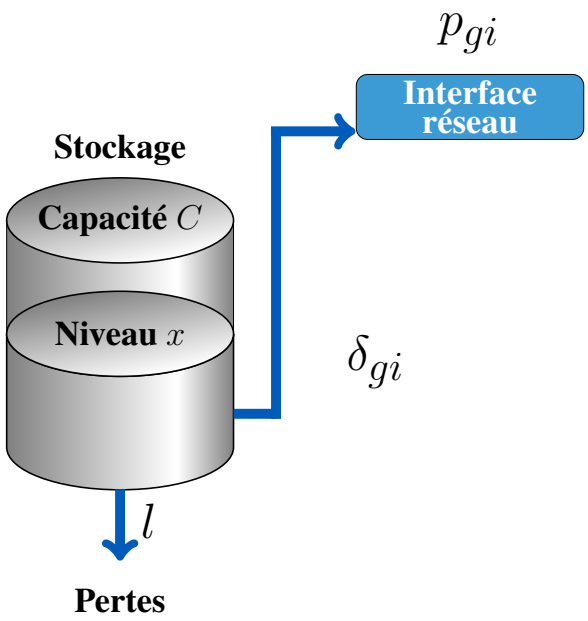

FIGURE 4.9 – Modélisation d'un procédé de stockage

4.3 Description détaillée du problème

Actuellement, les procédés de production doivent indiquer, pour le cas statique du jour pour le lendemain, leur plan de production pour une période d'une journée avec un pas de temps de 10 minutes. Pour le cas dynamique, ils doivent l'indiquer pour le reste de la journée avec le même pas de temps. Dans ce problème, nous considérons donc un procédé sur l'horizon de temps $\mathcal{T}$ le plus grand entre les deux cas, soit de 24 heures décomposé en $T = 144$ pas de temps de 10 minutes.

Les caractéristiques communes à tous les types de procédés sont décrites ci-après.

À chaque pas de temps t, chaque procédé a une consommation par défaut $P_{def}(t)$. Nous noterons $\delta_p(t)$ la différence au pas de temps t entre la puissance réellement consommée/produite $p_{gi}(t)$ et la puissance consommée/produite par défaut $P_{def}(t)$. Cette différence de puissances est strictement négative si le procédé consomme plus ou produit moins d'électricité que par défaut, elle est strictement positive si le procédé consomme moins ou produit plus d'électricité que par défaut, et elle est nulle si le système produit ou consomme $P_{def}(t)$. Il est à noter que tous les procédés, qu'ils soient consommateur, producteur ou de stockage peuvent échanger plus avec le réseau que par défaut en diminuant leur consommation ou en augmentant leur production. Nous dirons alors qu'ils injectent de l'énergie dans le réseau. Tous peuvent aussi échanger moins avec le réseau que par défaut en augmentant leur consommation ou en diminuant leur production. Nous dirons alors qu'ils soutirent de l'énergie depuis le réseau. Dans la suite de ce chapitre, nous ne parlerons plus de production ou de consommation mais d'injection ou de soutirage. Une valeur positive de $\delta_p(t)$ correspondra donc à une injection au pas de temps t, et une valeur négative de $\delta_p(t)$ à un soutirage.

Un coût de maintien C_{on} est associé à chaque procédé. Il correspond au coût de soutirage ou d'injection du procédé sur un pas de temps. Ce coût représente par exemple le coût lié à l'utilisation d'un stock qui n'est pas utilisé par défaut. À chaque procédé est également associé un coût marginal C_{marg} (coût des matières premières pour les systèmes de production ou coût de désagrément pour les systèmes de gestion de la demande). De plus, chaque procédé reçoit pour chaque pas de temps t un prix $P_r(t)$.

Nous chercherons, pour chaque procédé, à maximiser son bénéfice, c'est à dire la différence entre son gain $\sum_t P_r(t) * (\delta_p(t) + P_{def}(t))$ et le coût de production égal à la somme du coût marginal ($\sum_t (C_{marg} * \delta_p(t))$ et du coût de maintien (C_{on} multiplié par le nombre de pas de temps pendant lequel le procédé injecte/soutire). Comme nous le verrons par la suite, un coût de démarrage pourra aussi devoir être pris en compte.

Les procédés ont d'autres caractéristiques qui diffèrent d'un type de procédé à l'autre, ainsi que diverses contraintes techniques et de confort. Elles sont décrites dans les sections qui suivent.

4.3.1 Caractéristiques du problème des PDC

Ces procédés ont une consommation à positionner sur l'horizon de temps (par exemple la consommation d'un cycle de lavage d'une machine à laver). La tâche à satisfaire sur la journée, notée Θ, est composée de :

- $L_{engthTask}$, la durée de la tâche ;
- S_{power}, un tableau de longueur $L_{engthTask}$ qui définit les puissances consommées pendant la durée de la tâche (découpée en pas de temps).

La puissance par défaut de ce procédé correspond à la puissance consommée par la tâche si elle est démarrée à l'heure initialement prévue. L'objectif est de déterminer une meilleure date de démarrage de cette tâche (et donc les puissances injectées ou soutirées $\delta_p(t)$) de façon à maximiser le bénéfice du procédé compte tenu du prix qu'il aura reçu.

Un PDC sera soumis aux deux contraintes techniques listées ci-dessous :

a) Indisponibilité : un système n'est pas toujours disponible, pour raison de maintenance par exemple. Soit $I_{ndispo}(t)$, l'indisponibilité du procédé au pas de temps t. Si $I_{ndispo}(t)$ est nul alors le procédé est disponible, sinon $I_{ndispo}(t)$ est égale à 1 et le procédé est indisponible.

b) Intervalle de départ : la tâche à exécuter doit démarrer entre une période au plus tôt $T_{MinStart}$ et une période au plus tard $T_{MaxStart}$.

4.3.2 Caractéristiques du problème des PSS

Dans ce problème, il faut déterminer les injections et soutirages maximisant le bénéfice du procédé et respectant les contraintes techniques suivantes :

a) Indisponibilité (cf. contrainte décrite dans le problème précédent).

c) Domaine de puissance : à chaque instant t, $p_{gi}(t)$ est comprise entre une puissance minimale P_{min}^{conso} et une puissance maximale P_{max}^{prod}. Ainsi par exemple, pour un procédé de consommation $P_{min}^{conso} < 0$ et $P_{max}^{prod} = 0$. À chaque instant t, $\delta_p(t)$ doit donc prendre des valeurs entre une borne inférieure notée $P_{min}(t)$ (=$P_{min}^{conso} - P_{def}(t)$) et une borne supérieure notée $P_{max}(t)$ (=$P_{max}^{prod} - P_{def}(t)$).

D'autres contraintes techniques peuvent également devoir être respectées :

d) Coût de démarrage : un coût de démarrage peut être associé à certains procédés. Il sera noté C_{start}. Dans ce cas, le coût de production du procédé est égal à la somme du coût marginal, du coût de maintien et du nombre d'injections/soutirages du procédé multiplié par C_{start}.

e) Discrétisation : la puissance d'un procédé peut être discrète. Par exemple, dans un système de gestion de l'éclairage public qui contrôle $400\,W$ de DEL, les DEL sont contrôlées par bloc de $100\,W$. Ainsi, la puissance peut être égale à :

- $400\,W$ quand toutes les DEL sont allumées ;
- $300\,W$ quand un quart des DEL sont éteintes ;
- $200\,W$ quand la moitié des DEL sont éteintes ;
- $100\,W$ quand trois-quart des DEL sont éteintes ;
- $0\,W$ quand toutes les DEL sont éteintes.

Nous noterons $D_{iscr}(t)$ l'ensemble des puissances que le procédé peut atteindre au pas de temps t.

f) Gradient maximal entre deux puissances fournies consécutives : pour certains procédés, l'augmentation ou la diminution de la puissance d'un pas de temps à l'autre est limitée. Nous noterons G_{max} le gradient maximal entre deux puissances consécutives.

Un PSS peut aussi devoir satisfaire une ou plusieurs des contraintes de confort listées ci-dessous :

g) Durée maximale d'appel à une injection et à un soutirage : par exemple un radiateur ne doit pas être éteint trop longtemps afin d'éviter trop d'inconfort pour l'utilisateur. Ainsi, suivant le type de procédé, nous pouvons rencontrer les contraintes suivantes :

- le procédé ne peut injecter dans le réseau pendant plus de $T_{maxOn}^{p^+}$ périodes ;
- le procédé ne peut soutirer depuis le réseau pendant plus de $T_{maxOn}^{p^-}$ périodes.

h) Durée minimale d'appel : par exemple une pompe à chaleur ne doit pas être allumée puis éteinte directement afin de limiter des casses matérielles. Ainsi suivant le type de procédé, nous pouvons rencontrer les contraintes suivantes :

- le procédé doit injecter dans le réseau pendant au moins de $T_{minOn}^{p^+}$ périodes ;

- le procédé doit soutirer depuis le réseau pendant au moins de $T^{p^-}_{minOn}$ périodes.

i) Durée minimale entre un arrêt et le prochain départ : par exemple cette contrainte interdit si nous avons éteint un radiateur pendant plusieurs pas de temps le matin de l'éteindre à nouveau avant le soir. Ceci permet de répartir les périodes d'inconfort sur l'horizon de temps. Ainsi un procédé peut devoir attendre après un arrêt au moins :

- $T^{p^+}_{minOff}$ périodes avant de pouvoir injecter de nouveau dans le réseau ;
- $T^{p^-}_{minOff}$ périodes avant de pouvoir soutirer de nouveau depuis le réseau.

j) Nombre maximal d'appels sur l'horizon $\mathcal{T}$: cette contrainte permet par exemple de limiter le nombre total d'inconforts liées à l'extinction d'un radiateur. Ainsi, suivant le type de procédé :

- le nombre d'injections du procédé dans le réseau peut être limité à $M^{p^+}_{axCall}$ fois ;
- le nombre de soutirages du procédé depuis le réseau peut être limité à $M^{p^-}_{axCall}$ fois.

k) Quantité maximale d'énergie sur l'horizon $\mathcal{T}$: cette contrainte permet par exemple de limiter l'inconfort global. Ainsi, suivant le type de procédé :

- l'énergie injectée dans le réseau sur l'horizon $\mathcal{T}$ peut être limitée à une quantité maximale $M^{p^+}_{axVol}$;
- l'énergie soutirée depuis le réseau sur l'horizon $\mathcal{T}$ peut être limitée à une quantité maximale $M^{p^-}_{axVol}$.

La figure 4.10 représente par exemple la variation de puissance pour un PSS qui a une durée minimale d'injection ($T^{p^+}_{minOn}$) de deux, une durée maximale d'injection ($T^{p^+}_{maxOn}$) de six, une durée minimale entre deux injections ($T^{p^+}_{minOff}$) de trois et un nombre maximal de soutirages ($M^{p^-}_{axCall}$) de deux. Nous voyons que les durées, respectivement de trois et de quatre pas de temps, des deux injections sont supérieures ou égales à $T^{p^+}_{minOn}$ (2) et inférieures ou égales à $T^{p^+}_{maxOn}$ (6). De plus la durée de trois pas de temps entre les deux injections est supérieure ou égale à $T^{p^+}_{minOff}$ (3). Et enfin le nombre de soutirages pendant $\mathcal{T}$ est de deux qui est inférieur ou égal à $M^{p^-}_{axVol}$ (2).

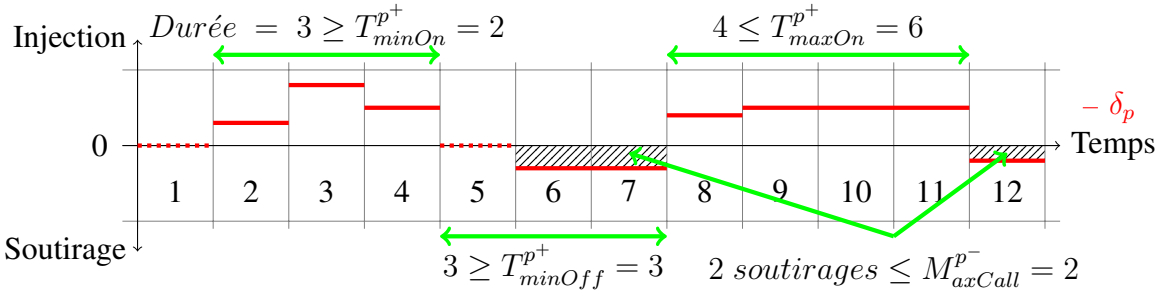

FIGURE 4.10 – Exemple de plan de production pour les procédés sans stockage

4.3.3 Caractéristiques du problème des PAS

Dans le cas des PAS, la puissance consommée/produite $p_{ip}(t)$ est connue et notée $P_{ip}(t)$. Un plan de charge et décharge du stockage est initialement prévu, et permet de déduire une puissance par défaut pour le procédé : si le procédé stocke de l'énergie à un pas de temps t, la puissance par défaut à ce pas de temps

est donc égale à $p_{ip}(t)$ moins la quantité stockée ; si le procédé déstocke au pas de temps t, la puissance par défaut est égale à $p_{ip}(t)$ plus la quantité destockée. L'objectif est alors de déterminer un nouveau plan de charge et décharge du stockage maximisant le bénéfice du procédé compte tenu du prix reçu.

Les PAS sont, comme les PSS, soumis aux contraintes techniques obligatoires a) et c).

Il faudra de plus respecter les contraintes techniques suivantes qui concernent le stock :

l) Domaine du stock : à chaque instant t, le niveau de stock $x(t)$ doit être compris entre 0 et C.

m) Pertes de stockage : nous allons supposer que les pertes du stockage sont linéaires par rapport au niveau de stock. À chaque instant t, nous supposerons donc que $l(t) = P_{Loss} * x(t) + C_{Loss}$ ou C_{Loss} et P_{Loss} sont des constantes.

n) Conservation de l'énergie : afin de respecter la conservation de l'énergie dans le procédé à chaque instant t, la quantité $p_{ip}(t)$ échangée depuis ou vers le processus interne moins la quantité $\delta_{ip}(t)$ échangée entre le processus interne et le stockage doit être égale à la quantité d'énergie $p_{gi}(t)$ injectée ou soutirée au réseau moins la quantité d'énergie $\delta_{gi}(t)$ échangée entre l'interface réseau et le stockage.

o) Variation du stock : à chaque pas de temps t, la variation du stock du procédé est égale à la différence entre la puissance interne du procédé et la puissance demandée au réseau moins les pertes du stockage.

Un PAS pourra de plus, comme les PSS, devoir respecter les contraintes techniques optionnelles d), e), f), g), h), i), j) et k).

Un PAS sera également soumis aux mêmes contraintes de confort qu'un PSS (durée maximale d'une injection et d'un soutirage, durée minimale d'une injection ou d'un soutirage, durée minimale entre un appel et le prochain départ d'une injection et d'un soutirage, nombre maximal d'injections et de soutirages sur l'horizon, et volume maximal d'injection et de soutirage), mais aussi à des contraintes similaires concernant le stockage et le destockage (nous les avons donc identifiées par la même lettre) :

g) Durée maximale d'appel :

- le procédé ne peut être en charge pendant plus de $T^{x^+}_{maxOn}$ périodes ;
- le procédé ne peut être en décharge pendant plus de $T^{x^-}_{maxOn}$ périodes.

h) Durée minimale d'appel :

- le procédé doit être en charge pendant au moins de $T^{x^+}_{minOn}$ périodes ;
- le procédé doit être en décharge pendant au moins de $T^{x^-}_{minOn}$ périodes.

i) Durée minimale entre un arrêt et le prochain départ :

- $T^{x^+}_{minOff}$ périodes pour charger de nouveau ;
- $T^{x^-}_{minOff}$ périodes pour décharger de nouveau.

j) Nombre maximal d'appels sur l'horizon $\mathcal{T}$:

- le nombre de charges peut être limité à $M^{x^+}_{axCall}$ fois ;
- le nombre de décharges peut être limité à $M^{x^-}_{axCall}$ fois.

Ces procédés pourront aussi devoir respecter les contraintes supplémentaires suivantes :

p) Niveau de stock borné à chaque pas de temps t : un utilisateur peut par exemple vouloir faire varier la plage de températures acceptables de son habitation. Ceci revient à borner le stock entre un niveau minimal $X_{min}(t)$ et un niveau maximal $X_{max}(t)$.

q) Obligation d'attendre un niveau de stock minimal avant de pouvoir déstocker : cette contrainte modélise par exemple le fait qu'une pièce doit se réchauffer suffisamment (atteindre X_{min}^{unload}) avant d'éteindre de nouveau le chauffage.

Dans la section suivante, nous allons voir comment nous avons généré des instances afin de tester nos méthodes.

4.4 Instances

Afin de tester nos méthodes, nous avons généré un ensemble d'instances basées sur des procédés réels.

Pour tous les types de procédés, nous avons généré les caractéristiques suivantes :

- l'horizon de temps $\mathcal{T}$ est fixé à 24 heures avec un pas temporel de 10 minutes (i.e. $T = 144$ pas de temps) ;
- le signal de prix est choisi aléatoirement dans l'historique des prix du marché spot (prix de gros de l'électricité pour l'échange infra-journalier pour plusieurs pays européens) de l'année 2011. Comme il n'y a qu'un prix spot EPEX par heure, et que le pas de temps est de 10 minutes, nous avons décidé de les linéariser afin d'avoir une évolution moins brutale du signal de prix ;
- le nombre de plages d'indisponibilité est choisi aléatoirement entre 0 et 6. Leur durée est générée aléatoirement entre 1 et T pas de temps. Leurs emplacements dans $\mathcal{T}$ sont choisis aléatoirement. Cependant, deux indisponibilités doivent être au moins séparées d'une période.

Concernant les PDC, la consommation à positionner est choisie aléatoirement dans un historique de courbes de charge de systèmes électroménagers. De plus $T_{MinStart}$ est choisi aléatoirement entre 0 et $T - L_{engthTask}$, et $T_{MaxStart}$ entre $T_{MinStart}$ et T.

Pour les PSS, les valeurs des paramètres $P_{max}(t)$, $P_{min}(t)$, $\forall\alpha \in \{p^+; p^-\}$, T^{α}_{minOn}, T^{α}_{maxOn}, T^{α}_{minOff}, M^{α}_{axVol}, M^{α}_{axCall} et G_{max} sont choisies aléatoirement entre une valeur minimale et une valeur maximale tel que décrit dans le tableau 4.1.

Pour les PAS, les valeurs des paramètres communs avec les PSS (*i.e.* $P_{max}(t)$, $P_{min}(t)$, $\forall\alpha \in \{p^+; p^-\}$, T^{α}_{minOn}, T^{α}_{maxOn}, T^{α}_{minOff}, M^{α}_{axVol}, M^{α}_{axCall} et G_{max}) sont générées comme pour les PSS. Les autres paramètres ($P_{ip}(t)$, $\forall\alpha \in \{x^+; x^-\}$, T^{α}_{minOn}, T^{α}_{maxOn}, T^{α}_{minOff}, M^{α}_{axCall}, C, X_{min}, X_{max}, X_{min}^{unload}, C_{Loss} et P_{Loss}) sont choisis aléatoirement entre une valeur minimale et une valeur maximale tel que décrit dans le tableau 4.2.

Pour exprimer les paramètres de type T^{α}_{minOn}, T^{α}_{maxOn} et T^{α}_{minOff} en nombre de pas de temps, nous utiliserons la partie entière supérieure de sa division par la durée du pas de temps (10 minutes).

D'autre part, afin de générer des procédés PSS et PAS avec une puissance continue et d'autres avec une puissance discrète, la contrainte de discrétisation est activée avec un probabilité de 0,5. Le nombre de valeurs discrètes est choisi aléatoirement entre 1 et 5. Et les valeurs sont choisies aléatoirement entre $\max_{t\in\mathcal{T}} P_{min}(t)$ et $\min_{t\in\mathcal{T}} P_{max}(t)$.

Pour terminer, pour les PSS et les PAS, le coût de départ C_{start} est choisi aléatoirement entre 0 et $\max(T^{p^+}_{maxOn}, T^{p^-}_{maxOn}) * \max_{t\in\mathcal{T}} P_{max}(t) * \max_{t\in\mathcal{T}} |PS(t)|$. Si le coût de départ est plus grand que cette valeur, le procédé ne sera jamais rentable. Si le procédé n'est pas soumis à la contrainte g) (donc si aucune valeur n'a été générée pour $T^{p^+}_{maxOn}$ et $T^{p^-}_{maxOn}$, la borne supérieure de l'intervalle du coût de départ est $T * \max_{t\in\mathcal{T}} P_{max}(t) * \max_{t\in\mathcal{T}} |PS(t)|$

Pour les mêmes raisons, le coût de maintien C_{on} est choisi aléatoirement entre 0 et $\max_{t\in\mathcal{T}} P_{max}(t) * \max_{t\in\mathcal{T}} |PS(t)|$ et le coût marginal C_{marg} entre 0 et $\max_{t\in\mathcal{T}} |PS(t)|$.

TABLE 4.1 – Données pour la génération des instances de procédés sans stockage

Donnée	Minimum	Maximum
$\forall t \in \mathcal{T}$, $P_{max}(t)$	0 W	5000 W
$\forall t \in \mathcal{T}$, $P_{min}(t)$	-5000 W	0 W
$\forall \alpha \in \{p^+; p^-\}$, T^{α}_{minOn}, T^{α}_{maxOn}, T^{α}_{minOff}	1 minute	24 heures
$M^{p^+}_{axVol}$	1 W.h	$24 * \max_{t\in\mathcal{T}} P_{max}(t)$ W.h
$M^{p^-}_{axVol}$	1 W.h	$24 * \max_{t\in\mathcal{T}} P_{min}(t)$ W.h
$\forall \alpha \in \{p^+; p^-\}$, M^{α}_{axCall}	1 appel	12 appels
G_{max}	1 W	$\max_{t\in\mathcal{T}} P_{max}(t) - P_{min}(t)$ W

TABLE 4.2 – Données pour la génération des instances de procédés avec stockage

Donnée	Minimum	Maximum
$\forall t \in \mathcal{T}$, $P_{ip}(t)$	-5000 W	5000 W
$\forall \alpha \in \{x^+; x^-\}$ T^{α}_{minOn}, T^{α}_{maxOn}, T^{α}_{minOff}	1 minute	24 heures
$\forall \alpha \in \{x^+; x^-\}$ M^{α}_{axCall}	1 appel	12 appels
C	0 W.h	$24 * \max_{t\in\mathcal{T}} P_{max}(t) - P_{min}(t)$ W.h
X_{min}, X_{max}, X^{unload}_{min}	0 W.h	C W.h
C_{Loss}	0 W.h	C W.h
P_{Loss}	0	1

4.5 Complexité

Nous allons dans ce chapitre étudier la complexité des problèmes des différents types de procédé. Il est à noter que la taille des problèmes considérés ici est le nombre de pas de temps. Nous allons dans un premier temps nous intéresser aux procédés de type PDC. Nous étudierons ensuite les procédés PSS puis nous terminerons par les PAS.

4.5.1 PDC

Nous rappelons dans le tableau ci-dessous les différentes contraintes auxquelles sont soumis les PDC.

Nom	$a)$ I_{ndispo}	b) Intervalle de départ
PDC	X	X

Le problème de PDC revient à déterminer la date de démarrage de la tâche à réaliser en respectant les indisponibilités du procédé et de façon à maximiser son gain.

Proposition 1
PDC est polynomial.

Preuve : La solution optimale de ce problème est obtenue en appliquant l'algorithme 1. Le principe de cet algorithme consiste à calculer pour tous les pas de démarrage possibles le coût de la tâche à effectuer, et à conserver le meilleur pas de temps de départ. Cet algorithme est en $O((T - L_{engthTask}) * L_{engthTask})$. $L_{engthTask}$ étant compris entre 1 et T, la complexité est donc en $O(T^2)$.

Conclusion : PDC est donc polynomial.

Algorithm 1 Algorithme optimal pour les procédés de déplacement d'une consommation

1: Données :
- $L_{engthTask}$: durée de la tâche ;
- S_{power} : tableau de longueur $L_{engthTask}$ qui définit la puissance consommée à chaque pas de temps pendant l'exécution de la tâche ;
- $T_{MinStart}$: pas de temps au plus tôt de départ de la tâche ;
- $T_{MaxStart}$: pas de temps au plus tard de départ de la tâche ;
- T : nombre de périodes de l'horizon de temps ;
- P_r : tableau de signal de prix de longueur T ;
- I_{ndispo} : tableau indiquant pour chaque pas de temps si le procédé est indisponible.

2: **procedure** ALGOPDC($L_{engthTask}, S_{power}, T_{MinStart}, T_{MaxStart}, T, P_r, I_{ndispo}$)
3: $bestStart = -1$ ▷ Le meilleur pas de temps de départ.
4: $bestGain = 0$ ▷ Le meilleur gain.
5: $t_{dep} = T_{MinStart}$ ▷ Le pas de temps de départ courant
6: /% Pour tous les pas de temps possibles. %/
7: **while** $t_{dep} \leq \min(T_{MaxStart}, T - L_{engthTask})$ **do**
8: $gainCourant = 0$ ▷ gain de la solution courante.
9: $solutionRealisable = true$ ▷ booléen qui teste la faisabilité de la solution.
10: **for all** $t \in [0; L_{engthTask}]$ **do**
11: **if** $I_{indispo}(t_{dep} + t) = 0$ **then** ▷ Le procédé est disponible
12: $gainCourant+ = P_r(t)(t_{dep} + t) * (S_{power}(t))$
13: **else** ▷ Le procédé est indisponible
14: $solutionRealisable = false$
15: $break$
16: **end if**
17: **end for**
18: /% Si la solution courante est viable et meilleure que la meilleure solution actuelle alors nous remplaçons la meilleure solution actuelle par la solution courante. %/
19: **if** $solutionRealisable \wedge (gainCourant > bestGain)$ **then**
20: $bestGain = gainCourant$
21: $bestStart = t_{dep}$
22: **end if**
23: $t_{dep}+ = 1$
24: **end while**
25: /% Nous retournons le meilleur pas de temps de départ. %/
26: **return** $bestStart$
27: **end procedure**

4.5.2 PSS

Nous allons maintenant nous intéresser à certains cas particuliers de PSS puis au problème général.

4.5.2.1 PSS_1

Nous allons commencer par étudier le cas particulier PSS_1 de PSS dans lequel toutes les contraintes potentielles portant sur ces procédés sont présentes sauf la contrainte de limitation du volume injecté/soutiré et la contrainte de gradient maximum comme indiqué dans le tableau ci-dessous.

Nom	a) I_{ndispo}	c) P_{min} & P_{max}	d) C_{start}	e) D_{iscr}	f) G_{max}	g) T_{maxOn}	h) T_{minOn}	i) T_{minOff}	j) M_{axCall}	k) M_{axVol}
PSS_1	X	X	X	X		X	X	X	X	

Proposition 2
PSS_1 est polynomial.

Preuve : Afin de déterminer la complexité de ce cas particulier, nous allons étudier les états possibles du procédé. Pour simplifier les explications, nous allons considérer un procédé sans stockage qui ne peut qu'injecter ($\forall t \in \mathcal{T}, P_{min}(t) = 0$). Nous proposons de décrire l'état du procédé à l'instant t grâce à un triplet (t, E_t, C_t) dans lequel E_t est défini de la façon suivante :

- si le procédé injecte dans le réseau, alors E_t est strictement positif et correspond au nombre de pas de temps écoulés depuis le démarrage de l'injection ;
- si le procédé ne doit pas injecter, alors E_t est strictement négatif et correspond au nombre de pas de temps restants avant que le procédé puisse injecter à nouveau ;
- si le procédé n'injecte pas et qu'il peut injecter alors E_t est nul.

De plus, dans ce triplet, C_t correspond au nombre de fois qu'une injection a été démarrée par le procédé entre 0 et t.

Enfin nous allons définir $G(t, E_t, C_t)$ comme le gain optimal du procédé dans l'état (t, E_t, C_t).

Nous considérons que l'état initial E_0 du procédé est (0,0,0) avec un gain nul (ce qui suit est facilement adaptable au cas où l'état initial n'est pas nul). Il est à noter que comme la contrainte de volume maximal d'injection et de soutirage n'est pas considérée ici, le procédé injectera ou soutirera toujours à son maximum. Nous rappelons que le profit maximal $P_{ro}^{max}(t)$ que le procédé peut faire au pas de temps t est le maximum sur les $\mathcal{D}(t)$ du profit pour un pas de temps t : $P_{ro}^{max}(t) = \max_{p \in D_{iscr}(t))} ((P_r(t) - C_{marg}) * p - C_{on})$. Nous noterons $p^*(t)$ la puissance permettant le profit $P_{ro}^{max}(t)$.

Suivant l'état courant (t, E_t, C_t) au pas de temps t, plusieurs états peuvent être atteignables à l'instant $t + 1$:

Cas 1 : Si $E_t < 0$ alors le procédé doit rester éteint. Donc à l'instant $t + 1$, nous pouvons atteindre uniquement l'état $(t + 1, E_t + 1, C_t)$ avec un gain $G(t + 1, E_t + 1, C_t) = G(t, E_t, C_t)$.

Cas 2 : Si $E_t = 0$ (i.e. le procédé est éteint mais pourrait être allumé) alors :

Cas 2.1 : Si $C_t < M_{axCall}^{p+}$ et $I_{ndispo}(t) = 0$ alors le procédé peut soit rester éteint, soit démarrer. Donc à l'instant $t+1$, nous pouvons soit atteindre l'état $(t+1, 0, C_t)$ avec un gain $G(t+1, 0, C_t) = G(t, 0, C_t)$, soit atteindre l'état $(t+1, 1, C_t+1)$ en produisant la puissance $p^*(t)$ permettant le profit $P_{ro}^{max}(t)$. Le gain est alors : $G(t + 1, 1, C_t + 1) = G(t, 0, C_t) + P_{ro}^{max}(t) - C_{start}$.

Cas 2.2 : Sinon (i.e. $C_t = M^{p^+}_{axCall}$ ou $I_{ndispo}(t) = 1$) le procédé doit rester éteint. Donc à l'instant $t+1$, nous pouvons atteindre uniquement l'état $(t+1, 0, C_t)$ avec un gain $G(t+1, 0, C_t) = G(t, 0, C_t)$.

Cas 3 : Si $E_t > 0$ (i.e. le procédé est allumé) alors :

Cas 3.1 : Si $E_t < T^{p^+}_{minOn}$ (i.e. le procédé doit rester allumé) alors

Cas 3.1.1 : Si $I_{ndispo}(t) = 0$ alors le procédé doit rester allumé. Donc à l'instant $t+1$, nous pouvons atteindre uniquement l'état $(t+1, E_t+1, C_t)$ avec une production de $p^*(t)$, donc avec un gain $G(t+1, E_t+1, C_t) = G(t, E_t, C_t) + P^{max}_{ro}(t)$.

Cas 3.1.2 : Sinon ($I_{ndispo}(t) = 1$) alors le procédé doit à la fois rester allumé et s'éteindre, ce qui est contradictoire. Donc à l'instant $t+1$, nous ne pouvons atteindre aucun état.

Cas 3.2 : Sinon si $E_t \geq T^{p^+}_{minOn}$ (i.e. le procédé peut être éteint) alors

Cas 3.2.1 : Si $E_t < T^{p^+}_{maxOn}$ et $I_{ndispo}(t) = 0$ alors le procédé peut être éteint ou rester allumé. Donc à l'instant $t+1$, nous pouvons soit atteindre l'état $(t+1, -T^{p^+}_{minOff}+1, C_t)$ avec un gain $G(t+1, -T^{p^+}_{minOff}+1, C_t) = G(t, E_t, C_t)$ si nous l'éteignons, soit atteindre l'état $(t+1, E_t+1, C_t)$ avec un gain $G(t+1, E_t+1, C_t) = G(t, E_t, C_t) + P^{max}_{ro}(t)$ s'il reste allumé.

Cas 3.2.2 : Sinon ($E_t = T^{p^+}_{minOn}$ ou $I_{ndispo}(t) = 1$) alors le procédé doit être éteint. Donc à l'instant $t+1$, nous pouvons atteindre uniquement l'état $(t+1, -T^{p^+}_{minOff}+1, C_t)$ avec un gain $G(t+1, -T^{p^+}_{minOff}+1, C_t) = G(t, E_t, C_t)$.

Le gain optimal est la plus grande valeur de $G(t, E_t, C_t)$ à l'instant final.

La figure 4.11 représente le graphe d'états associés à un procédé ne pouvant qu'injecter et ayant les caractéristiques suivantes :

- $T = 5$.
- $\forall t \in \mathcal{T}, P_{min}(t) = 0$.
- $\forall t \in \mathcal{T}, P_{max}(t) = 1$.
- $T^{p^+}_{minOn} = 2$.
- $T^{p^+}_{maxOn} = 3$.
- $T^{p^+}_{minOff} = 2$.
- $M^{p^+}_{axCall} = 2$.
- $P_r = \{1, 2, 3, 1, 3\}$.
- $C_{start} = 2$.
- $C_{on} = 0$.
- $C_{marg} = 0$.
- $\forall t \in \mathcal{T}, I_{ndispo}(t) = 0$.

Dans cet exemple, si le procédé injecte, il injectera obligatoirement une puissance égale à 1 ($\forall t \in \mathcal{T}, D_{iscr}(t) \in \{0, 1\}$). Dans ce graphe, il existe un arc entre 2 états i et j si nous pouvons atteindre j à partir de i. Le coût sur chaque arc (i, j) correspond au gain apporté par le passage de i à j. Ce gain vaut :

- 0 si le procédé passe d'un état (allumé ou éteint) à un état éteint ;
- $P^{max}_{ro}(t) - C_{start}$ si le procédé passe d'un état éteint à un état allumé (dans notre exemple, $P^{max}_{ro}(t) = P_r(t) * 1$) ;

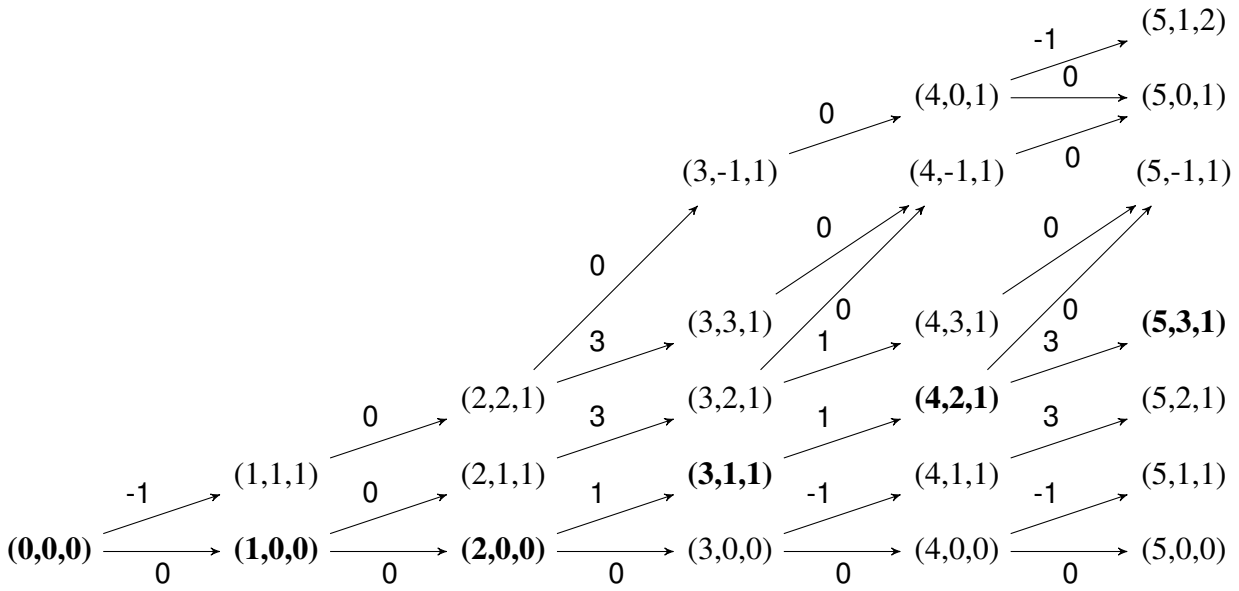

FIGURE 4.11 – Graphe pour l'exemple de PSS_1

- $P_{ro}^{max}(t)$ si le procédé passe d'un état allumé à un état allumé.

Ainsi par exemple, l'arc reliant l'état $(0,0,0)$ à l'état $(1,1,1)$ a un poids (gain) égal à $P_{ro}^{max}(1) - C_{start} = (P_r(1) * 1) - 2 = 1 - 2 = -1$. Un fois le graphe construit, il ne reste plus qu'à calculer un plus long chemin dans ce graphe. Le graphe étant décomposé en niveaux, nous pouvons déterminer ce plus long chemin en $O(nbArcs)$ grâce à l'algorithme de Bellman modifié pour la recherche d'un plus long chemin dans un graphe orienté acyclique [Sedgewick and Wayne, 2011]. Comme à partir de chaque état, nous pouvons au plus accéder à deux états (allumé ou éteint) alors $nbArcs$ est égal à $2 * nbEtats$. Or $nbEtats$ est égal à $Dimension(t) * Dimension(E_t) * Dimension(C_t)$ avec :

- t qui peut varier de 0 à T : $Dimension(t) = T + 1$;
- E_t qui peut varier de $-T_{minOff}^{p^+} + 1$ à $T_{maxOn}^{p^+}$: $Dimension(E_t) = T_{maxOn}^{p^+} + T_{minOff}^{p^+}$;
- C_t qui peut varier entre 0 et $M_{axCall}^{p^+}$: $Dimension(C_t) = M_{axCall}^{p^+} + 1$.

Puisque $T_{maxOn}^{p^+} \leq T, T_{minOff}^{p^+} \leq T$ et $M_{axCall}^{p^+} \leq \lceil \frac{T}{2} \rceil$, cet algorithme est en $O(T^3)$, donc polynomial. Dans notre exemple, le gain maximal est de 5 et est atteint dans l'état $(5,3,1)$.

Ces résultats sont facilement adaptables au problème d'un PSS qui peut injecter et soutirer. Chaque état est alors représenté par un quintuplet $(t, E_t^{inject}, C_t^{inject}, E_t^{extract}, C_t^{extract})$. L'algorithme est alors en $O(T^5)$.

Conclusion : il existe un algorithme polynomial permettant de résoudre PSS_1.

4.5.2.2 PSS_2

Nous allons maintenant étudier le cas particulier PSS_2 comportant uniquement la contrainte de limitation du volume demandé/produit en plus des contraintes de domaine de puissance et d'indisponibilité.

Nom	a) I_{ndispo}	c) P_{min} & P_{max}	d) C_{start}	e) D_{iscr}	f) G_{max}	g) T_{maxOn}	h) T_{minOn}	i) T_{minOff}	j) M_{axCall}	k) M_{axVol}	Complexité
PSS_1	X	X	X	X		X	X	X	X		Polynomial
PSS_2	X	X								X	

Proposition 3
PSS_2 est polynomial.

Preuve : La solution optimale de ce cas particulier est obtenue en appliquant l'algorithme 2. Cet algorithme consiste, tant que le volume maximal d'injection $M^{p^+}_{axVol}$ n'est pas atteint, à déterminer à chaque itération le pas de temps non encore considéré qui a le signal de prix le plus fort positif, et à produire à cet instant le plus possible. Puis tant que le volume maximal de soutirage $M^{p^-}_{axVol}$ n'est pas atteint, à déterminer à chaque itération le pas de temps non encore considéré qui a le signal de prix le plus faible négatif, et à consommer à cet instant le plus possible. Cet algorithme peut être implémenté en $O(T * log(T))$: il nécessite en effet dans un premier temps de trier les pas de temps par signal de prix croissant. Cette étape coûte $O(T * log(T))$. Puis nous parcourons au plus T pas de temps pour déterminer les puissances injectées et T pas de temps pour déterminer les puissances soutirées.

Conclusion : il existe un algorithme polynomial permettant de résoudre PSS_2.

4.5.2.3 PSS_3

Nous allons considérer maintenant le cas PSS_3 qui inclut la contrainte de discrétisation de la puissance en plus des contraintes de PSS_2.

Nom	a) I_{ndispo}	c) P_{min} & P_{max}	d) C_{start}	e) D_{iscr}	f) G_{max}	g) T_{maxOn}	h) T_{minOn}	i) T_{minOff}	j) M_{axCall}	k) M_{axVol}	Complexité
PSS_1	X	X	X	X		X	X	X	X		Polynomial
PSS_2	X	X								X	Polynomial
PSS_3	X	X		X						X	

Proposition 4
PSS_3 est NP-Difficile.

Preuve : Nous allons montrer que le problème de décision associé à PSS_3 est un problème NP-Complet en partant du problème Multiple choice subset sum ($MCSSP$). Plus précisément, nous allons considérer le problème de décision $PDPSS_3$ associé au cas particulier de PSS_3 dans lequel le procédé ne peut qu'injecter ($M^{p^-}_{axVol} = 0$ et $\forall t \in \mathcal{T}, P_{min}(t) = 0$) et dans lequel le signal de prix est constant et égal à 1.

Voici les définitions de **MCSSP** et $\mathbf{PDPSS_3}$.

MCSSP : soit m ensembles $\mathcal{S}^1$, $\mathcal{S}^2$, ..., $\mathcal{S}^m$, chaque ensemble $\mathcal{S}^i$ contenant N_i objets y^i_1 à $y^i_{N_i}$ où l'objet y^i_k a une valeur V^i_k, et un entier positif C. Existe-t-il un ensemble $\mathcal{E}$ contenant au plus un objet de chaque ensemble $\mathcal{S}^i$ tel que la somme des valeurs des objets de $\mathcal{E}$ soit égale à C ?

Algorithm 2 Algorithme optimal pour les PSS_2

1: /% Données :
- T : nombre de périodes de l'horizon de temps ;
- P_{max} : tableau contenant la puissance maximale de chaque pas de temps ;
- P_{min} : tableau contenant la puissance minimale de chaque pas de temps ;
- P_r : tableau contenant le signal de prix de chaque pas de temps ;
- $M^{p^+}_{axVol}$: volume maximal d'injections autorisé sur l'horizon ;
- $M^{p^-}_{axVol}$: volume maximal de soutirages autorisé sur l'horizon ;
- C_{on} : coût de maintien ;
- C_{marg} : coût marginal ;
- I_{ndispo} : tableau indiquant pour chaque pas de temps si le procédé est indisponible ;
- δ_p : tableau indiquant pour chaque pas de temps la variation de la puissance. %/

2: **procedure** $AlgoPSS_2(T, P_{max}, P_{min}, P_r, M^{p^+}_{axVol}, M^{p^-}_{axVol}, C_{on}, C_{marg}, I_{ndispo}, \delta_p)$
3: Trier les pas de temps dans l'ordre décroissant de leur signal de prix.
4: Les stocker dans cet ordre dans un tableau $tabTime$.
5: $volInject = 0$ et $gain = 0$ ▷ volume produit et gain sur l'horizon
6: $indInject = 0$ ▷ indice du tableau $tabTime$
7: **while** $indInject < T \ \& \ vol < M^{p^+}_{axVol}$ **do**
8: $t = tabTime[indInject]$ ▷ pas de temps courant
9: $puissance = \max(M^{p^+}_{axVol} - volInject, P_{max}(t))$ ▷ puissance maximale du pas de temps courant.
10: **if** $puissance * P_r(t) - |puissance| * C_{marg} - C_{on} > 0$ **then**
11: /% Si le profit associé est positif %/
12: **if** $I_{ndispo}(t) == 0$ **then**
13: /% Et si le procédé est disponible, alors mise à jour les variables. %/
14: $volInject+ = puissance$
15: $gain+ = puissance * P_r(t) - |puissance| * C_{marg} - C_{on}$
16: $indInject++$
17: $\delta_p(t) = puissance$
18: **end if**
19: **else**
20: break ▷ Si le profit est négatif à l'instant t, ce sera aussi le cas aux instants suivants.
21: **end if**
22: **end while**
23: $indExtract = T - 1$ ▷ indice du dernier pas de temps de soutirage.
24: $volExtract = 0$
25: **while** $indExtract \geq 0 \ \& \ vol < M^{p^-}_{axVol}$ **do**
26: $t = tabTime[indExtract]$
27: $puissance = -\min(M^{p^-}_{axVol} - volExtract, |P_{min}(t)|)$
28: **if** $puissance * P_r(t) - |puissance| * C_{marg} - C_{on} > 0$ **then**
29: **if** $I_{ndispo}(t) == 0$ **then** ▷ Si le profit associé est positif
30: /% Et si le procédé est disponible, alors mise à jour les variables. %/
31: $volExtract+ = |puissance|$
32: $gain+ = puissance * P_r(t) - |puissance| * C_{marg} - C_{on}$
33: $indExtract--$
34: $\delta_p(t) = puissance$
35: **end if**
36: **else**
37: break ▷ Si le profit est négatif à l'instant t, ce sera aussi le cas aux instants suivants.
38: **end if**
39: **end while**
40: **return** $gain$ ▷ Nous retournons le gain.
41: **end procedure**

$\mathbf{PDPSS_3}$:

- soit T, le nombre de périodes ;
- soit $M^{p^+}_{axVol}$, le volume maximal d'injection ;
- $\forall t \in [1, T]$:
  - soit $P_{max}(t)$, la puissance maximale d'injection au pas de temps t ;
  - soit $D_{iscr}(t)$, l'ensemble des puissances que le procédé peut atteindre au pas de temps t.

Existe-t-il $\forall t \in [1, T]$, une variation de puissance $\delta_p(t) \in D_{iscr}(t)$ telle que la somme des $\delta_p(t)$ est égale au volume d'injection $M^{p^+}_{axVol}$?

1. **Il est clair que $\mathbf{PDPSS_3}$ appartient à NP :** une proposition de solution OUI est vérifiable polynômialement ;
2. **Construction polynômiale d'une instance de $\mathbf{PDPSS_3}$ à partir d'une instance de MCSSP.** Pour toute instance de $MCSSP$, nous construisons une instance de $PDPSS_3$ dans laquelle :
   - $T = m$;
   - $M^{p^+}_{axVol} = C$;
   - $\forall t \in [1, T]$:
     - $P_{max}(t) = \max_{k \in [1, N_t]}(V^t_k)$;
     - $D_{iscr}(t) = \{0\} \cup \left(\cup_{k=1}^{N_t} V^t_k\right)$.

   Il est clair que cette construction est polynômiale.
3. **Si MCSSP a une solution alors $\mathbf{PDPSS_3}$ a une solution.** Si $MCSSP$ a une solution alors nous en déduisons une solution de $PDPSS_3$ dans laquelle $\forall t \in \mathcal{T}$, si la solution de $MCSSP$ ne contient pas d'objet de l'ensemble $\mathcal{S}^t$, alors le procédé a une variation de puissance nulle à t sinon le procédé à une puissance égale à la valeur de l'objet pris dans l'ensemble $\mathcal{S}^t$. La somme des puissances injectées vaut alors C.
4. **Si $\mathbf{PDPSS_3}$ a une solution alors MCSSP a une solution.** Si $PDPSS_3$ a une solution alors nous en déduisons une solution de $MCSSP$ dans laquelle $\forall t \in \mathcal{T}$, si dans la solution de $PDPSS_3$ la variation de puissance est nulle, alors la solution de $MCSSP$ ne contient pas d'objet dans la classe $\mathcal{S}^t$ sinon la solution de $MCSSP$ contient l'objet de valeur égale à la puissance produite au pas de temps t dans la solution de $PDPSS_3$. Cette solution a bien pour valeur C.
5. **Conclusion :** comme $MCSSP$ est NP-Complet [Pisinger, 1995], $PDPSS_3$ est NP-Complet. Ce cas particulier de PSS_3 est donc NP-Difficile, et donc le cas général PSS_3 aussi.

Bien que nous ne l'ayons pas prouvé, nous soupçonnons que les autres cas particuliers du problème (par exemple le problème comportant les contraintes de PSS_2 plus la durée maximale d'appel) sont également NP-Difficiles. Nous allons maintenant nous intéresser à la complexité du cas général.

4.5.2.4 Cas général de PSS

Nous allons étudier maintenant le cas général de PSS avec toutes les contraintes activées.

Nom	a) I_{ndispo}	c) P_{min} & P_{max}	d) C_{start}	e) D_{iscr}	f) G_{max}	g) T_{maxOn}	h) T_{minOn}	i) T_{minOff}	j) M_{axCall}	k) M_{axVol}	Complexité
PSS_1	X	X	X	X		X	X	X	X		Polynomial
PSS_2	X	X								X	Polynomial
PSS_3	X	X		X						X	NP-Difficile
PSS_{Global}	X	X	X	X	X	X	X	X	X	X	

Proposition 5
PSS est NP-Difficile au sens faible.

Preuve :

Comme PSS_3 est NP-Difficile, le problème général l'est également. Nous allons maintenant proposer un algorithme pseudo polynomial pour PSS, montrant du même coup que PSS est NP-Difficile au sens faible.

Comme dans l'algorithme proposé pour PSS_1, nous allons étudier les différents états possibles d'un procédé de type PSS.

Pour simplifier la compréhension, nous allons exposer les états atteignables pour un procédé ne pouvant qu'injecter ($\forall t \in \mathcal{T}, P_{min}(t) = 0$). Nous proposons de décrire l'état du procédé à l'instant t grâce à un quintuplet $(t, E_t, C_t, V_t, \delta_p(t))$ dans lequel E_t et C_t sont définis de la même façon que dans PSS_1. C'est-à-dire, E_t est défini de la façon suivante :

- si le procédé injecte dans le réseau, alors E_t est strictement positif et correspond au nombre de pas de temps écoulés depuis le démarrage de l'injection ;
- si le procédé ne doit pas injecter, alors E_t est strictement négatif et correspond au nombre de pas de temps restants avant que le procédé puisse injecter à nouveau ;
- si le procédé n'injecte pas et qu'il peut injecter alors E_t est nul.

De plus, C_t correspond au nombre de fois qu'une injection a été démarrée par le procédé entre 0 et t. De plus, V_t correspond au volume total que le procédé a injecté jusqu'à l'instant t. Finalement $\delta_p(t)$ correspond à la puissance injectée au pas de temps courant.

Enfin nous allons définir $G(t, E_t, C_t, V_t, \delta_p(t))$ comme le gain du procédé dans l'état $(t, E_t, C_t, V_t, \delta_p(t))$.

Nous considérons que l'état initial du procédé est (0,0,0,0,0) avec un gain nul (ce qui suit est facilement adaptable au cas où $E_0 \neq (0, 0, 0, 0, 0)$).

Suivant l'état courant $(t, E_t, C_t, V_t, \delta_t)$ à l'instant t, plusieurs états peuvent être atteignables à l'instant t+1 :

Cas 1 : si $E_t < 0$ alors le procédé doit rester éteint. Donc à l'instant t+1, nous pouvons atteindre uniquement l'état $(t+1, E_t+1, C_t, V_t, 0)$ avec un gain $G(t+1, E_t+1, C_t, V_t, 0) = G(t, E_t, C_t, V_t, \delta_t)$.

Cas 2 : si $E_t = 0$ (i.e. le procédé est éteint mais pourrait être allumé) alors :

Cas 2.1 : si $C_t = M^{p^+}_{axCall}$ ou $Indispo(t) = 1$ ou $V_t = M^{p^+}_{axVol}$ alors le procédé doit rester éteint. Donc à l'instant t+1, nous pouvons atteindre uniquement l'état $(t + 1, 0, C_t, V_t, 0)$ avec un gain $G(t + 1, 0, C_t, V_t, 0) = G(t, 0, C_t, V_t, 0)$.

Cas 2.2 : sinon (i.e. $E_t = 0$ et $C_t < M^{p^+}_{axCall}$ et $Indispo(t) = 0$ et $V_t < M^{p^+}_{axVol}$) alors le procédé peut rester éteint ou démarrer. Donc à l'instant t+1, nous pouvons soit atteindre l'état $(t+1, 0, C_t, V_t, 0)$ avec un gain $G(t+1, 0, C_t, V_t, 0) = G(t, 0, C_t, V_t, 0)$ s'il reste éteint, soit atteindre chacun des états $(t+1, 1, C_t+1, V_t+p, p)$ avec $p \in [1, min(P_{max}(t), M^{p^+}_{axVol} - V_t, G_{max})] \cap \mathcal{D}_{iscret}$ avec un gain $G(t+1, 1, C_t+1, V_t+p, p) = G(t, 0, C_t, V_t, 0) + (P_r(t) - C_{marg}) * p - C_{on} - C_{start}$ s'il est allumé.

Cas 3 : si $E_t > 0$: (i.e. le procédé est allumé) alors :

Cas 3.1 : si $\delta_t > G_{max}$ (i.e. le procédé doit rester allumé) alors :

Cas 3.1.1 : si $V_t = M^{p^+}_{axVol}$ ou $Indispo(t) = 1$ ou $E_t = T^{p^+}_{maxOn}$ (i.e le procédé doit être éteint) alors à l'instant t+1, nous ne pouvons pas atteindre aucun état.

Cas 3.1.2 : sinon (i.e. $V_t < M^{p^+}_{axVol}$ et $Indispo(t) = 0$ et $E_t < T^{p^+}_{maxOn}$) alors le procédé doit et peut rester allumé. Donc à l'instant t+1, nous pouvons atteindre chacun des états $(t+1, E_t+1, C_t, V_t+p, p)$ avec $p \in [\delta_t - G_{max}, min(P_{max}(t), M^{p^+}_{axVol} - V_t, \delta_t + G_{max})] \cap \mathcal{D}_{iscret}$ avec un gain $G(t+1, E_t+1, C_t, V_t+p, p) = G(t, E_t, C_t, V_t, \delta_t) + (P_r(t) - C_{marg}) * p - C_{on}$.

Cas 3.2 : sinon (i.e. $\delta_t \leq G_{max}$ le procédé peut être éteint) alors :

Cas 3.2.1 : si $E_t < T^{p^+}_{minOn}$ (i.e. le procédé doit rester allumé) alors :

Cas 3.2.1.1 : si $V_t = M^{p^+}_{axVol}$ ou $Indispo(t) = 1$ (i.e le procédé doit être éteint) alors à l'instant t+1, nous ne pouvons pas atteindre aucun état.

Cas 3.2.1.2 : sinon (i.e. $V_t < M^{p^+}_{axVol}$ et $Indispo(t) = 0$) alors le procédé doit et peut rester allumé. Donc à l'instant t+1, nous pouvons atteindre chacun des états $(t+1, E_t+1, C_t, V_t+p, p)$ avec $p \in [1, min(P_{max}(t), M^{p^+}_{axVol} - V_t, \delta_t + G_{max})] \cap \mathcal{D}_{iscret}$ avec un gain $G(t+1, E_t+1, C_t, V_t+p, p) = G(t, E_t, C_t, V_t, \delta_t) + (P_r(t) - C_{marg}) * p - C_{on}$.

Cas 3.2.2 : sinon (i.e. $E_t \geq T^{p^+}_{minOn}$) alors :

Cas 3.2.2.1 : si $V_t = M^{p^+}_{axVol}$ ou $Indispo(t) = 1$ ou $E_t = T^{p^+}_{maxOn}$ alors le procédé doit être éteint. Donc à l'instant t+1, nous pouvons atteindre uniquement l'état $(t+1, -T^{p^+}_{minOff} + 1, C_t, V_t, 0)$ avec un gain $G(t+1, -T^{p^+}_{minOff} + 1, C_t, V_t, 0) = G(t, E_t, C_t, V_t, \delta_t)$.

Cas 3.2.2.2 : sinon (i.e. $V_t < M^{p^+}_{axVol}$ et $Indispo(t) = 0$ et $E_t < T^{p^+}_{maxOn}$) alors le procédé peut être éteint ou rester allumé. Donc à l'instant t+1, nous pouvons soit atteindre l'état $(t+1, -T^{p^+}_{minOff} + 1, C_t, V_t, 0)$ avec un gain $G(t+1, -T^{p^+}_{minOff} + 1, C_t, V_t, 0) = G(t, E_t, C_t, V_t, \delta_t)$, soit atteindre chacun des états $(t+1, E_t+1, C_t, V_t+p, p)$ avec $p \in [1, min(P_{max}(t), M^{p^+}_{axVol} - V_t, \delta_t + G_{max})] \cap \mathcal{D}_{iscret}$ avec un gain $G(t+1, E_t+1, C_t, V_t+p, p) = G(t, E_t, C_t, V_t, \delta_t) + (P_r(t) - C_{marg}) * p - C_{on}$.

Ainsi le graphe des états possibles est défini de la façon suivante : il existe un arc entre 2 états i et j si nous pouvons atteindre j à partir de i. Le coût sur chaque arc (i, j) correspond au gain apporté par le passage de i à j. Ce gain vaut :

- 0 si le procédé passe d'un état (allumé ou éteint) à un état éteint ;
- $(P_r(t) - C_{marg}) * p - C_{on} - C_{start}$ si le procédé passe d'un état éteint à un état allumé avec une puissance p ;
- $(P_r(t) - C_{marg}) * p - C_{on}$ si le procédé passe d'un état allumé à un état allumé avec une puissance p.

Un fois le graphe construit, il ne reste plus qu'à calculer un plus long chemin dans ce graphe. Comme dans le cas de PSS_1, le graphe étant décomposé en niveaux, nous pouvons le déterminer en $O(nbArcs)$ grâce à l'algorithme de Bellman modifié pour la recherche d'un plus long chemin dans un

graphe orienté acyclique. Comme à partir d'un état, nous pouvons accéder à au plus P_{max} états alors $nbArcs$ est égal à $\max_t P_{max}(t) * nbEtats$. Or $nbEtats$ est égal à $Dimension(t) * Dimension(E_t) * Dimension(C_t) * Dimension(V_t) * Dimension(\delta_t)$ avec :

- t qui peut varier de 0 à T : $Dimension(t) = T + 1$;
- E_t qui peut varier de $-T^{p^+}_{minOff} + 1$ à $T^{p^+}_{maxOn}$: $Dimension(E_t) = T^{p^+}_{maxOn} + T^{p^+}_{minOff}$;
- C_t qui peut varier entre 0 et $M^{p^+}_{axCall}$: $Dimension(C_t) = M^{p^+}_{axCall} + 1$;
- V_t qui peut varier entre 0 et $M^{p^+}_{axVol}$: $Dimension(V_t) = M^{p^+}_{axVol} + 1$;
- δ_t qui peut varier entre 0 et $P_{max}(t)$: $Dimension(\delta_t) = P_{max} + 1$ avec $P_{max} = \max_t P_{max}(t)$.

Puisque $T^{p^+}_{maxOn} \leq T$, $T^{p^+}_{minOff} \leq T$ et $M^{p^+}_{axCall} \leq \lceil \frac{T}{2} \rceil$, cet algorithme est en $O(T^3 * M^{p^+}_{axVol} * P^2_{max})$, et donc pseudo-polynomial.

Ces résultats sont facilement adaptables à un procédé sans stockage qui peut injecter et soutirer. Chaque état est alors représenté par un octuplet $(t, E^{inject}_t, C^{inject}_t, V^{inject}_t, E^{extract}_t, C^{extract}_t, V^{extract}_t, \delta_p(t))$. L'algorithme est alors en $O(T^5 * M^{p^+}_{axVol} * M^{p^-}_{axVol} * (P_{max} + P_{min})^2)$, avec $P_{min} = \max_t P_{min}(t)$.

Conclusion : PSS est donc NP-Difficile au sens faible.

4.5.3 PAS

Nous allons maintenant nous intéresser aux PAS, dont les contraintes sont les suivantes :

Nom	a) I_{ndispo}	c) P_{min} & P_{max}	d) C_{start}	e) D_{iscr}	f) G_{max}	g) T_{maxOn}	h) T_{minOn}	i) T_{minOff}	j) M_{axCall}	k) M_{axVol}	l) Domaine stock	m) Pertes	n) Conservation	o) Variation stock	p) X_{min} & X_{max}	q) X^{unload}_{min}
PAS_1	X	X									X	X	X	X		
PAS	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Proposition 6
PAS est NP-Difficile.

Preuve : Nous allons commencer par étudier le cas particulier PAS_1 de PAS dans lequel le procédé n'est soumis à aucune contrainte optionnelle hormis celle de stock. Nous allons montrer que le problème de décision $PDPAS_1$ associé à un cas particulier de PAS_1 dans lequel le procédé est toujours disponible, le stockage est parfait, les coûts de départ et de maintien sont nuls et qui est consommateur, est un problème NP-Complet en partant d'un problème de planification de la production avec des coûts de départ et des coûts de stockage nuls (PP).

Problème de décision PP : [Garey and Johnson, 1979]

- soit N, le nombre de périodes ;
- $\forall t \in [1, N]$:
  - soit $r_t \in \mathbb{N}$, la demande au pas de temps t ;
  - soit $c_t \in \mathbb{N}$, la capacité de production du pas de temps t ;
  - soit $p_t \in \mathbb{N}$, le coût marginal de production du pas de temps t.
- soit B, un entier positif.

Existe-t-il $\forall t \in [1, N]$, une quantité de production y_t positive associée à un niveau de stock $I_t = \sum_{j=1}^{t}(y_j - r_j)$ tel que $0 \leq y_t \leq c_t$, $I_t \geq 0$ et $\sum_{j=1}^{t} p_j * y_i \leq B$?

Problème de décision $PDPAS_1$:

- soit T, le nombre de périodes ;
- soit C, la capacité de stockage ;
- $\forall t \in [1, T]$:
  - soit $P_{def}(t)$, la puissance par défaut au pas de temps t ;
  - soit $P_{min}^{conso}(t)$, la puissance minimale d'échange avec le réseau au pas de temps t ;
  - soit $P_r(t)$, le signal de prix au pas de temps t.
- soit G, un entier positif.

Existe-t-il $\forall t \in [1, T]$, une puissance $p_{gi}(t)$ associée à un niveau de stock $x(t) = \sum_{j=1}^{t}(P_{def}(j) - p_{gi}(j))$ telle que $P_{min}^{conso}(t) \leq p_{gi}(t) \leq 0$, $0 \leq x(t) \leq C$ et $\sum_{j=1}^{t} P_r(j) * (p_{gi}(j)) \geq G$?

Il est à noter que nous avons défini $PDPAS_1$ avec les $p_{gi}(t)$ plutôt qu'avec les $\delta_p(t)$ afin de faciliter la compréhension de la proposition.

1. **Il est clair que $PDPAS_1$ appartient à NP :** une proposition de solution OUI est vérifiable polynômialement.
2. **Construction polynômiale d'une instance de $PDPAS_1$ à partir d'une instance de PP.**
   Pour toute instance de PP, nous construisons une instance de $PDPAS_1$ ainsi :
   - $T = N$.
   - $C = \sum_{t=1}^{T}(c_t - r_t)$.
   - $\forall t \in [1, T]$:
     - $P_{def}(t) = -r_t$,
     - $P_{min}^{conso}(t) = -c_t$,
     - $P_r(t) = p_t$.
   - $G = B$.

   Il est clair que cette construction est polynomiale.
3. **Si PP a une solution alors $PDPAS_1$ a une solution.** Si PP a une solution alors nous en déduisons une solution de $PDPAS_1$ où $\forall t \in [1, T], p_{gi}(t) = -y_t$ et $x(t) = I_t$ ainsi :
   - $\forall t \in [1, T], x(t) = I(t) = \sum_{j=1}^{t}(y_j - r_j) \Rightarrow x(t) = \sum_{t=1}^{T}(P_{def}(t) - p_{gi}(t)) \Rightarrow x(t) \leq C$
   - $\forall t \in [1, T]$:
     - $0 \leq y_t \leq c_t \Rightarrow -c_t \leq -p_{gi}(t) \leq 0 \Rightarrow P_{min}^{conso}(t) \leq p_{gi}(t) \leq 0$
     - $I_t \geq 0 \Rightarrow x(t) \geq 0$
   - $\sum_{j=1}^{t} p_j * y_i \leq B \Rightarrow \sum_{j=1}^{t} P_r(j) * (-p_{gi}(j)) \leq G \Rightarrow \sum_{j=1}^{t} P_r(j) * p_{gi}(j) \geq G$
4. **Si $PDPAS_1$ a une solution alors PP a une solution.** Si $PDPAS_1$ a une solution alors nous en déduisons une solution de PP où $\forall t \in [1, T], y_t = -p_{gi}(t)$ et $x(t) = I_t$ ainsi
   - $\forall t \in [1, T]$:
     - $P_{min}^{conso}(t) \leq p_{gi}(t) \leq 0 \Rightarrow 0 \leq y_t \leq c_t$
     - $x(t) \geq 0 \Rightarrow I_t \geq 0$
   - $\sum_{j=1}^{t} P_r(j) * p_{gi}(j) \geq G \Rightarrow \sum_{j=1}^{t} p_j * (-y_j) \geq B \Rightarrow \sum_{j=1}^{t} p_j * y_i \leq B$

5. **Conclusion :** comme le problème de décision de planification de la production avec des coûts de départ et des coûts de stockage nuls est NP-Complet [Garey and Johnson, 1979], le problème de décision associé au cas particulier de PAS_1 est NP-Complet, et le problème de décision associé à PAS_1 est donc NP-Complet. PAS_1 et PAS sont donc NP-Difficiles.

Proposition 7
PAS est NP-Difficile au sens faible.

Preuve : Nous allons proposer un algorithme pseudo polynomial pour PAS. L'existence de cet algorithme démontrera du même coup que PAS est NP-Difficile au sens faible.

De la même façon que pour les PSS, nous pouvons énumérer les différents états possibles d'un PAS.

Nous proposons de décrire l'état X du procédé au pas de temps t grâce à l'ensemble suivant $(t, x_t, E_t^{inject}, C_t^{inject}, V_t^{inject}, E_t^{extract}, C_t^{extract}, V_t^{extract}, E_t^{charge}, C_t^{charge}, E_t^{decharge}, C_t^{decharge}, \delta_p(t))$ dans lequel x_t est le niveau du stock au pas de temps t et tous les autres paramètres sont définis par analogie de la même façon que dans le cas des PSS : par exemple C_t^{charge} correspond au nombre de fois qu'une charge a été démarrée par la procédé entre 0 et t. De plus nous allons définir $G(X)$ comme le gain du procédé dans l'état X.

De façon similaire au cas du problème PSS, nous pouvons construire un graphe dont les sommets sont les différents états possibles du procédé, et les arcs les transitions entre ces états. Il s'agit ensuite de résoudre un problème de plus long chemin dans un graphe orienté acyclique grâce à l'algorithme de Bellman en $O(nbEtats * nbArcs)$. À partir de chaque état indicé par t, nous pouvons au plus accéder à $P_{max}(t) - P_{min}(t)$ états suivant la puissance $\delta_p(t)$ produite, donc $nbArcs$ est égal à $\sum_t (P_{max}(t) - P_{min}(t)) * nbEtats(t)$. Or pour tout t, $nbEtats(t)$ est égal à :

$(C+1) * Dim(E_t^{inject}) * Dim(C_t^{inject}) * Dim(V_t^{inject}) * Dim(E_t^{extract}) * Dim(C_t^{extract}) * Dim(V_t^{extract}) * Dim(E_t^{charge}) * Dim(C_t^{charge}) * Dim(E_t^{decharge}) * Dim(C_t^{decharge}) * Dim(\delta)$

c'est-à-dire

$(C+1) * (T_{maxOn}^{p^+} + T_{minOff}^{p^+}) * M_{axCall}^{p^+} * M_{axVol}^{p^+} * (T_{maxOn}^{p^-} + T_{minOff}^{p^-}) * M_{axCall}^{p^-} * M_{axVol}^{p^-} * (T_{maxOn}^{x^+} + T_{minOff}^{x^+}) * M_{axCall}^{x^+} * (T_{maxOn}^{x^-} + T_{minOff}^{x^-}) * M_{axCall}^{x^-} * (P_{max}(t) - P_{min}(t))$

Cet algorithme est donc en $O(T^9 * C * M_{axVol}^{p^+} * M_{axVol}^{p^-} * \Delta_P^2)$ avec $\Delta_P = \max_t (P_{max}(t) - P_{min}(t))$. Il est donc pseudo-polynomial.

Conclusion : PAS est donc NP-Difficile au sens faible.

Il est à noter que les algorithmes pseudo-polynomiaux que nous venons de présenter pour les PSS et les PAS peuvent être réunis en un seul algorithme permettant de résoudre les problèmes de ces deux types de procédés.

4.5.4 Résultats

Nous avons testé ces algorithmes sur les instances que nous avons générées aléatoirement. Les calculs ont été réalisés sur un ordinateur avec un processeur Intel®Xeon®X5670 et une carte graphique NVIDIA®GeForce ®GTX 780.

Nous avons testé les algorithmes polynomiaux de PDC, PSS_1 et PSS_2 sur 100 instances générées aléatoirement comme expliqué dans la section 4.4. Ces instances composeront l'ensemble S_1. Le plan de production des instances de PDC et PSS_2 pour lesquels il existe un algorithme respectivement en $O(T^2)$ et en $O(T * log(T))$ sont résolues instantanément. Alors que le plan de production des instances de PSS_1 pour lequel il existe un algorithme en $O(T^5)$ est calculé en moins d'une minute avec une moyenne à 10 secondes.

Nous avons ensuite testé les algorithmes pseudo-polynomiaux des procédés PSS et PAS. 100 instances formant l'ensemble S_2 ont été générées aléatoirement (50 PSS et 50 PAS) comme expliqué dans la section 4.4. Ces algorithmes ne permettent pas de trouver des solutions pour des instances de plus de 380 millions d'états par période. Or, cette limite est atteinte très rapidement : aucune instance de S_2 ne peut être résolue par cette approche, la plus petite instance ayant plus d'un milliard d'états par période.

4.5.5 Conclusion

La tableau ci-dessous résume la complexité des différents problèmes :

Nom	a) I_{ndispo}	b) Intervalle de départ	c) P_{min} & P_{max}	d) C_{start}	e) D_{iscr}	f) G_{max}	g) T_{maxOn}	h) T_{minOn}	i) T_{minOff}	j) M_{axCall}	k) M_{axVol}	l) Domaine stock	m) Pertes	n) Conservation	o) Variation stock	p) X_{min} & X_{max}	q) X_{min}^{unload}	Complexité
PDC	X	X																Polynomial
PSS_1	X		X	X	X		X	X	X	X								Polynomial
PSS_2	X		X								X							Polynomial
PSS_3	X		X		X						X							NP-Difficile au sens faible
PSS	X		X	X	X	X	X	X	X	X	X							NP-Difficile au sens faible
PAS_1												X	X	X	X			NP-Difficile au sens faible
PAS	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	NP-Difficile au sens faible

Nous avons dans cette section prouvé que les PDC pouvaient être résolus facilement comme certains cas particuliers des PSS. Nous avons également proposé des algorithmes pseudo-polynomiaux pour les autres procédés. Cependant les limites de ces algorithmes sont très vite atteintes. Dans la prochaine section, nous allons proposer un modèle de programmation linéaire en nombres entiers pour les PSS et les PAS.

4.6 Programmation linéaire en nombres entiers

Nous allons dans cette section nous intéresser à la modélisation mathématique des problèmes des PSS et des PAS. Nous présentons tout d'abord un premier modèle de programmation linéaire en nombres entiers adapté de celui de [Delon and Normand, 2011], puis nous proposerons des améliorations.

4.6.1 Modèle initial

Le modèle mathématique proposé par [Delon and Normand, 2011] ne concerne que les procédés d'injection, et ne permet pas d'exprimer finement la contrainte de stock de ces procédés. Cette section présente une adaptation de ce programme linéaire de façon à modéliser tous les procédés PSS et PAS.

Les variables utilisées sont les suivantes :

- $\delta_p(t)$: la différence au pas de temps t entre la puissance réellement consommée/produite $p_{gi}(t)$ et la puissance consommée/produite par défaut $P_{def}(t)$;
- $\delta_{p^+}(t)$ et $\delta_{p^-}(t)$: respectivement la partie positive et la partie négative de $\delta_p(t)$;

- $x(t)$: le niveau du stock au pas de temps t ;
- $\delta_x(t)$: la variation du stock entre le pas de temps $t-1$ et le pas de temps t ;
- $\delta_{x^+}(t)$ et $\delta_{x^-}(t)$: respectivement la partie positive et la partie négative de la variation du stock entre le pas de temps $t-1$ et le pas de temps t ;
- $l(t)$: les pertes du stock au pas de temps t ;
- $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$ $p_\alpha(t)$ (p comme $positive$), $s_\alpha(t)$ (s comme $start$) et $e_\alpha(t)$ (e comme end) sont trois variables binaires reliées respectivement à $\delta_p^+(t)$, $\delta_p^-(t)$, $\delta_x^+(t)$ et $\delta_x^-(t)$ de la façon suivante :
  - $\forall t \in \mathcal{T}, p_\alpha(t) = \begin{cases} 1 \text{ si } \delta_\alpha(t) \text{ est positif} \\ 0 \text{ sinon} \end{cases}$

    Donc par exemple pour $\alpha = p^+$, $p_\alpha(t) = 1$ si $\delta_\alpha(t) > 0$, c'est à dire si le procédé injecte au pas de temps t.
  - $\begin{cases} s_\alpha(0) = \begin{cases} 1 \text{ si } \delta_\alpha(0) \text{ est positif} \\ 0 \text{ sinon} \end{cases} \\ \forall t \in \mathcal{T} \setminus \{0\}, s_\alpha(t) = \begin{cases} 1 \text{ si } \delta_\alpha(t) \text{ est positif et } \delta_\alpha(t-1) \text{ est nul} \\ 0 \text{ sinon} \end{cases} \end{cases}$

    Donc par exemple pour $\alpha = p^+$, $s_\alpha(t) = 1$ si $\delta_\alpha(t) > 0$ et $\delta_\alpha(t-1) = 0$, c'est à dire si le procédé démarre une injection au pas de temps t.
  - $\begin{cases} e_\alpha(0) = 0 \\ \forall t \in \mathcal{T} \setminus \{0\}, e_\alpha(0) = \begin{cases} 1 \text{ si } \delta_\alpha(t) \text{ est nul et } \delta_\alpha(t-1) \text{ est positif} \\ 0 \text{ sinon} \end{cases} \end{cases}$

    Donc par exemple pour $\alpha = p^+$, $e_\alpha(t) = 1$ si $\delta_\alpha(t) = 0$ et $\delta_\alpha(t-1) > 0$, c'est à dire si le procédé termine une injection au pas de temps t ;
- g : le gain du procédé sur l'horizon ;
- $c(t)$: le coût de production du procédé sur l'horizon ;

Les contraintes du problème s'expriment alors de la façon suivante :

- Relations entre les variables :
  - Relation entre $\delta_p(t)$ et $p_{gi}(t)$:

$$\forall t \in \mathcal{T}, \delta_p(t) = p_{gi}(t) - P_{def}(t) \tag{4.1}$$

  - Relation entre $\delta_p(t)$, $\delta_{p^+}(t)$ et $\delta_{p^-}(t)$:

$$\forall t \in \mathcal{T}, \delta_p(t) = \delta_{p^+}(t) - \delta_{p^-}(t) \tag{4.2}$$

  - Relation entre $p_{p^-}(t)$ et $p_{p^+}(t)$:

$$\forall t \in \mathcal{T}, p_{p^-}(t) + p_{p^+}(t) \leq 1 \tag{4.3}$$

    (la relation entre $p_{p^-}(t)$ et $\delta_{p^-}(t)$ d'une part, et $p_{p^+}(t)$ et $\delta_{p^+}(t)$ d'autre part est prise en compte dans l'équation 4.12 présentée plus loin).
  - Relation entre $\delta_x(t)$ et $x(t)$:

$$\forall t \in \mathcal{T}, \setminus\{0\}, \delta_x(t) = x(t) - x(t-1) \tag{4.4}$$

– Relation entre $\delta_x(t)$, $\delta_{x^+}(t)$ et $\delta_{x^-}(t)$:

$$\forall t \in \mathcal{T} \setminus \{0\}, \delta_x(t) = \delta_{x^+}(t) - \delta_{x^-}(t) \tag{4.5}$$

– Relation entre $p_{x^-}(t)$ et $p_{x^+}(t)$:

$$\forall t \in \mathcal{T} \setminus \{0\}, p_{x^-}(t) + p_{x^+}(t) \leq 1 \tag{4.6}$$

– Relation entre $p_{x^-}(t)$, $p_{x^+}(t)$ et $\delta_x(t)$:

$$\forall t \in \mathcal{T} \setminus \{0\}, -p_{x^-}(t) * C \leq \delta_x(t) \leq p_{x^+}(t) * C \tag{4.7}$$

– Relation entre les variables $s_\alpha(t)$ et $p_\alpha(t)$:
$s_\alpha(t)$ est égale à 1 si et seulement si $p_\alpha(t-1)$ est nulle et $p_\alpha(t)$ est égale à 1.

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \begin{cases} s_\alpha(0) = p_\alpha(0) \\ \forall t \in \mathcal{T} \setminus \{0\}, \begin{cases} p_\alpha(t) - p_\alpha(t-1) - s_\alpha(t) \leq 0 \\ -p_\alpha(t) + s_\alpha(t) \leq 0 \\ p_\alpha(t-1) + s_\alpha(t) \leq 1 \end{cases} \end{cases} \tag{4.8}$$

– Relation entre les variables $e_\alpha(t)$ et $p_\alpha(t)$:
$e_\alpha(t)$ est égale à 1 si et seulement si $p_\alpha(t-1)$ est égale à 1 et $p_\alpha(t)$ est nulle.

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \begin{cases} e_\alpha(0) = 0 \\ \forall t \in \mathcal{T} \setminus \{0\}, \begin{cases} p_\alpha(t-1) - p_\alpha(t) - e_\alpha(t) \leq 0 \\ -p_\alpha(t-1) + e_\alpha(t) \leq 0 \\ p_\alpha(t) + e_\alpha(t) \leq 1 \end{cases} \end{cases} \tag{4.9}$$

- Indisponibilité (contrainte a) :

$$\forall t \in \mathcal{T}, p_{p^+}(t) \leq 1 - I_{ndispo}(t) \tag{4.10}$$

$$\forall t \in \mathcal{T}, p_{p^-}(t) \leq 1 - I_{ndispo}(t) \tag{4.11}$$

- Domaine de puissance (contrainte c) :

$$\forall t \in \mathcal{T}, p_{p^-}(t) * P_{min}(t) \leq \delta_p(t) \leq p_{p^+}(t) * P_{max}(t) \tag{4.12}$$

- Discrétisation (contrainte e) : soit y_{ti} une variable binaire qui est égale à 1 si $\delta_p(t) = \mathcal{D}_{iscr}(t)[i]$, et 0 sinon.

$$\forall t \in \mathcal{T}, \begin{cases} \delta_p(t) = \sum_{i \in \mathcal{D}_{iscr}(t)} y_{ti} * \mathcal{D}_{iscr}(t)[i] \\ \sum_{i \in \mathcal{D}_{iscr}(t)} y_{ti} = 1 \end{cases} \tag{4.13}$$

- Gradient maximal entre deux puissances fournies consécutives (contrainte f) :

$$\forall t \in \mathcal{T} \setminus \{0\}, -G_{max} \leq \delta_p(t) - \delta_p(t-1) \leq G_{max} \tag{4.14}$$

- Durée maximale d'appel (contrainte g) : si le procédé commence à injecter (respectivement soutirer, stocker, décharger) au pas de temps t (i.e. $s_\alpha(t) = 1$) alors il doit arrêter d'injecter (respectivement soutirer, stocker, décharger) au plus tard à $t + T^\alpha_{maxOn}$ (i.e. $\sum_{t'=t+1}^{t+T^\alpha_{maxOn}} e_\alpha(t') \geq 1$).

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in [0; T - T^\alpha_{maxOn} - 1], s_\alpha(t) - \sum_{t'=t+1}^{t+T^\alpha_{maxOn}} e_\alpha(t') \leq 0 \tag{4.15}$$

- Durée minimale d'appel (contrainte h) : si le procédé commence à injecter (respectivement soutirer, stocker, décharger) au pas de temps t (i.e. $s_\alpha(t) = 1$) alors il ne peut s'arrêter d'injecter (respectivement soutirer, stocker, décharger) avant $t + T^\alpha_{minOn}$ (i.e. $\sum_{t'=t+1}^{\min(T-1,t+T^\alpha_{minOn}-1)} e_\alpha(t') \leq 0$).

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, s_\alpha(t) + \sum_{t'=t+1}^{\min(T-1,t+T^\alpha_{minOn}-1)} e_\alpha(t') \leq 1 \tag{4.16}$$

- Durée minimale entre un arrêt et le prochain départ (contrainte i) : si le procédé finit d'injecter (respectivement soutirer, stocker, décharger) au pas de temps t (i.e. $e_\alpha(t) = 1$) alors il ne peut pas recommencer à injecter (respectivement soutirer, stocker, décharger) avant le pas de temps $t + T^\alpha_{minOff}$ (i.e. $\sum_{t'=t+1}^{t+T^\alpha_{minOff}} s_\alpha(t') \leq 0$).

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, e_\alpha(t) + \sum_{t'=t+1}^{\min(T-1,t+T^\alpha_{minOff}-1)} s_\alpha(t') \leq 1 \tag{4.17}$$

- Nombre maximal d'appels pendant l'horizon $\mathcal{T}$ (contrainte j) :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \sum_{t \in \mathcal{T}} s_\alpha(t) \leq M^\alpha_{axCall} \tag{4.18}$$

- Quantité maximale d'énergie injectée ou soutirée pendant l'horizon $\mathcal{T}$ (contrainte k) :

$$\forall \alpha \in \{p^+; p^-\}, \sum_{t \in \mathcal{T}} \delta_\alpha(t) \leq M^\alpha_{axVol} \tag{4.19}$$

- Domaine du stock (contrainte l) :

$$\forall t \in \mathcal{T}, 0 \leq x(t) \leq C \tag{4.20}$$

- Pertes de stockage (contrainte m) :

$$\forall t \in \mathcal{T}, l(t) = P_{Loss} * x(t) + C_{Loss} \tag{4.21}$$

- Conservation de l'énergie (contrainte n) :

$$\forall t \in \mathcal{T}, \delta_{di}(t) = p_{ip}(t) - \delta_{ip}(t) \tag{4.22}$$

$$\forall t \in \mathcal{T}, \delta_{di}(t) = p_{gi}(t) - \delta_{gi}(t) \tag{4.23}$$

- Variation du stock (contrainte o) :

$$\forall t \in \mathcal{T} \setminus \{0\}, \delta^+_x(t) - \delta^-_x(t) = \delta_{ip}(t) - \delta_{gi}(t) - l(t) \tag{4.24}$$

- Bornes sur le niveau de stock à chaque instant (contrainte p) :

$$\forall t \in \mathcal{T}, X_{min}(t) \leq x(t) \leq X_{max}(t) \tag{4.25}$$

- Obligation d'attendre un niveau de stock minimal avant de pouvoir déstocker (contrainte q) :

$$\forall t \in \mathcal{T}, x(t) \geq s_{x^-}(t) * X^{unload}_{min} \tag{4.26}$$

- Gain :

$$g = \sum_{t \in \mathcal{T}} P_r(t) * (\delta_p(t) + P_{def}(t)) \tag{4.27}$$

- Coût de production (intégrant la contrainte d) concernant le coût de démarrage) :

$$\forall t \in \mathcal{T}, c = \sum_{t \in \mathcal{T}} C_{start} * (s_{p^+}(t) + s_{p^-}(t)) + C_{on} * (p_{p^+}(t) + p_{p^-}(t)) + C_{marg} * (\delta_{p^+}(t) + \delta_{p^-}(t)) \tag{4.28}$$

L'objectif est de maximiser le bénéfice :

$$\max g - c \tag{4.29}$$

4.6.2 Améliorations

Pour commencer, nous proposons deux reformulations des contraintes de durée minimale d'utilisation (4.16) et de durée minimale entre un arrêt et le prochain départ (4.17) :

- La contrainte 4.16 peut être reformulée grâce aux deux contraintes suivantes :
  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $s_\alpha(t) = 1$ (*i.e.* si par exemple le procédé commence à injecter au pas de temps t), alors pour tous les pas de temps t' entre t et $t + T^\alpha_{minOn} - 1$, $p_\alpha(t') = 1$ (*i.e.* le procédé doit continuer à injecter pendant au moins T^α_{minOn} pas de temps).
    Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T}, \min(T^\alpha_{minOn}, T - t) * s_\alpha(t) \leq \sum_{t'=t}^{\min(T-1, t+T^\alpha_{minOn}-1)} p_\alpha(t') \tag{4.30}$$

  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $e_\alpha(t) = 1$ (*i.e.* si par exemple le procédé arrête une injection au pas de temps t), alors pour tous les pas de temps t' entre $t - T^\alpha_{minOn}$ et $t - 1$, $p_\alpha(t') = 1$ (*i.e.* le procédé a injecté à chaque pas de temps entre $t - T^\alpha_{minOn}$ et $t - 1$). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T} \setminus \{0\}, \min(T^\alpha_{minOn}, t) * e_\alpha(t) \leq \sum_{t'=\max(0, t-T^\alpha_{minOn})}^{t-1} p_\alpha(t') \tag{4.31}$$

- La contrainte 4.17 peut être reformulée grâce aux deux contraintes suivantes :
  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $s_\alpha(t) = 1$ (*i.e.* si par exemple le procédé commence à injecter au pas de temps t), alors pour tout pas de temps t' entre $t - T^\alpha_{minOff}$ et $t - 1$, $p_\alpha(t') = 0$ (*i.e.* le procédé n'injecte pas pendant les T^α_{minOff} pas de temps précédant t). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T} \setminus \{0\}, \min(T^\alpha_{minOff}, t) * s_\alpha(t) \leq \sum_{t'=\max(0, t-T^\alpha_{minOff})}^{t-1} (1 - p_\alpha(t')) \tag{4.32}$$

  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $e_\alpha(t) = 1$ (*i.e.* si par exemple le procédé arrête d'injecter au pas de temps t), alors pour tous les pas de temps t' entre t et $t + T^\alpha_{minOff} - 1$, $p_\alpha(t') = 0$ (*i.e.* le procédé ne peut pas injecter à nouveau avant que T_{minOff} pas de temps se soient écoulés). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T}, \min(T^\alpha_{minOff}, T - t) * e_\alpha(t) \leq \sum_{t'=t}^{\min(T-1, t-T^\alpha_{minOff}-1)} 1 - p_\alpha(t') \tag{4.33}$$

De plus, nous proposons des contraintes "surrogate" découlant des contraintes de durée maximale d'appel (4.15), de durée minimale d'appel (4.16) et de durée minimale entre un arrêt et le prochain départ (4.17)

- Combinaison des contraintes 4.15 et 4.16 :
  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $s_\alpha(t) = 1$ (*i.e.* si par exemple le procédé commence à injecter au pas de temps t), alors il existe un pas de temps t' entre $t + T^\alpha_{minOn}$ et $t + T^\alpha_{maxOn}$ tel que $e_\alpha(t') = 1$ (*i.e.* le procédé doit arrêter d'injecter au plus tôt T^α_{minOn} et au plus tard T^α_{maxOn} pas de temps plus tard). Nous avons donc les contraintes suivantes :

$$\forall t \in [0; T - T^\alpha_{maxOn}], s_\alpha(t) \leq \sum_{t'=t+T^\alpha_{minOn}}^{t+T^\alpha_{maxOn}} e_\alpha(t') \tag{4.34}$$

  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $e_\alpha(t) = 1$ (*i.e.* si par exemple le procédé arrête d'injecter au pas de temps t), alors il existe un pas de temps t' entre $t - T^\alpha_{maxOn}$ et $t - T^\alpha_{minOn}$ tel que $s_\alpha(t') = 1$ (*i.e.* le procédé a commencé à injecter au plus tôt T^α_{maxOn} et au plus tard T_{minOn} pas de temps auparavant). Nous avons donc les contraintes suivantes :

$$\forall t \in [T^\alpha_{maxOn}; T], e_\alpha(t) \leq \sum_{t'=t-T^\alpha_{maxOn}}^{t-T^\alpha_{minOn}} s_\alpha(t') \tag{4.35}$$

- Combinaison des contraintes 4.16 et 4.17 :
  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $s_\alpha(t) = 1$ (*i.e.* si par exemple le procédé commence à injecter au pas de temps t), alors toutes les variables $e_\alpha(t')$ avec t' entre $t - T^\alpha_{minOff} + 1$ et $t + T^\alpha_{minOn} - 1$ sont nulles (le procédé n'a pas pu arrêter d'injecter entre t et $t - T^\alpha_{minOff} + 1$, et ne peut pas arrêter d'injecter avant $t + T^\alpha_{minOn} - 1$). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T}, s_\alpha(t) \leq 1 - \sum_{t'=\max(0,t-T^\alpha_{minOff}+1))}^{\min(T-1,t+T^\alpha_{minOn}-1)} e_\alpha(t') \tag{4.36}$$

  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, si $e_\alpha(t) = 1$ (*i.e.* si par exemple le procédé arrête d'injecter au pas de temps t), alors pour tous les pas de temps t' entre $t - T^\alpha_{minOn} + 1$ et $t + T^\alpha_{minOff} - 1$, $s_\alpha(t') = 0$ (*i.e.* le procédé ne peut pas avoir commencé à injecter entre $t - T^\alpha_{minOn}$ et t, et il ne peut pas commencer à injecter entre t et $t + T^\alpha_{minOff} - 1$). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T}, e_\alpha(t) \leq 1 - \sum_{t'=\max(0,t-T^\alpha_{minOn}+1))}^{\min(T-1,t+T^\alpha_{minOff}-1)} s_\alpha(t') \tag{4.37}$$

  - $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, $s_\alpha(t)$ ne peut prendre la valeur 1 qu'une seule fois sur chaque intervalle $[t; t+T^\alpha_{minOn}+T^\alpha_{minOff}-1]$ (*i.e.* par exemple le procédé ne peut démarrer qu'une seule injection sur l'intervalle $[t; t + T^\alpha_{minOn} + T^\alpha_{minOff} - 1]$). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T}, \quad \sum_{t'=t}^{\min(T-1,t+T^\alpha_{minOn}+T^\alpha_{minOff}-1)} s_\alpha(t') \leq 1 \tag{4.38}$$

– $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, $e_\alpha(t)$ ne peut prendre la valeur 1 qu'une seule fois sur chaque intervalle $[t; t + T^\alpha_{minOn} + T^\alpha_{minOff} - 1]$ (*i.e.* par exemple le procédé ne peut arrêter qu'une seule injection sur l'intervalle $[t; t + T^\alpha_{minOn} + T^\alpha_{minOff} - 1]$). Nous avons donc les contraintes suivantes :

$$\forall t \in \mathcal{T}, \qquad \sum_{t'=t}^{\min(T-1, t+T^\alpha_{minOn}+T^\alpha_{minOff}-1)} e_\alpha(t') \leq 1 \tag{4.39}$$

Il est à noter que toutes ces contraintes peuvent être écrites de façon désagrégées. La version désagrégée de ces contraintes figure en annexe (cf. Annexe A). Nous avons choisi de les présenter sous cette forme agrégée car même si les contraintes désagrégées trouvent en moyenne une meilleure borne supérieure au bout d'une minute, la solution entière trouvée au bout d'une minute est en moyenne moins bonne que celle trouvée avec les versions agrégées de ces contraintes.

Pour terminer, nous pouvons ajouter les trois contraintes suivantes qui limitent le nombre de départs (d'injection, de soutirage, de charge et de décharge), le nombre d'arrêts (d'injection, de soutirage, de charge et de décharge) et le nombre de périodes pendant lesquelles le procédé peut être utilisé (en injection, en soutirage, en charge et en décharge) :

- le nombre de départs (d'injection, de soutirage, de charge et de décharge) sur l'horizon $\mathcal{T}$ est limité à M^α_{axCall} (contrainte 4.18). Nous pouvons aussi le borner par $\lceil \frac{T}{T^\alpha_{minOn}+T^\alpha_{minOff}} \rceil$. Nous avons donc les contraintes suivantes :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \sum_{t \in \mathcal{T}} s_\alpha(t) \leq \lceil \frac{T}{T^\alpha_{minOn} + T^\alpha_{minOff}} \rceil \tag{4.40}$$

- Le nombre d'arrêts (d'injection, de soutirage, de charge et de décharge) sur l'horizon $\mathcal{T}$ est inférieur à $\min(M^\alpha_{axCall}, \lceil \frac{T}{T^\alpha_{minOn}+T^\alpha_{minOff}} \rceil)$. Nous avons donc les contraintes suivantes :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \sum_{t \in \mathcal{T}} e_\alpha(t) \leq \min(M^\alpha_{axCall}, \lceil \frac{T}{T^\alpha_{minOn} + T^\alpha_{minOff}} \rceil) \tag{4.41}$$

- Le nombre de périodes pendant lesquelles le procédé peut être utilisé (en injection, en soutirage, en charge et en décharge) est inférieur à $\min(M^\alpha_{axCall}, \lceil \frac{T}{T^\alpha_{minOn}+T^\alpha_{minOff}} \rceil) * T^\alpha_{maxOn}$. Nous avons donc les contraintes suivantes :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \sum_{t \in \mathcal{T}} p_\alpha(t) \leq \min(M^\alpha_{axCall}, \lceil \frac{T}{T^\alpha_{minOn} + T^\alpha_{minOff}} \rceil) * T^\alpha_{maxOn} \tag{4.42}$$

4.6.3 Expérimentations

Le modèle de base a été testé sur un processeur Intel®Xeon®X5670 en utilisant différents solveurs du marché : des solveurs payants (CPLEX, XPRESS-MP et GUROBI) et des solveurs gratuits (LPSolve, GLPK et Coin-OR). Aucun solveur gratuit arrivait à optimiser nos instances générées et les solveurs payants donnaient des résultats semblables. Nous avons donc choisi de tester les différentes versions du modèle en utilisant la version 12.2 de CPLEX. Le tableau 4.3 montre les résultats obtenus par le modèle de base pour l'ensemble d'instances S_2. Ce tableau est séparé en deux parties : la partie supérieure contient le nombre d'instances résolues par le modèle de base et la partie inférieure contient le temps de résolution en millisecondes. Ce tableau montre que le modèle de base prouve l'optimum en moins d'une minute pour 99 des 100 instances, avec un temps de résolution moyen de 763 millisecondes.

Les tableaux 4.4 et 4.5 contiennent le nombre d'instances résolues par le modèle de base et les temps d'exécution en millisecondes pour les instances de S_{2a} (50 PSS) et S_{2b} (50 PAS) respectivement. D'après

TABLE 4.3 – Résultats du modèle de programmation linéaire en nombres entiers de base sur S_2

Nombre d'instances pour lesquelles le temps de résolution du modèle de base est			
$< 10\,s$	entre $10\,s$ et $1\,min$	entre $1\,min$ et $5\,min$	$> 5\,min$
93	6	1	0
Temps de résolution (en ms)			
moyenne	minimum	maximum	95ème centile
763	3	80054	15863

TABLE 4.4 – Résultats du modèle de programmation linéaire en nombres entiers de base sur S_{2a} (PSS)

Nombre d'instances pour lesquelles le temps de résolution du modèle de base est			
$< 10\,s$	entre $10\,s$ et $1\,min$	entre $1\,min$ et $5\,min$	$> 5\,min$
44	5	1	0
Temps de résolution (en ms)			
moyenne	minimum	maximum	95ème centile
824	10	80054	24569

TABLE 4.5 – Résultats du modèle de programmation linéaire en nombres entiers de base sur S_{2b} (PAS)

Nombre d'instances pour lesquelles le temps de résolution du modèle de base est			
$< 10\,s$	entre $10\,s$ et $1\,min$	entre $1\,min$ et $5\,min$	$> 5\,min$
49	1	0	0
Temps de résolution (en ms)			
moyenne	minimum	maximum	95ème centile
702	3	15054	932

ces résultats, les PSS semblent un peu plus difficiles à résoudre que les PAS (44 instances résolues en moins de 10s contre 49 pour les PAS).

Afin de tester la stabilité et les limites de ce modèle, nous avons généré 5000 autres instances de PSS et 5000 autres instances de PAS dans le but de conserver les 100 plus difficiles de chaque ensemble. Nous avons donc gardé 100 instances de PSS qui nécessitent plus de 30 minutes de temps de résolution au modèle de base. Cet ensemble sera noté S_3. Cet ensemble sera par la suite divisé en 2 sous-ensembles : un sous-ensemble noté S_{3a} composé de 70 instances dont nous connaissons l'optimum, et un sous-ensemble S_{3b} composé des 30 instances restantes pour lesquelles nous ne connaissons qu'une borne supérieure. Nous n'avons par contre pas trouvé d'instances de PAS nécessitant plus de 2 minutes de temps de résolution. Nous pensons que les contraintes liées au stockage limitent grandement la combinatoire du problème.

Le tableau 4.6 montre l'impact des différentes améliorations que nous proposons sur la résolution de ses instances de S_3. Il est séparé en deux parties : la partie supérieure contient les écarts par rapport à l'optimum obtenus au bout d'une minute pour les instances de S_{3a}, et la partie inférieure présente les écarts par rapport à la meilleure borne supérieure trouvée par Cplex pour les instances de S_{3b}. Dans ce tableau, la ligne "reformulation" contient les résultats obtenues avec uniquement les contraintes de reformulation (contraintes 4.30 à 4.32). La ligne "surrogate" contient les résultats obtenus avec uniquement les contraintes "surrogate" (contraintes 4.34 à 4.39). La ligne "bornes supérieures" présente les résultats obtenus avec uniquement les bornes supérieures (contraintes 4.40 à 4.42). Enfin, la ligne "modèle amélioré" correspond aux résultats du modèle contenant toutes les améliorations (reformulations, contraintes surrogates, et bornes supérieures).

Nous pouvons constater que toutes les améliorations ont un apport sur la qualité de la solution. En effet, l'écart moyen du modèle de base sur les instances de S_{3a} est de 19,07 % contre 0,67 % pour les reformulations, 0,45 % avec les contraintes surrogates, et 3,08 % pour les bornes supérieures. De même sur les instances S_{3b}, l'écart moyen du modèle de base est de 9,29 % contre 5,75 % pour les reformulations, 5,65 % pour les contraintes surrogates et 6,35 % pour les bornes supérieures.

Nous voyons également que la version la plus performante et celle regroupant toutes les améliorations. En effet, pour cette version, l'écart moyen pour les instances S_{3a} est de 0,03 % avec un écart minimal à 0 % et un écart maximal à 0,44 % tandis qu'avec le modèle de base nous obtenons un écart moyen de 19,07 % avec un écart minimal à 0,01 % et un écart maximal à 100 %. Sur les instances S_{3b}, l'écart moyen est de 5,19 % avec un écart minimal à 0,01 % et un écart maximal à 44,41 % tandis qu'avec le modèle de base nous obtenons un écart moyen de 9,29% avec un écart minimal à 0,01 % et un écart maximal à 100 %.

Le tableau 4.7 montre les résultats obtenus sur l'ensemble S_2. Ce tableau est séparé en deux parties : la partie supérieure contient le nombre d'instances résolues et la partie inférieure contient le temps de résolution en millisecondes. Les résultats sont encore meilleurs avec le modèle amélioré : 99 instances sont résolues en moins de 10 secondes. De plus, le temps de résolution moyen est divisé par deux en passant de 763 ms à 310 ms.

Nous avons étudié d'un peu plus près les instances les plus difficiles à résoudre (ensemble S_{3b}). Nous avons pu constater que ces instances ont la particularité d'avoir un nombre maximal d'appels sur l'horizon $\mathcal{T}$ qui est très grand : en effet, le nombre maximal d'appels moyen des instances de S_{3a} est de 17, alors que celui des instances de S_{3b} est de 36. Le procédé peut alors être appelé un grand nombre de fois et le solveur semble avoir de la difficulté à bien placer ces appels.

De plus nous avons remarqué que les premières solutions trouvées par le solveur de programmation linéaire en nombres entiers étaient très éloignées de l'optimum : la première solution est la solution triviale consistant à conserver la puissance par défaut du procédé.

Étant donné qu'il est souhaitable de disposer de méthodes de résolution "gratuites", nous avons comparé les résultats du modèle amélioré sur S_{3a} et sur S_{3b} obtenues avec un logiciel payant (CPLEX) avec ceux obtenus avec un logiciel gratuit (LPSolve). Le tableau 4.8 contient dans sa partie supérieure les écarts obtenus par rapport à l'optimum pour les instances S_{3a}, et dans sa partie inférieure les écarts par rapport à la meilleure borne inférieure connue pour les instances S_{3b}.

Les performances du solveur LPSolve sont nettement moins bonnes que celles de CPlex. En effet, l'écart

moyen de LPSolve sur les instances de S_{3a} est de 56,12 % avec un écart minimal à 0,1 % et un écart maximal à 100 %. Sur les instances de S_{3b}, l'écart est de 51,89 % avec un minimum à 2,88 % et un maximum à 100 %. Cette approche n'est donc pas envisageable si nous souhaitons une méthode gratuite.

Nous proposons donc une approche "gratuite" dans la section suivante.

4.7 Programmation par contraintes

Nous allons dans cette section présenter un modèle de programmation par contraintes qui sera résolu par un solveur gratuit (Choco). Nous présentons dans un premier temps les contraintes globales utilisées dans le modèle, puis le modèle de programmation par contraintes et enfin les heuristiques de branchement que nous avons développées.

4.7.1 Contraintes utilisées dans le modèle

Nous présentons ici les quatre contraintes globales qui vont être utilisées dans le modèle de programmation par contrainte :

- $in(v, \mathcal{V})$ contraint la variable v à prendre une valeur de l'ensemble $\mathcal{V}$;
- $implies(A, B)$ contraint B à être vrai si A est vrai ;
- $regular(s, DFA)$ contraint la séquence de variables s à respecter l'expression régulière définie par l'automate fini déterministe (DFA) [Pesant, 2004]. Un DFA est une machine à états finis qui accepte/rejette une séquence finie de symboles et produit une unique exécution de l'automate pour chaque séquence [Hopcroft *et al.*, 2001] ;
- $costRegular(s, DFAWC, c)$ contraint en plus de $regular(s, DFA)$, la variable c à être égale à la somme des coûts des transitions donnés par l'automate fini déterministe avec des coûts sur ses transitions (DFAWC) [Demassey *et al.*, 2006].

4.7.2 Modèle

Cette section présente le modèle de programmation par contraintes que nous proposons. Il est à noter que dans ce modèle, nous avons directement intégré les contraintes qui limitent le nombre de départs (d'injection, de soutirage, de charge et de décharge), le nombre d'arrêts (d'injection, de soutirage, de charge et de décharge) et le nombre de périodes pendant lesquelles le procédé peut être utilisé (en injection, en soutirage, en charge et en décharge) trouvées pour le modèle de programmation linéaire en nombres entiers.

Ce modèle utilise les variables suivantes :

- $\delta_p(t)$: la différence au pas de temps t entre la puissance réellement consommée/produite $p_{gi}(t)$ et la puissance consommée/produite par défaut $P_{def}(t)$;
- $x(t)$: le niveau du stock au temps t ;
- $\delta_x(t)$: la variation du stock entre $t-1$ et t ;
- c^{p^+}, c^{p^-}, c^{x^+}, c^{x^-} : respectivement le nombre d'injections, d'extractions, de charges et de décharges ;
- g : le gain du procédé sur l'horizon ;
- c : le coût de production du procédé sur l'horizon.

Les contraintes du problème s'expriment alors de la façon suivante :

TABLE 4.6 – Comparaison des apports des améliorations sur S_3

Pour S_{3a}				
Méthodes	Gap au bout d'une minute			
	moyenne	min	max	95ème centile
Modèle de base	19,07 %	0,01 %	100 %	100 %
+ Reformulations	0,67 %	0,00 %	6,32%	2,75 %
+ Surrogate	0,45 %	0,00 %	3,63 %	2,05 %
+ Bornes supérieures	3,08 %	0,00 %	66%	29,49 %
Modèle amélioré	0,03 %	0,00 %	0,44 %	0,16 %
Pour S_{3b}				
Modèle de base	9,29 %	0,01 %	100 %	35,76 %
+ Reformulations	5,75 %	0,03 %	44,41%	25,69 %
+Surrogate	5,65 %	0,01 %	44,41 %	25,47 %
+Bornes supérieures	6,35 %	0,01 %	44,41 %	32,99 %
Modèle amélioré	5,19 %	0,01 %	44,41 %	22,90 %

TABLE 4.7 – Résultats sur S_2

Méthodes	Nombre d'instances pour lesquelles le temps de résolution est			
	$< 10\ s$	entre $10\ s$ et $1\ min$	entre $10\ s$ et $5\ min$	$> 5\ min$
Modèle de base	93	6	1	0
Modèle amélioré	99	1	0	0
	Temps de résolution (en ms)			
	moyenne	minimum	maximum	95ème centile
Modèle de base	763	3	60054	932
Modèle amélioré	310	8	7440	1539

TABLE 4.8 – Comparaison de solveurs sur S_3

Pour S_{3a}				
Solveur	Gap au bout d'une minute			
	moyenne	min	max	95ème centile
CPLEX	0,03 %	0,00 %	0,44 %	0,16 %
LPSolve	56,12 %	0,10 %	100 %	100 %
Pour S_{3b}				
CPLEX	5,19 %	0,01 %	44,41 %	22,90 %
LPSolve	51,89 %	2,88 %	100 %	100 %

- Relations entre les variables :
  - Relation entre $\delta_p(t)$ et $p_{gi}(t)$:

$$\forall t \in \mathcal{T}, \delta_p(t) = p_{gi}(t) - P_{def}(t) \tag{4.43}$$

  - Relation entre $\delta_x(t)$ et $x(t)$:

$$\forall t \in \mathcal{T} \setminus \{0\}, \delta_x(t) = x(t) - x(t-1) \tag{4.44}$$

- Indisponibilité (contrainte a) :

$$\forall t \in \mathcal{T}, implies(I_{ndispo}(t) = 1,\ \delta_p(t) = 0) \tag{4.45}$$

- Domaine de puissance (contrainte c) :

$$\forall t \in \mathcal{T}, in(\delta_p(t), [P_{min}(t), P_{max}(t)]) \tag{4.46}$$

- Discrétisation (contrainte e) :

$$\forall t \in \mathcal{T}, in(\delta_p(t), D_{iscr}(t)) \tag{4.47}$$

- Gradient maximal entre deux puissances fournies consécutives (contrainte f) :

$$\forall t \in \mathcal{T} \setminus \{0\}, |\delta_p(t) - \delta_p(t-1)| \leq G_{max} \tag{4.48}$$

- Contraintes sur les durées minimale et maximale d'appels, et sur la durée minimale entre un arrêt et un nouvel appel (contraintes g, h et i) :

  Chacune des ces contraintes peut être exprimée séparément avec la contrainte $regular$. Ainsi par exemple la contrainte de durée maximale d'injection s'écrit :

$$regular(\delta_p, MaxOn^{p^+})$$

  où $MaxOn^{p^+}$ est un DFA illustré par la figure 4.12. Chaque état représente le nombre de pas de temps depuis que le procédé a commencé à injecter. La transition entre deux états est possible si la condition sur la transition est vérifiée. Ainsi par exemple, à partir de l'état 0, l'automate reste dans cet état 0 tant que $\delta_p \leq 0$. Dès que la puissance devient positive, l'automate offre la possibilité soit de retourner dans l'état 0 (si l'injection s'arrête) soit de continuer l'injection et ce, jusqu'à l'état T^{p+}_{maxOn}. À partir de cet état T^{p+}_{maxOn}, la seule transition possible ramène à l'état 0 (arrêt de l'injection).

  De façon similaire, les autres contraintes de durée maximale d'appel s'écrivent ainsi :

$$\begin{cases} regular(\delta_p, MaxOn^{p^-}) \\ regular(\delta_x, MaxOn^{x^+}) \\ regular(\delta_x, MaxOn^{x^-}) \end{cases} \tag{4.49}$$

  où $MaxOn^{p^-}$, $MaxOn^{x^+}$ et $MaxOn^{x^-}$ sont respectivement les automates associés aux contraintes de durée maximale de soutirage, de charge et de décharge illustrés respectivement par les figures 4.13, 4.14 et 4.15. Il est à noter que les seules différences entre ces quatre automates sont les conditions sur leurs transitions.

  De la même façon, la durée minimale d'injection s'écrit :

$$regular(\delta_p, MinOn^{p^+})$$

où $MinOn^{p^+}$ est un DFA illustré par la figure 4.16. Et les autres contraintes s'écrivent :

$$\begin{cases} regular(\delta_p, MinOn^{p^-}) \\ regular(\delta_x, MinOn^{x^+}) \\ regular(\delta_x, MinOn^{x^-}) \end{cases} \tag{4.50}$$

où $MinOn^{p^-}$, $MinOn^{x^+}$ et $MinOn^{x^-}$ sont respectivement les automates associés aux contraintes de durée minimale de soutirage, de charge et de décharge.

De la même façon, la durée minimale entre un arrêt et la prochaine injection s'écrit :

$$regular(\delta_p, MinOff^{p^+})$$

où $MinOff^{p^+}$ est un DFA illustrés par la figure 4.17. Et les autres contraintes s'écrivent :

$$\begin{cases} regular(\delta_p, MinOff^{p^-}) \\ regular(\delta_x, MinOff^{x^+}) \\ regular(\delta_x, MinOff^{x^-}) \end{cases} \tag{4.51}$$

où $MinOff^{p^-}$, $MinOff^{x^+}$ et $MinOff^{x^-}$ sont respectivement les automates pour la durée minimale entre un arrêt et le prochain soutirage, charge et décharge .

Toutes ces contraintes peuvent être regroupées en une seule :

$$regular(\delta_p, Inject)$$

où $Inject$ est un DFA illustré par la figure 4.18 concaténant les contraintes de durées minimale et maximale d'injection, et de durée minimale entre un arrêt et le prochain départ d'une injection. De la même façon, les contraintes portant sur les durées minimale et maximale, et sur la durée minimale entre un arrêt et un départ de soutirage, de charge et de décharge s'écrivent :

$$\begin{cases} regular(\delta_p, Extract) \\ regular(\delta_x, Charge) \\ regular(\delta_x, Decharge) \end{cases} \tag{4.52}$$

où $Extract$, $Charge$ et $Decharge$ sont respectivement les automates illustrés respectivement par les figures 4.19, 4.20 et 4.21.

- Nombre maximal d'appels pendant $\mathcal{T}$ (contrainte j) :

  Chacune des contraintes limitant le nombre d'appels à une injection, un soutirage, une charge et à une décharge peut être exprimée séparément avec la contrainte $costregular$. Ainsi par exemple la contrainte du nombre maximal d'injections s'écrit $costregular(\delta_p, Call^{p^+}, c^{p^+})$ où $Call^{p^+}$ est un DFAWC illustré par la figure 4.22. Il y a deux états possibles dans cet automate : Off quand le procédé est éteint et On quand il est allumé. La transition entre deux états est possible si la condition sur la transition est vérifiée. La variable c^{p^+} est incrémentée du coût de la transition dès que l'automate passe par cette transition. Dans l'exemple, tant que $\delta_p \leq 0$, le procédé reste dans l'état Off. Dès que la puissance devient positive, il passe dans l'état On, et y reste tant que la puissance reste positive. La variable c^{p^+} est incrémentée de 1 dès que l'automate passe de l'état Off à l'état On. Cette variable compte donc le nombre de départs.

  De la même façon, les autres contraintes s'écrivent :

$$\begin{cases} costregular(\delta_p, Call^{p^-}, c^{p^-}) \\ costregular(\delta_x, Call^{x^+}, c^{x^+}) \\ costregular(\delta_x, Call^{x^-}, c^{x^-}) \end{cases} \tag{4.53}$$

  où $Call^{p^-}$, $Call^{x^+}$ et $Call^{x^-}$ sont respectivement les DFAWCs pour le nombre maximal de soutirages, de charges et de décharges illustrés respectivement par les figures 4.23, 4.24 et 4.25.

- Quantité maximale d'énergie pendant l'horizon $\mathcal{T}$ (contrainte k) :

$$\begin{cases} \sum_{t\in\mathcal{T}} \max(0, \delta_p(t)) \leq M_{axVol}^{p^+} \\ \sum_{t\in\mathcal{T}} \min(0, \delta_p(t)) \leq M_{axVol}^{p^-} \end{cases} \tag{4.54}$$

- Domaine du stock (contrainte l) :

$$\forall t \in \mathcal{T}, in(x(t), [0, C]) \tag{4.55}$$

- Pertes de stockage (contrainte m) :

$$\forall t \in \mathcal{T}, l(t) = P_{Loss} * x(t) + C_{Loss} \tag{4.56}$$

- Conservation de l'énergie (contrainte n) :

$$\forall t \in \mathcal{T}, \delta_{di}(t) = p_{ip}(t) - \delta_{ip}(t) \tag{4.57}$$

$$\forall t \in \mathcal{T}, \delta_{di}(t) = p_{gi}(t) - \delta_{gi}(t) \tag{4.58}$$

- Variation du stock (contrainte o) :

$$\forall t \in \mathcal{T} \setminus \{0\}, \delta_x(t) = \delta_{ip}(t) - \delta_{gi}(t) - l(t) \tag{4.59}$$

- Bornes sur le niveau de stock (contrainte p) :

$$\forall t \in \mathcal{T}, in(x(t), [X_{min}(t), X_{max}(t)]) \tag{4.60}$$

- Obligation d'attendre un niveau de stock minimal avant de pouvoir déstocker (contrainte q)) :

$$\begin{cases} implies(x(0) < X_{min}^{unload}, \delta_x(0) \geq 0) \\ \forall t \in \mathcal{T} \setminus \{0, T-1\}, \\ \quad implies(x(t) < X_{min}^{unload} \& \delta_x(t-1) \geq 0, \delta_x(t) \geq 0) \end{cases} \tag{4.61}$$

  Il est à noter que cette contrainte peut être modélisée par une contrainte regular mais l'automate associé aurait beaucoup d'états ($2 * C$) et beaucoup de transitions ($(P_{max}(t) - P_{min}(t)) * 2 * C$), ce qui ralentirait grandement la résolution du problème.

- Bornes supérieures sur le nombre d'appels à une injection, un soutirage, une charge et une décharge $\forall \alpha \in \{p^+; p^-; x^+; x^-\}$, c^α est compris entre 0 et le minimum entre M_{axCall}^α et $\lceil \frac{T}{T_{minOn}^\alpha + T_{minOff}^\alpha} \rceil$:

$$in(c^\alpha, [0, \min(M_{axCall}^\alpha, \lceil \frac{T}{T_{minOn}^\alpha + T_{minOff}^\alpha} \rceil)]) \tag{4.62}$$

- Gain :

$$g = \sum_{t\in\mathcal{T}} \big(PS(t) * (\delta_p(t) + P_{def}(t))\big) \tag{4.63}$$

- Coût de production :

  $costregular(\delta_p, Cost, c)$ où $Cost$ est le DFAWC illustré par la figure 4.26. Il y a deux états possibles dans cet automate : Off quand le procédé est éteint et On quand il est allumé. La transition entre deux états est possible si la condition sur la transition est vérifiée. Et la variable est incrémentée du coût de la transition dès que l'automate passe par cette transition. Dans l'exemple, tant que $\delta_p = 0$, le procédé reste dans l'état Off. Dès que la puissance devient non nulle, il passe dans l'état On et y reste tant qu'elle reste non nulle. La variable c est incrémentée du coût de départ, du coût de maintien et du coût marginal dès que l'automate passe de l'état Off à l'état On. Et elle est incrémentée du coût de maintien et du coût marginal tant que l'automate reste dans l'état On.

L'objectif est de maximiser le bénéfice :

$$\max\big(g - c\big) \tag{4.64}$$

4.7.3 Heuristiques de branchement

Un solveur de programmation par contraintes utilise une stratégie de branchement qui est composée d'un sélecteur de variables et d'un sélecteur de valeurs.

La stratégie par défaut de Choco consiste à choisir la première variable puis à l'instancier à sa borne supérieure. Nous proposons dans cette section deux autres stratégies : l'une destinée aux PSS, et l'autre aux PAS.

Pour un PSS, il est plus rentable d'injecter quand le signal de prix est positif et au plus haut, de soutirer quand le signal de prix est négatif et au plus bas, et de limiter le nombre d'injections et de soutirages. Afin de trouver les meilleures périodes pour augmenter sa production et sa consommation, nous proposons l'algorithme suivant :

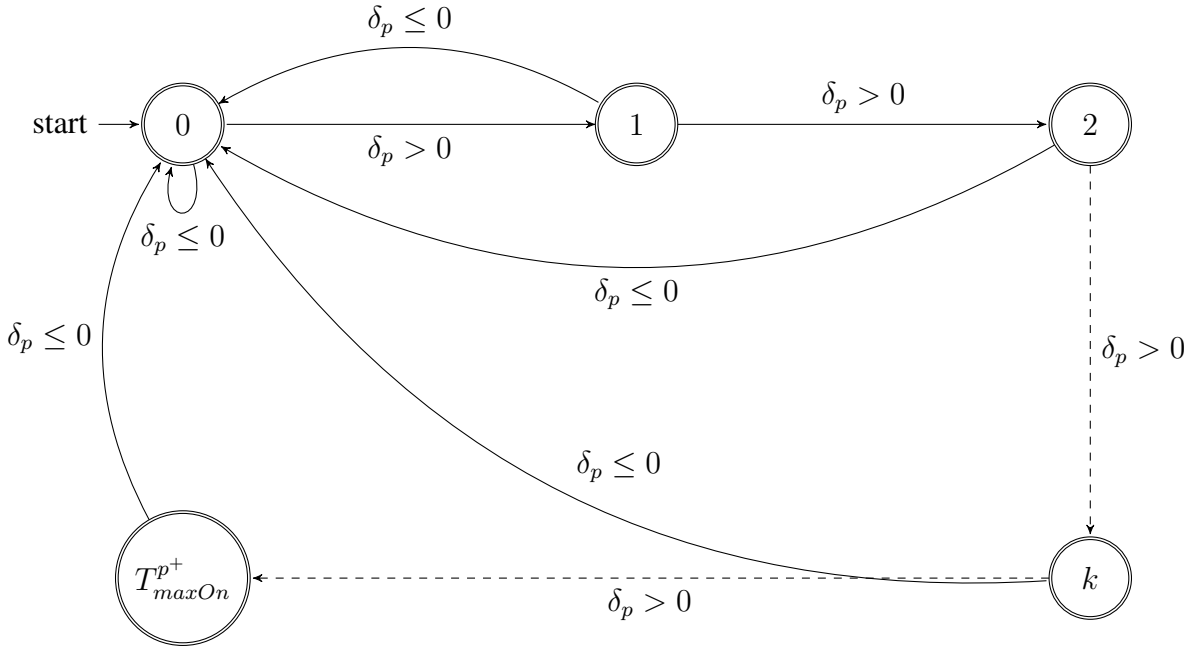

FIGURE 4.12 – Automate $MaxOn^{p^+}$ fini déterministe pour la contrainte de durée maximale d'injection

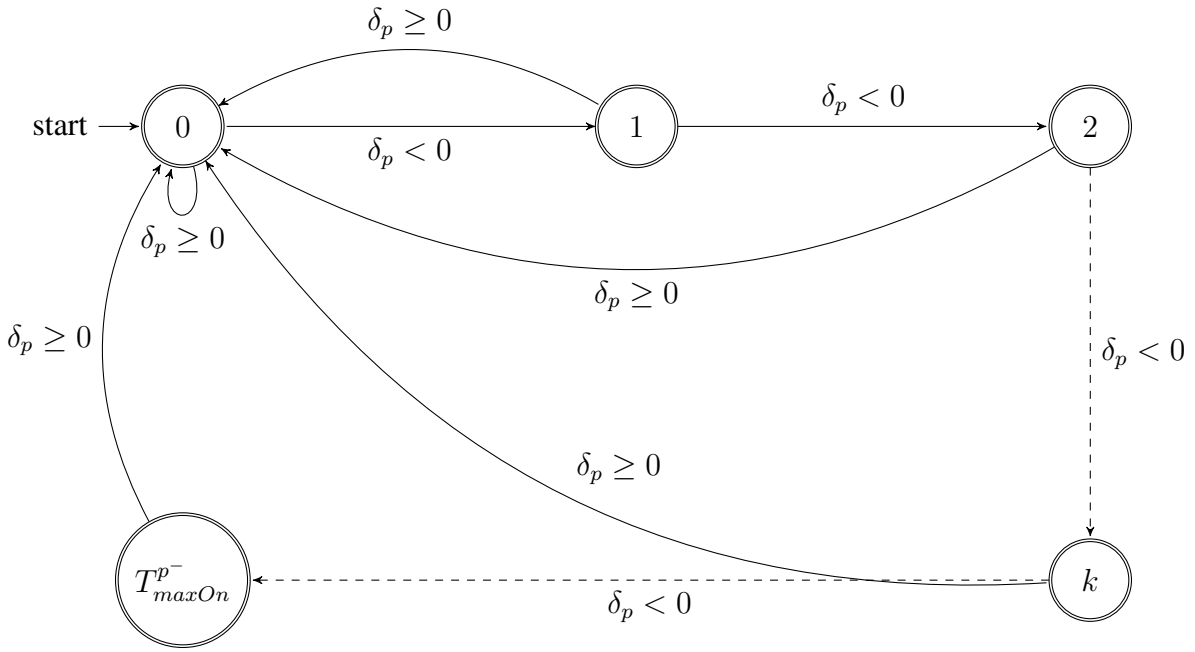

FIGURE 4.13 – Automate $MaxOn^{p^-}$ fini déterministe pour la contrainte de durée maximale de soutirage

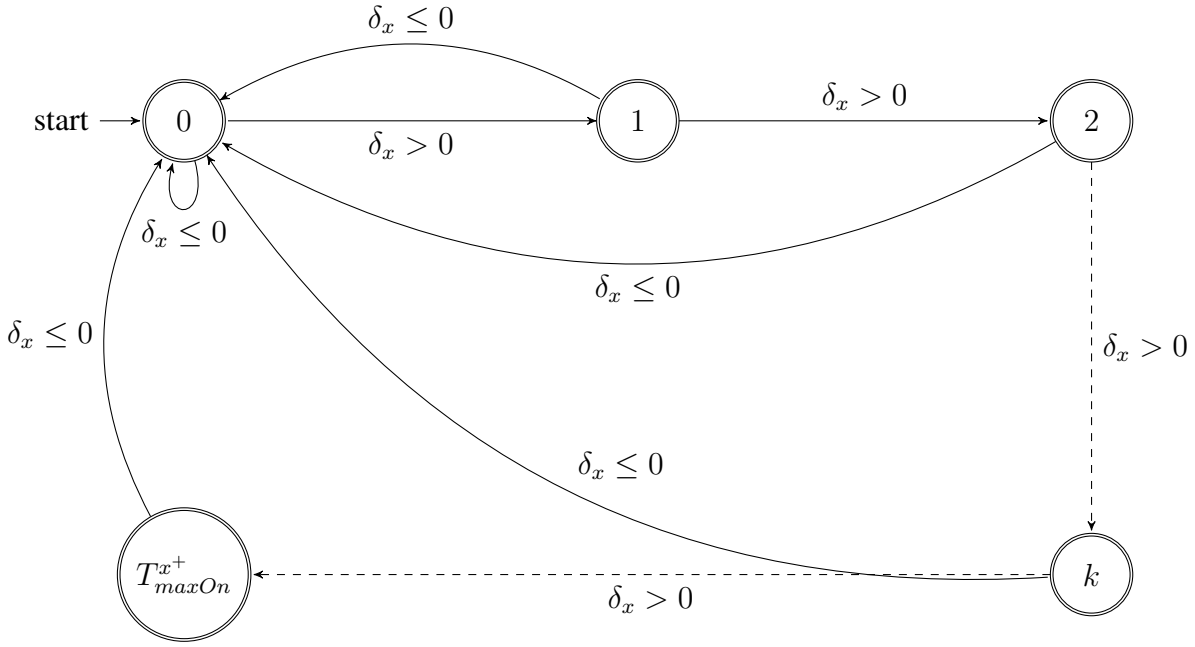

FIGURE 4.14 – Automate $MaxOn^{x^+}$ fini déterministe pour la contrainte de durée maximale de charge

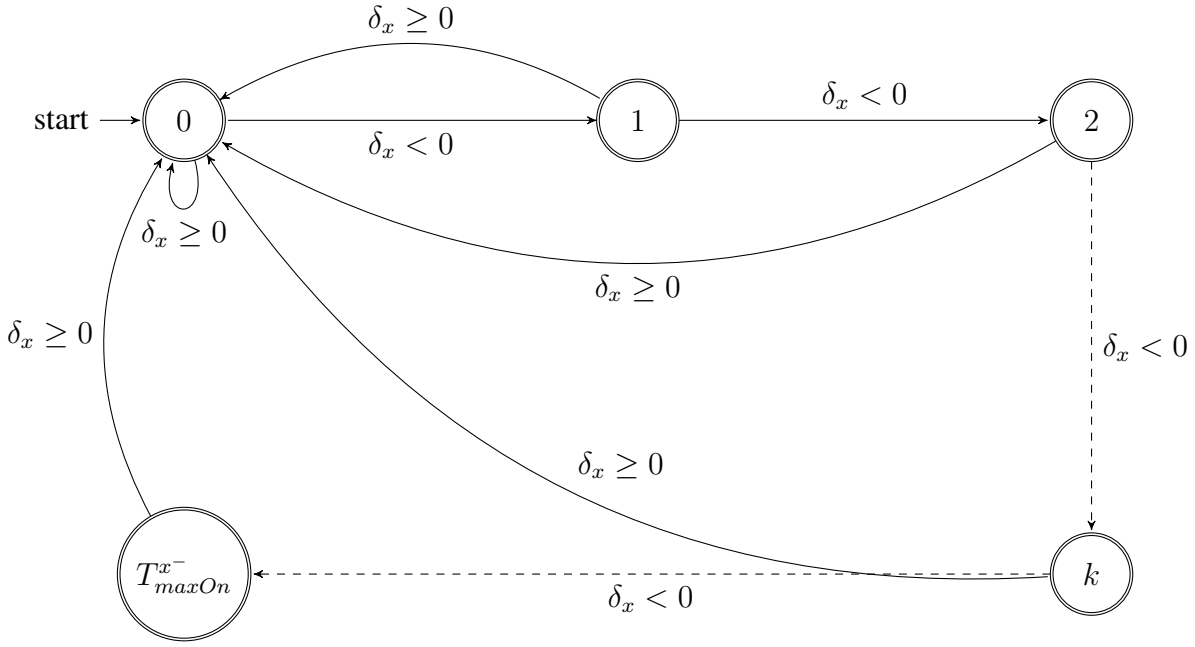

FIGURE 4.15 – Automate $MaxOn^{x^-}$ fini déterministe pour la contrainte de durée maximale de décharge

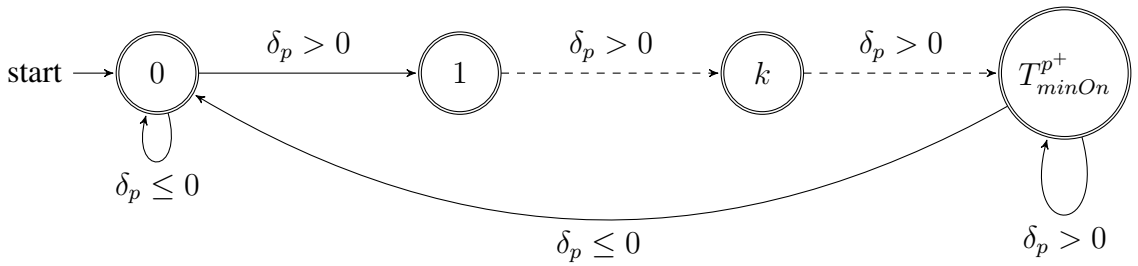

FIGURE 4.16 – Automate $MinOn^{p^+}$ fini déterministe pour la contrainte de durée minimale d'injection

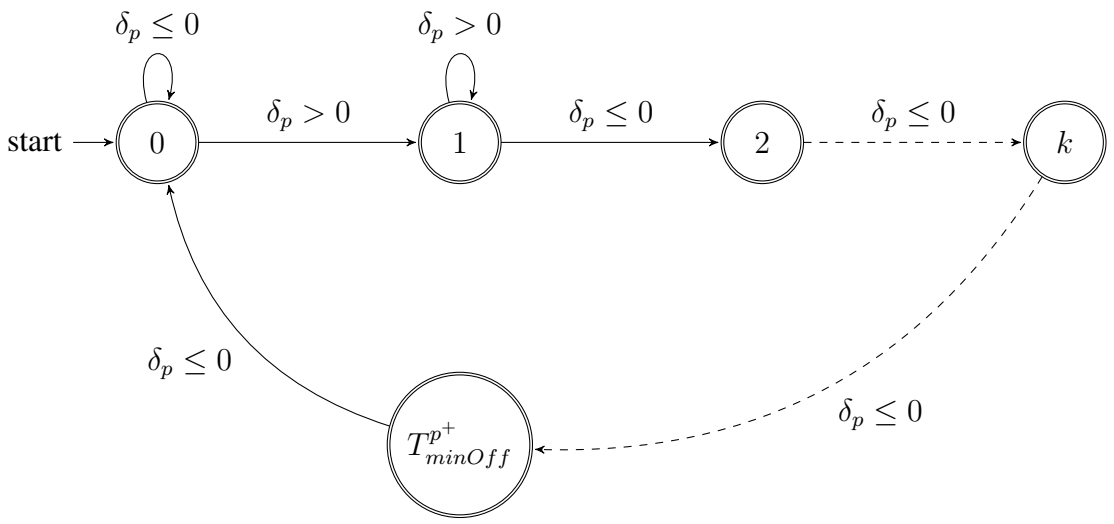

FIGURE 4.17 – Automate $MinOff^{p^+}$ fini déterministe pour la contrainte de durée minimale entre un arrêt et le prochain départ d'une injection

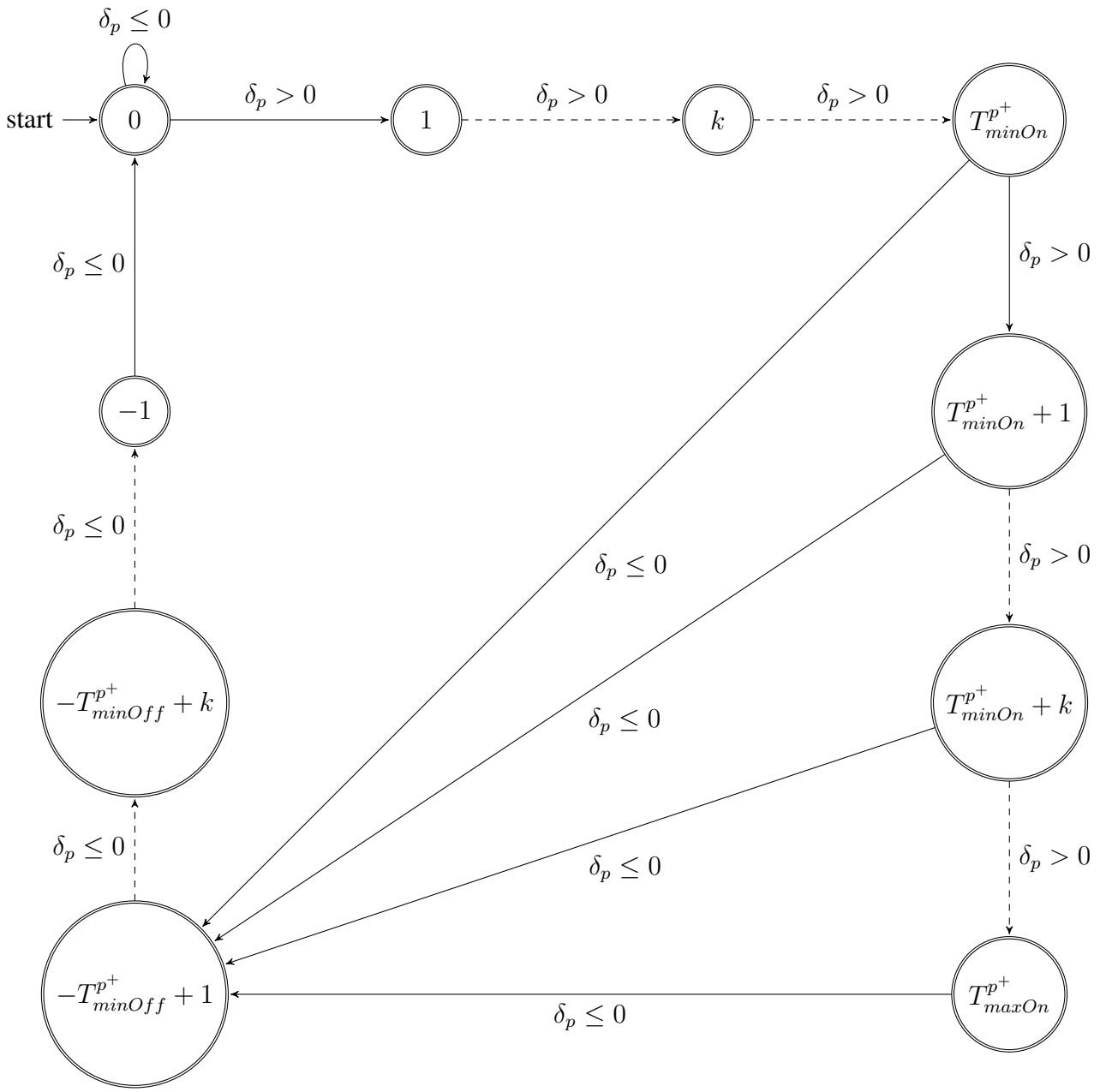

FIGURE 4.18 – Automate $Inject$ fini déterministe concaténant les contraintes de durées minimale et maximale d'injection, et de durée minimale entre un arrêt et le prochain départ d'une injection

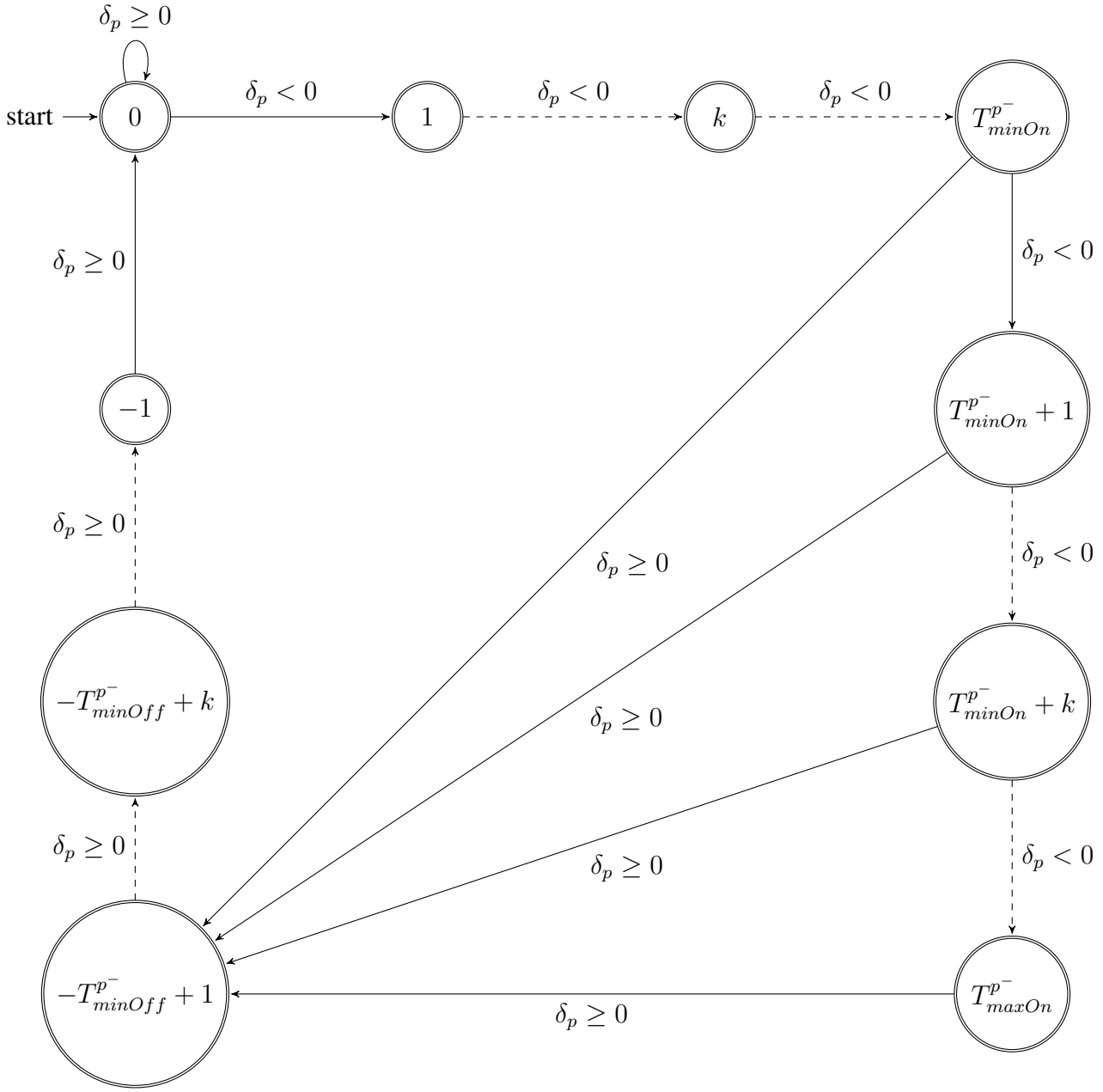

FIGURE 4.19 – Automate $Extract$ fini déterministe concaténant les contraintes de durées minimale et maximale de soutirage, et de durée minimale entre un arrêt et le prochain départ d'un soutirage

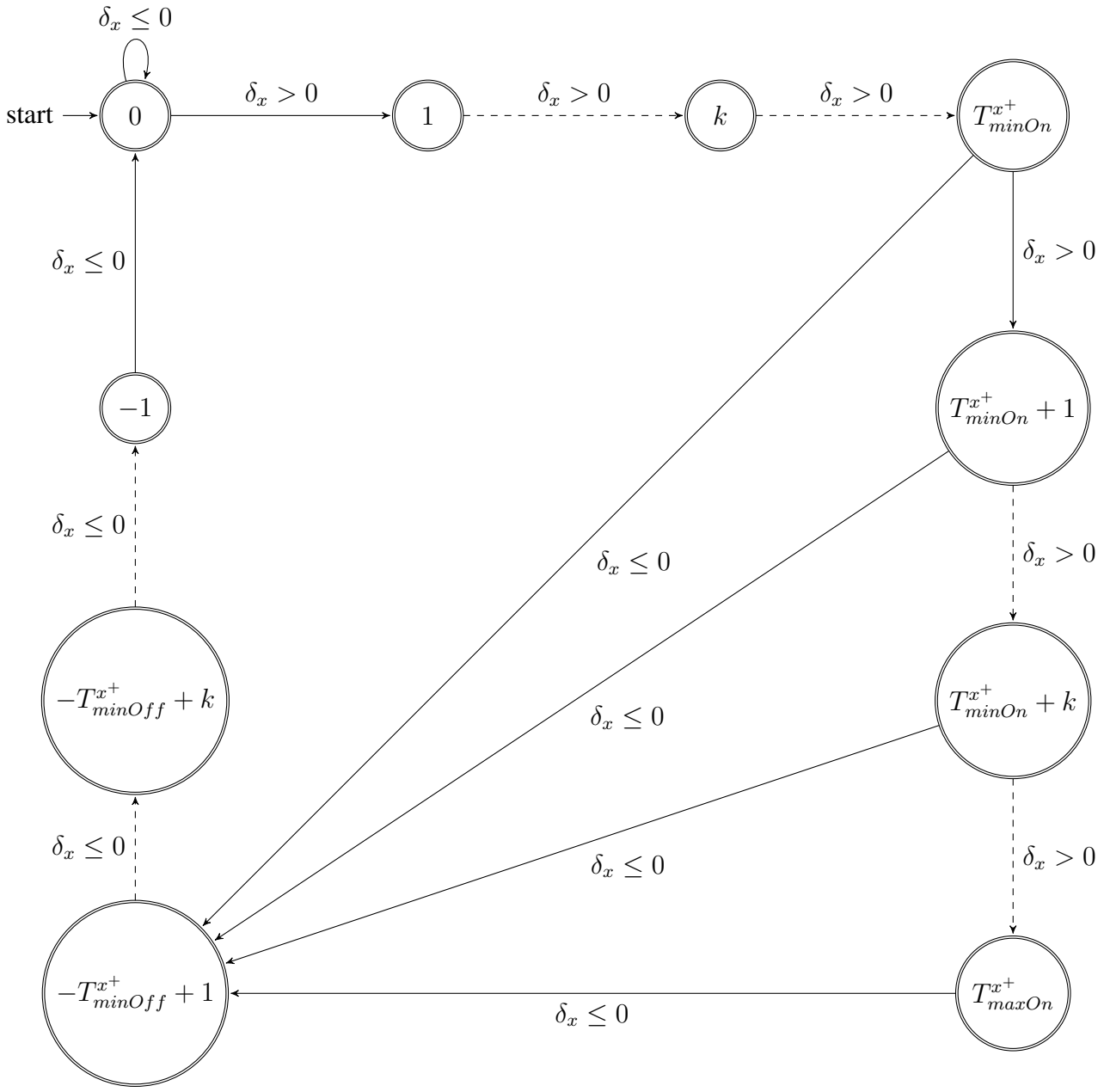

FIGURE 4.20 – Automate $Charge$ fini déterministe concaténant les contraintes de durées minimale et maximale de charge, et de durée minimale entre un arrêt et le prochain départ d'une décharge

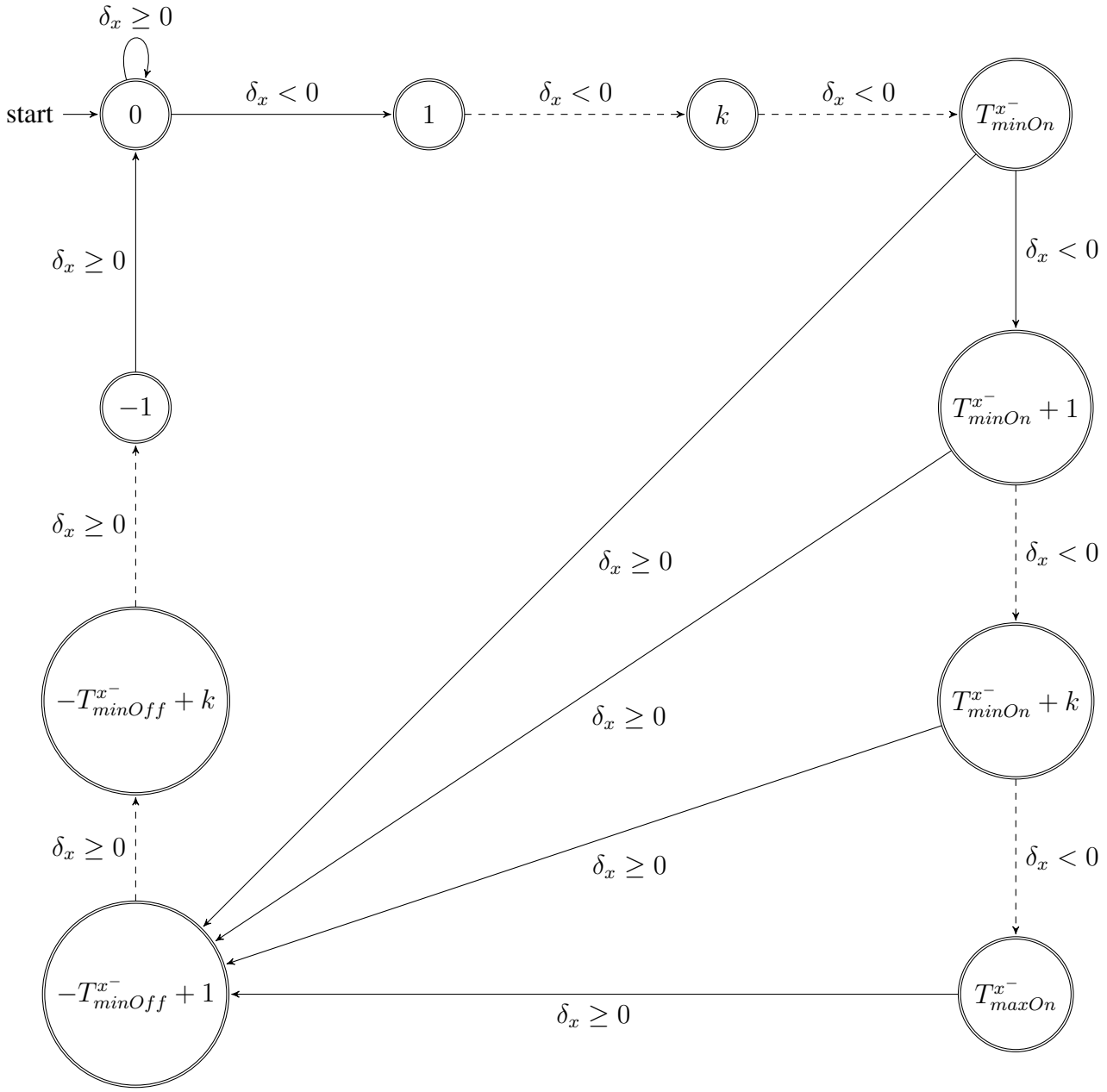

FIGURE 4.21 – Automate $Decharge$ fini déterministe concaténant les contraintes de durées minimale et maximale de décharge, et de durée minimale entre un arrêt et le prochain départ d'une décharge

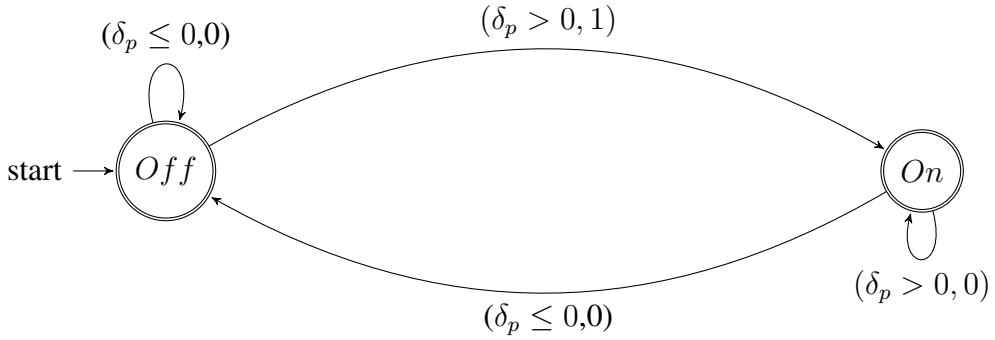

FIGURE 4.22 – Automate $Call^{p^+}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre d'injections

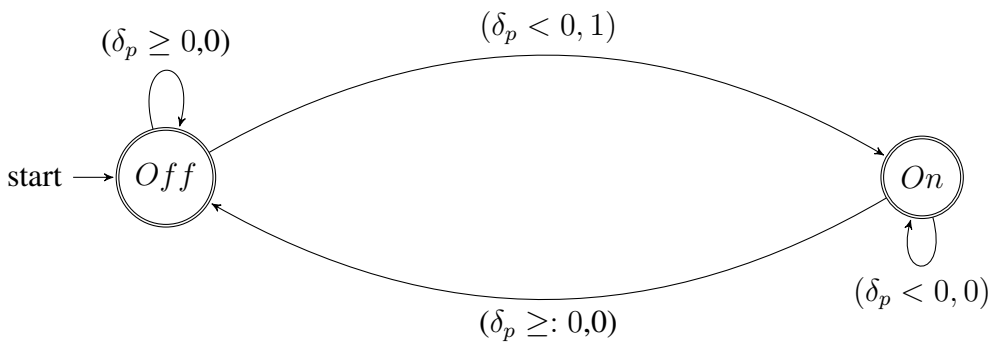

FIGURE 4.23 – Automate $Call^{p^-}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre de soutirages

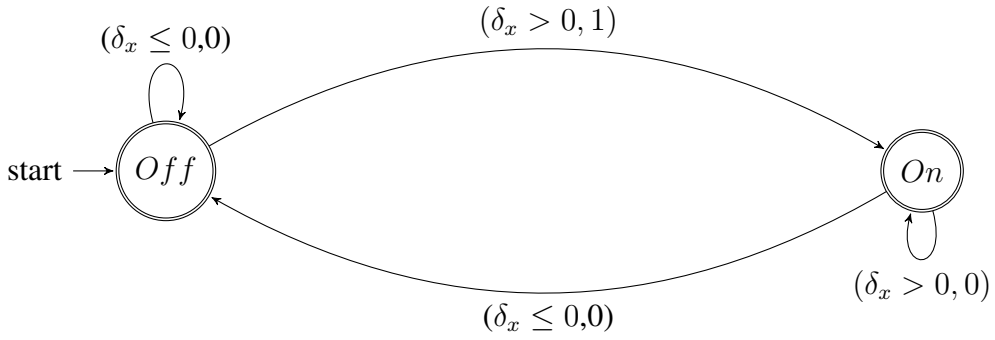

FIGURE 4.24 – Automate $Call^{x^+}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre de charges

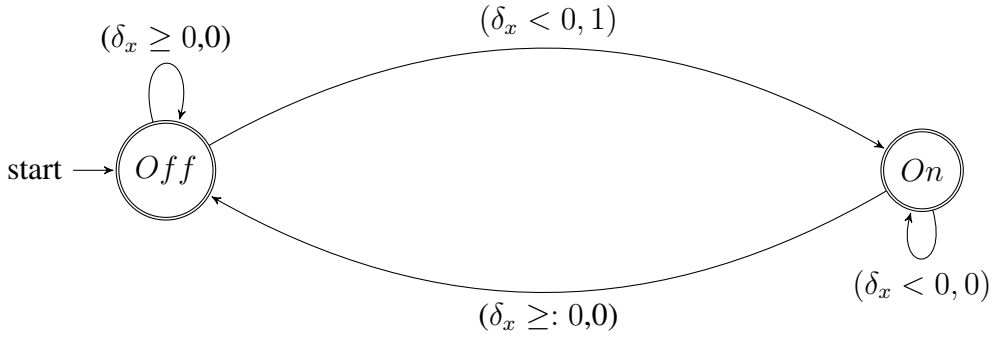

FIGURE 4.25 – Automate $Call^{x^-}$ fini déterministe avec un coût sur ses transitions pour la contrainte qui limite le nombre de décharges

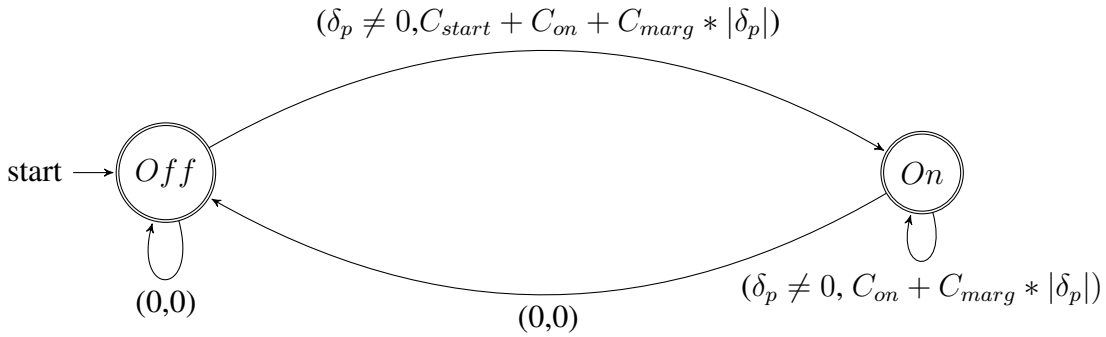

FIGURE 4.26 – Automate $Cost$ fini déterministe avec un coût sur ses transitions pour la contrainte de coût de production

Étape 1 : Premièrement, le sélecteur de variables choisit la variable $\delta_p(t)$ non encore instanciée associée à la période t ayant le plus grand prix par kWh en valeur absolue, et supérieur au coût marginal. Si aucune variable ne peut être instanciée alors l'algorithme s'arrête. Sinon nous initialisons un ensemble ordonné S avec t. Si le prix est positif, le procédé a intérêt à injecter. Le sélecteur de valeur assigne alors à $\delta_p(t)$ la plus grande valeur possible et l'algorithme passe à l'étape 2. Sinon le procédé a intérêt à soutirer. Le sélecteur de valeur assigne alors à $\delta_p(t)$ la plus petite valeur négative possible et l'algorithme passe à l'étape 3.

Étape 2 : Dans cette étape, l'algorithme essaye d'étendre l'injection courante afin de limiter le nombre d'injections. Le sélecteur de variables choisit donc la variable $\delta_p(t')$ non encore instanciée associée à la période t' située à une extrémité de S (t' est soit $min(S) - 1$ soit $max(S) + 1$) ayant le plus prix par kWh positif supérieur au coût marginal. Si aucune variable ne peut être instanciée alors l'algorithme retourne à l'étape 1. Sinon nous ajoutons t' à S. Ensuite le sélecteur de valeur assigne à la variable $\delta_p(t')$ la plus grande valeur possible et l'algorithme refait de nouveau cette étape.

Étape 3 : Dans cette étape, l'algorithme essaye d'étendre le soutirage courant afin de limiter le nombre de soutirages. Le sélecteur de variables choisit donc la variable $\delta_p(t')$ non encore instanciée associée à la période t' située à une extrémité de S (t' est soit $min(S) - 1$ ou $max(S) + 1$) ayant le plus petit prix négatif supérieur en valeur absolue au coût marginal. Si aucune variable ne peut être instanciée alors l'algorithme retourne à l'étape 1. Sinon nous ajoutons t' à S. Ensuite le sélecteur de valeur assigne à $\delta_p(t')$ la plus petite valeur possible et l'algorithme refait de nouveau cette étape.

En ce qui concerne les PAS, ils ont intérêt à charger leur stock quand le prix par kWh est négatif et au plus bas, à le décharger quand il est positif et au plus haut, et le nombre de charges et de décharges est limité. Il parait pour ce problème plus naturel de brancher non pas sur les variables $\delta_p(t)$ mais sur les variables $\delta_x(t)$. Les valeurs des variables $\delta_p(t)$ seront directement déduites de celles des variables $\delta_x(t)$ par les contraintes 4.43, 4.57, 4.58 et 4.59. Voici l'algorithme que nous proposons :

Étape 1 : Premièrement, le sélecteur de variable choisit la variable $\delta_x(t)$ non encore instanciée associée à la période t ayant le plus petit prix par kWh. Si aucune variable ne peut être instanciée, alors l'algorithme s'arrête. Sinon nous initialisons un ensemble ordonné S_{charge} avec t. Ensuite le sélecteur de valeur assigne à $\delta_x(t)$ la plus petite valeur négative possible et l'algorithme passe à l'étape 2.

Étape 2 : Dans cette étape, l'algorithme essaye d'étendre la période de charge courante afin de limiter le nombre de charges. Le sélecteur de variable choisit donc la variable qui n'est pas encore instanciée associée à la période t' située à une extrémité de S_{charge} (t' est soit $min(S_{charge}) - 1$ soit $max(S_{charge}) + 1$) ayant le plus petit prix. Si aucune variable ne peut être instanciée alors l'algorithme passe à l'étape 3. Sinon nous ajoutons t' à S_{charge}. Ensuite le sélecteur de valeur assigne à $\delta_x(t')$ la plus petite valeur négtive possible et l'algorithme refait de nouveau cette étape.

Étape 3 : À ce stade de l'algorithme, le sélecteur de variable choisit la variable $\delta_x(t)$ qui n'est pas encore instanciée associée à la période t ayant le plus grand signal de prix. Si aucune variable ne peut être instanciée alors l'algorithme s'arrête. Sinon nous initialisons un ensemble ordonné $S_{decharge}$ avec t. Ensuite le sélecteur de valeur assigne à $\delta_x(t)$ la plus grande valeur positive possible et l'algorithme passe à l'étape 4.

Étape 4 : Dans cette étape, l'algorithme essaye d'étendre la décharge courante afin de limiter le nombre de décharges. Le sélecteur de variable choisit donc la variable qui n'est pas encore instanciée associée à la période t' située à une extrémité de $S_{decharge}$ (t' est soit $min(S_{decharge}) - 1$ soit $max(S_{decharge}) + 1$) ayant le plus grand signal de prix. Si aucune variable ne peut être instanciée alors l'algorithme retourne à l'étape 1. Sinon nous ajoutons t' à $S_{decharge}$. Ensuite le sélecteur de valeur assigne à $\delta_x(t')$ la plus grande valeur positive possible et l'algorithme refait de nouveau cette étape.

4.7.4 Résultats

Nous avons testé notre approche en utilisant le solveur de programmation par contraintes Choco 2.1.5 sur un Intel®Xeon®X5670. Les tests ont été réalisés sur les ensembles d'instance S_2 (50 PSS et 50 PAS), S_{3a} (70 instances de PSS difficiles pour lesquelles nous connaissons l'optimum) et S_{3b} (30 instances de PSS difficiles pour lesquelles nous ne connaissons pas l'optimum).

Le tableau 4.9 contient le nombre d'instances résolues en moins d'une seconde, une minute et 5 minutes respectivement pour les trois ensembles d'instances avec la stratégie par défaut de Choco et notre stratégie de branchement. Avec la stratégie par défaut, le solveur de programmation par contraintes résout en moins d'une minute 87 des 100 premières instances et aucune des instances difficiles (ensembles S_{3a} et S_{3b}). Les résultats sont meilleurs avec notre stratégie : 91 des 100 premières instances et 9 des instances difficiles sont résolues en moins d'une minute.

Ces résultats sont confirmés par les tableaux 4.10, 4.11 et 4.12 qui contiennent respectivement pour les instances S_2, S_{3a} et S_{3b}, pour chaque stratégie de branchement, l'écart entre la solution obtenue respectivement après 1 seconde, 1 minute et 5 minutes et la solution optimale (ou la meilleure borne connue). Avec notre stratégie, l'écart moyen après 1 minute est respectivement de 0,32 % pour les instances S_2, de 4,68 % pour les instances S_{3a}, et de 8,43 % pour les instances S_{3b}, contre respectivement 1%, 48,79 % et 66,33 % pour la stratégie par défaut de Choco.

Ces tableaux montrent ainsi que le modèle avec notre stratégie de branchement trouve rapidement une bonne solution, mais qu'il a des difficultés pour l'améliorer. En effet, 87 des 100 instances standards sont résolues avec la nouvelle stratégie de branchement après 1 seconde, 91 après 1 minute, et encore 91 après 5 minutes. De plus 6 instances difficiles sont résolues après 1 seconde et seulement 9 instances sont résolues au bout de 5 minutes. Par ailleurs, comme nous pouvons le remarquer dans les tableaux 4.11 et 4.12, l'écart moyen est seulement amélioré de quelques dixièmes de pourcents entre 1 minute et 5 minutes : il diminue de 0,32% à 0,29% pour les premières instances, de 4,68 % à 4,67 % pour les 70 instances dont l'optimum est connu et de 8,43% à 7,77 % pour les 30 instances dont l'optimum n'est pas connu.

Comme pour la PLNE nous avons remarqué que les instances les plus difficiles à résoudre sont celles pour lesquelles le nombre maximal d'appels sur l'horizon $\mathcal{T}$ est grand.

Les résultats de la programmation par contraintes montrent que bien que tous les cas ne soient pas résolus en moins d'une minute, ce modèle de programmation par contraintes, combiné à une bonne stratégie de branchement, produit rapidement (en moins de 1 seconde dans la majorité des cas) de très bonnes solutions.

4.8 Comparaison des différentes approches

4.8.1 Comparaison PLNE avec programmation par contraintes (PPC)

Nous allons dans cette section comparer les meilleures versions des deux méthodes présentées précédemment.

Le tableau 4.13 contient l'écart des solutions obtenues par les meilleurs modèles de PLNE et de PPC par rapport à l'optimum ou à la meilleure borne supérieure connue sur les deux ensembles d'instances difficiles (S_{3a} et S_{3b}) après une seconde de résolution. Ce tableau montre que les premières solutions trouvées par la PPC sont meilleures que celles trouvées par le PLNE. En effet, l'écart moyen à une seconde avec la PPC est respectivement de 6,48 % pour les instances de S_{3a} et de 9,47 % pour les instances de S_{3b}, contre respectivement 10,67% et 13,36 % pour la PLNE.

Le tableau 4.14 contient l'écart des solutions obtenues par les mêmes modèles après une minute de résolution. Nous constatons que la PLNE arrive mieux au cours du temps à améliorer les solutions que la PPC. En effet, avec la PLNE, l'écart moyen passe de 9,47% pour une seconde de résolution à 0.03% pour une minute de résolution pour les instances de S_{3a}, et de 13,36% à 5,19% pour les instances de S_{3b}, soit une amélioration de 9,43% et de 8,17% respectivement. Pour la PPC, l'écart passe de 6,48% à 4,68% pour les

TABLE 4.9 – Nombre d'instances résolues optimalement par le modèle de programmation par contraintes

	Stratégie de branchement	1 s	1 min	5 min
S_2	Stratégie par défaut	80	87	90
	Notre stratégie	87	91	91
S_{3a}	Stratégie par défaut	0	0	0
	Notre stratégie	6	9	9
S_{3b}	Stratégie par défaut	0	0	0
	Notre stratégie	0	0	0

TABLE 4.10 – Écart des solutions obtenues par le modèle de programmation par contraintes par rapport à l'optimum ou à la meilleure borne connue pour les instances S_2

Temps de résolution	Stratégie de branchement	moyenne	min	max	95ème centile
1 s	Stratégie par défaut	10,89 %	0,00 %	97,37 %	12,75 %
	Notre stratégie	3,30 %	0,00 %	100 %	3,45 %
1 min	Stratégie par défaut	1,00 %	0,00 %	29,08 %	7,04 %
	Notre stratégie	0,32 %	0,00 %	10,51 %	1,71 %
5 min	Stratégie par défaut	0,43 %	0,00 %	14,13 %	1,22 %
	Notre stratégie	0,29 %	0,00 %	10,51 %	1,32 %

TABLE 4.11 – Écart des solutions obtenues par le modèle de programmation par contraintes par rapport à l'optimum ou à la meilleure borne connue pour les instances S_{3a}

Temps de résolution	Stratégie de branchement	moyenne	min	max	95ème centile
1 s	Stratégie par défaut	78,82 %	17,33 %	100 %	100 %
	Notre stratégie	6,48%	0,00 %	100 %	20,93 %
1 min	Stratégie par défaut	48,79 %	0,16 %	100 %	97,91 %
	Notre stratégie	4,68%	0,00 %	46,40 %	18,15 %
5 min	Stratégie par défaut	44,79 %	0,14 %	100 %	91,28 %
	Notre stratégie	4,67 %	0,00 %	46,40 %	18,15 %

TABLE 4.12 – Écart des solutions obtenues par le modèle de programmation par contraintes par rapport à l'optimum ou à la meilleure borne connue pour les instances S_{3b}

1 s	Stratégie par défaut	85,35 %	58,81 %	98,36 %	97,02 %
	Notre stratégie	10,67 %	0,25 %	100 %	38,63 %
1 min	Stratégie par défaut	66,33 %	0,26 %	95 %	91,02 %
	Notre stratégie	8,43 %	0,21 %	44,47 %	38,63 %
5 min	Stratégie par défaut	63,16 %	0,12 %	94,98 %	90,74 %
	Notre stratégie	7,77 %	0,20 %	44,47 %	38,62 %

instances de S_{3a} et de 10,67% à 8,43% pour les instances de S_{3b}, soit une amélioration de seulement 1,8% et 2,24% respectivement.

Nous avons alors tenté de lancer la solveur de PLNE à partir de la solution fournie par la PPC. Les tableaux 4.15 et 4.16 contiennent les écarts moyens des solutions obtenues au bout de 10 secondes et une minute respectivement par rapport à l'optimum ou à la meilleure borne inférieure connue pour les ensembles d'instances S_{3a} et S_{3b}. Le tableau 4.15 montre que la combinaison des deux approches est meilleure que la PLNE seule : l'écart moyen est en effet de 0,04 % contre 0,10 % pour la PLNE seule sur les instances S_{3a}, et de 5,64 % contre 6,17 % sur les instances S_{3b}. Dans le tableau 4.16 nous constatons par contre que l'écart moyen est quasiment similaire au bout d'une minute d'exécution : l'écart moyen est de 0,02 % contre 0,03 % pour la PLNE seule sur les instances S_{3a}, et de 5,17 % contre 5,19 % sur les instances S_{3b}. Fournir au solveur de PLNE une bonne solution de départ permet donc de converger plus rapidement vers une meilleure solution.

4.9 Conclusion

Nous venons d'exposer plusieurs approches différentes afin de résoudre le problème de calcul du plan de production des procédés pilotés directement : un algorithme de programmation dynamique, un programme linéaire en nombres entiers, un modèle de programmation par contraintes, et une approche utilisant la solution obtenue par le modèle de programmation par contraintes comme solution initiale du programme linéaire en nombres entiers. Le tableau 4.17 résume les avantages et les inconvénients de ces approches. L'avantage principal de l'algorithme de programmation dynamique est sa convergence rapide. Mais comme nous l'avons vu précédemment, cet algorithme pseudo-polynomial très gourmand en mémoire ne permet de résoudre aucune des instances "faciles" (ensemble S_2).

L'avantage principal de la PPC est que c'est une méthode gratuite qui fournit rapidement des solutions pour toutes les instances. Cependant, cette méthode converge lentement vers l'optimum.

Les résultats obtenus par le modèle de PLNE sont meilleurs au bout d'une minute que ceux obtenus avec le modèle de PPC. Cependant cette approche nécessite un solveur payant (les gratuits n'étant pas suffisamment performants).

La meilleure approche est celle combinant la PPC et la PLNE : cette méthode est en effet celle qui fournit les meilleurs résultats pour un temps d'exécution de quelques secondes. Cependant, son utilisation requière également un logiciel payant.

Pour résumer, l'approche par programmation par contraintes est recommandée comme méthode gratuite puisqu'elle fournit rapidement une solution de bonne qualité pour toutes les instances. Pour une solution payante, nous préférerons l'approche combinant la PPC et la PLNE qui est plus performante.

Le prochain chapitre est consacré à la résolution du second problème d'optimisation qui consiste, connaissant les plans de production des procédés, à calculer le plan prévisionnel de la centrale d'agrégation.

TABLE 4.13 – Écarts moyens obtenus au bout d'une seconde par la programmation par contraintes et la programmation linéaire en nombres entiers sur les instances de S_3

Ensemble d'instances	Méthodes	moyenne	min	max	95ème centile
S_{3a}	PPC	6,48 %	0,00 %	100 %	20,93 %
	PLNE	9,47 %	0,00 %	100 %	35,82 %
S_{3b}	PPC	10,67 %	0,25%	100%	38,63 %
	PLNE	13,36 %	0,01 %	100 %	39,96%

TABLE 4.14 – Écarts moyens obtenus au bout d'une minute par la programmation par contraintes et la programmation linéaire en nombres entiers sur les instances de S_3

Ensemble d'instances	Méthodes	moyenne	min	max	95ème centile
S_{3a}	PPC	4,68 %	0,00 %	46,40 %	18,15 %
	PLNE	0,03 %	0,00 %	0,44 %	0,16 %
S_{3b}	PPC	8,43 %	0,21 %	44,47 %	38,63 %
	PLNE	5,19 %	0,01 %	44,41 %	22,90%

TABLE 4.15 – Écarts moyens obtenus au bout de 10 secondes par la programmation linéaire en nombres entiers en lui fournissant ou non la solution de la programmation par contraintes comme solution initiale sur les instances de S_3

Ensemble d'instances	Méthodes	moyenne	min	max	95ème centile
S_{3a}	PLNE	0,10 %	0,00 %	1,40 %	0,54 %
	PPC puis PLNE	0,04 %	0,00 %	0,54 %	0,19 %
S_{3b}	PLNE	6,17 %	0,21 %	59,45 %	25,73 %
	PPC puis PLNE	5,64 %	0,01 %	45,84 %	24,65 %

TABLE 4.16 – Écarts moyens obtenus au bout d'une minute par la programmation linéaire en nombres entiers en lui fournissant ou non la solution de la programmation par contraintes comme solution initiale sur les instances de S_3

Ensemble d'instances	Méthodes	moyenne	min	max	95ème centile
S_{3a}	PLNE	0,03 %	0,00 %	0,44 %	0,16 %
	PPC puis PLNE	0,02 %	0,00 %	0,43 %	0,15 %
S_{3b}	PLNE	5,19 %	0,01 %	44,41 %	22,90%
	PPC puis PLNE	5,17 %	0,01 %	44,41 %	22,85 %

TABLE 4.17 – Avantages et inconvénients des méthodes pour le problème de calcul du plan de production des procédés pilotés directement

Méthodes	Avantages	Inconvénients
Programmation dynamique	- gratuit - solution optimale trouvée très rapidement - connaissance à priori du temps de calcul et du besoin en mémoire	- inutilisable sur de nombreuses instances (limitée par la taille de la mémoire)
PPC	- gratuit - résultats corrects sur toutes les instances	- convergence lente (peu de solutions optimales trouvées en moins de 1 minute)
PLNE	- bons résultats sur toutes les instances - convergence rapide vers une bonne solution	- payant
PLNE + PPC	- bons résultats sur toutes les instances - convergence très rapide vers une bonne solution	- payant

5

Agrégation des plans de production

Ce chapitre décrit le problème d'optimisation qui consiste à calculer les puissances proposées par le coordinateur d'une centrale d'agrégation de flexibilité pour chaque pas de temps en fonction des plans de production proposés par les procédés pilotés directement et/ou les coordinateurs qu'il pilote. L'objectif est de maximiser le gain du coordinateur de la centrale.

Dans la suite du chapitre, de façon à simplifier la présentation du problème, les procédés pilotés directement et les coordinateurs pilotés seront tous appelés "systèmes pilotés" : pour chaque système piloté, le coordinateur aura le choix entre les puissances proposées et les puissances par défaut.

5.1 Description détaillée du problème

Nous considérons un horizon de temps $\mathcal{T}$ composé de T périodes, et un ensemble $\mathcal{SP}$ de systèmes pilotés. Notons $P_r^{coor}(t)$ le prix envoyé par le client du coordinateur pour le pas de temps t, soit P_{min}^{coor} et P_{max}^{coor} les puissances minimale (valeur négative ou nulle) et maximale (valeur positive ou nulle) qu'il est possible d'échanger entre le coordinateur et le client, soit P_{coor}^{pro} la puissance proposée par le coordinateur à son client au pas de temps t et soit P_{coor}^{def} la puissance par défaut du coordinateur au pas de temps t.

Pour chaque système piloté sp, les données suivantes sont supposées connues :

- $P_{def}^{sp}(t)$: puissance par défaut au pas de temps t ;
- $P_{pro}^{sp}(t)$: puissance proposée au pas de temps t ;
- $P_r^{sp}(t)$: le prix envoyé par le coordinateur au système piloté sp au pas de temps t. Il représente le prix d'achat par le coordinateur de chaque unité de puissance du système qu'il pilote. Ce prix peut être différent du prix $P_r^{coor}(t)$ envoyé par le client au coordinateur.

Afin de calculer son plan de production, le coordinateur doit, pour chaque système piloté, décider s'il accepte le plan de production proposé par ce procédé pour l'horizon $\mathcal{T}$, ou s'il conserve sa production par défaut. Il est à noter que nous considèrerons que la puissance proposée des procédés de production fatale ou de consommation fatale est égale à leur puissance par défaut (i.e. leur puissance prévisionnelle). En effet, par définition, le coordinateur ne peut contrôler la puissance de ces procédés.

À chaque pas de temps t, la puissance proposée par le coordinateur $P_{coor}^{pro}(t)$ sera donc la somme des puissances proposées par les systèmes pilotés qu'il aura sélectionnés, à laquelle s'ajoute la somme des puissances par défaut des procédés qu'il n'aura pas sélectionnés. Et à chaque pas de temps t, la puissance par défaut du coordinateur $P_{coor}^{def}(t)$ sera la somme des puissances par défaut des systèmes pilotés.

Le gain du coordinateur est la somme, sur chaque pas de temps t, du prix $P_r^{coor}(t)$ envoyé par le client multiplié par la puissance proposée par le coordinateur au pas de temps t. Le coût de production est la somme, sur chaque pas de temps t et pour chaque procédé sp, du prix $P_r^{sp}(t)$ envoyé par le coordinateur au système piloté sp, multiplié par la puissance du procédé (puissance proposée si le procédé est sélectionné, puissance par défaut sinon).

L'objectif du coordinateur est de maximiser son profit.

Voici un exemple d'instance dans laquelle un coordinateur pilote six systèmes. Le prix par kilowattheure P_r^{coor} reçu par le coordinateur est donné par la figure 5.1. Ce coordinateur pilote six systèmes différents qui ont tous reçu le même prix égal à $0,9 * P_r^{coor}$ (le coordinateur prend une marge de 10 %). Les puissances par défaut et proposées des différents procédés sont respectivement données par les figures 5.2, 5.3, 5.4, 5.5, 5.6, 5.7. Sur ces figures, la puissance par défaut des procédés est indiquée en bleu, et la puissance proposée en rouge. Nous pouvons noter que le système piloté numéro un est un procédé de déplacement de la consommation non pilotable, le numéro deux et trois sont des procédés de consommation pilotable, le cinq un procédé de déplacement de production fatale et le quatre et le six sont respectivement des procédés de consommation fatale et de production fatale (leur puissance proposée est égale à leur puissance par défaut). La puissance minimale d'échange P_{min}^{coor} est de -10 MW, et la puissance maximale P_{max}^{coor} est de 0 W.

La figure 5.8 représente la solution dans laquelle le coordinateur ne sélectionne aucun système piloté. La puissance proposée par le coordinateur est alors la somme des puissances par défaut des systèmes pilotés. Par exemple :

$$P_{coor}^{pro}(1) = P_1^{def}(1) + P_2^{def}(1) + P_3^{def}(1) + P_4^{def}(1) + P_5^{def}(1) + P_6^{def}(1) \tag{5.1}$$
$$= -2000 - 2000 - 2000 + 0 + 1750 + 0 \tag{5.2}$$
$$= -4250 \tag{5.3}$$

Nous voyons que cette solution n'est pas faisable car la contrainte de puissance maximale n'est pas respectée aux pas de temps 6, 7 et 8.

La figure 5.9 représente la solution dans laquelle le coordinateur sélectionne tous les systèmes pilotés. La puissance proposée pour le coordinateur est alors la somme des puissances proposées par les systèmes pilotés. Par exemple :

$$P_{coor}^{pro}(10) = P_1^{pro}(10) + P_2^{pro}(10) + P_3^{pro}(10) + P_4^{pro}(10) + P_5^{pro}(10) + P_6^{pro}(10) \tag{5.4}$$
$$= 0 - 1200 - 750 - 2250 + 4500 + 0 \tag{5.5}$$
$$= 300 \tag{5.6}$$

Nous voyons que cette solution n'est pas réalisable non plus car la contrainte de puissance maximale n'est pas respectée au pas de temps 10.

La figure 5.10 représente la combinaison optimale pour cet exemple. Elle consiste à sélectionner les systèmes pilotés 1, 3, 4, 5 et 6. Le système piloté 2 ne peut être retenu car la contrainte de puissance maximale ne serait pas satisfaite au pas de temps 10.

La figure 5.11 représente le plan de production associé à cette combinaison. Sur cette figure, la puissance proposée pour le coordinateur est la somme des puissances proposées par les systèmes pilotés 1, 3, 4, 5, 6 plus la puissance par défaut du système 2. Par exemple, la puissance proposée au pas de temps 10 vaut :

$$P_{coor}^{pro}(10) = P_1^{pro}(10) + P_2^{def}(10) + P_3^{pro}(10) + P_4^{pro}(10) + P_5^{pro}(10) + P_6^{pro}(10) \tag{5.7}$$
$$= -0 - 2000 - 750 - 2250 + 4500 + 0 \tag{5.8}$$
$$= -500 \tag{5.9}$$

Le gain associé à ce plan de production vaut 151125.

Comme dans les chapitres précédents, afin de simplifier les notations, nous définissons pour le coordinateur $\delta_{coor}(t)$ comme la variation de puissance du coordinateur au pas de temps t.

$$\forall t \in \mathcal{T}, \delta_{coor}(t) = P_{pro}^{coor}(t) - P_{def}^{coor}(t) \tag{5.10}$$

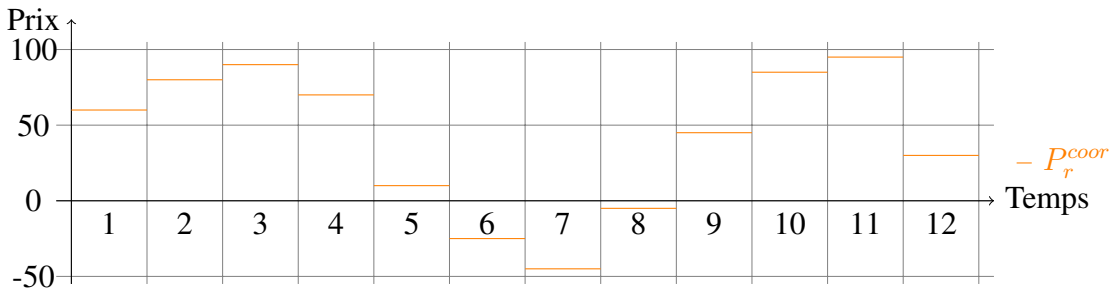

FIGURE 5.1 – Prix reçu par le coordinateur

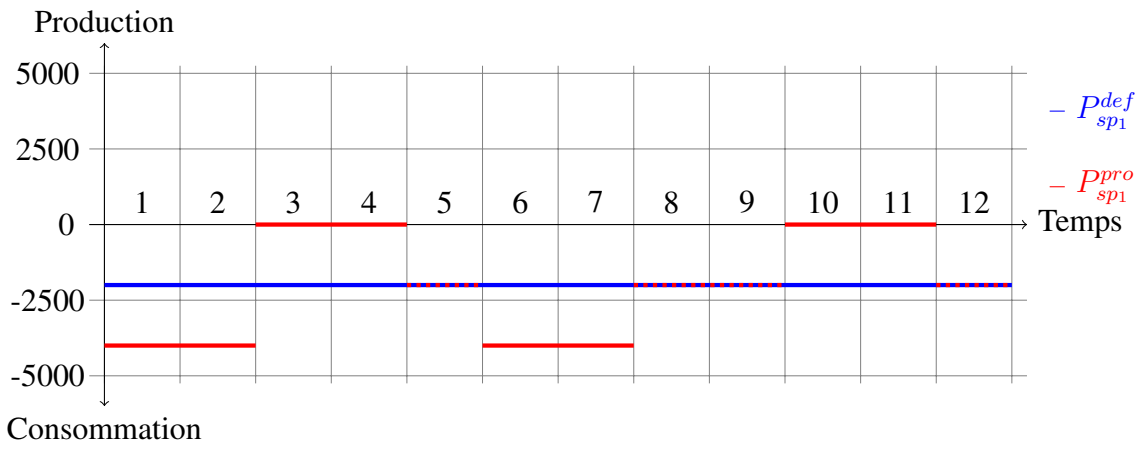

FIGURE 5.2 – Plan de production du système piloté numéro 1

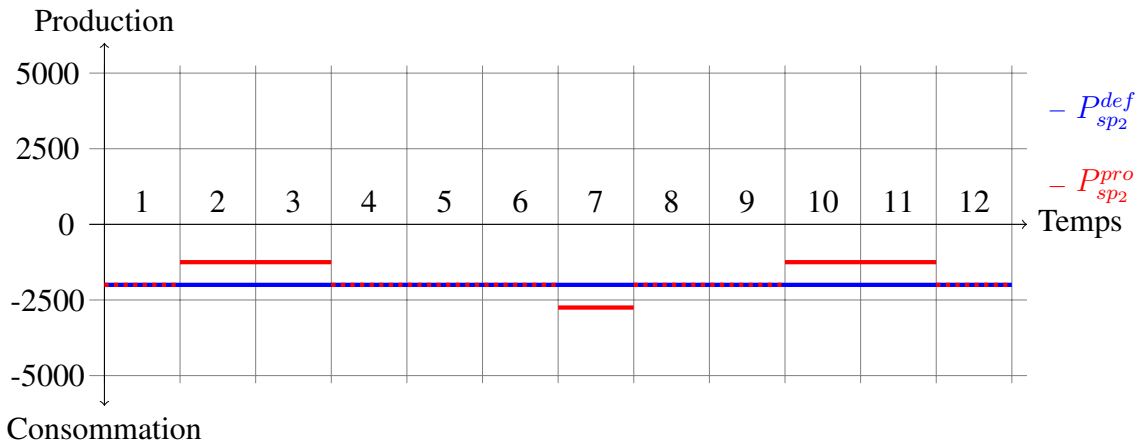

FIGURE 5.3 – Plan de production du système piloté numéro 2

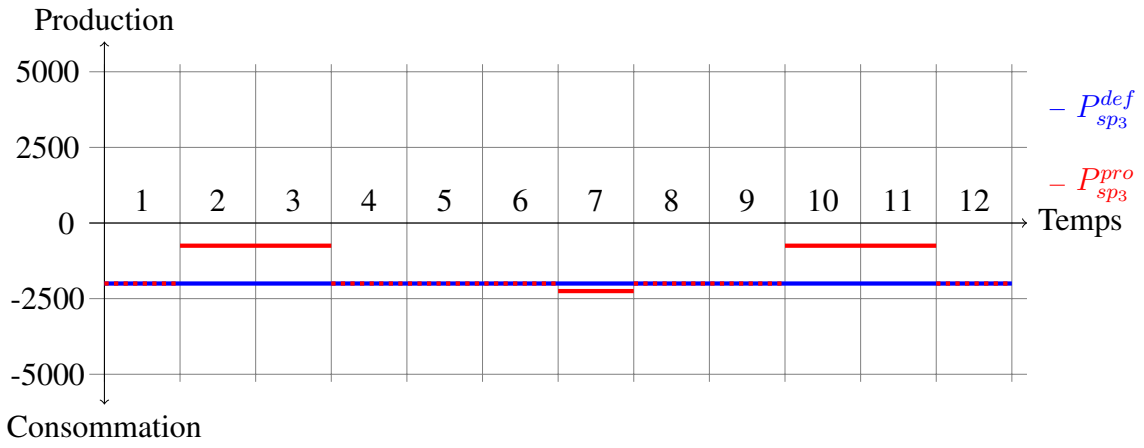

FIGURE 5.4 – Plan de production du système piloté numéro 3

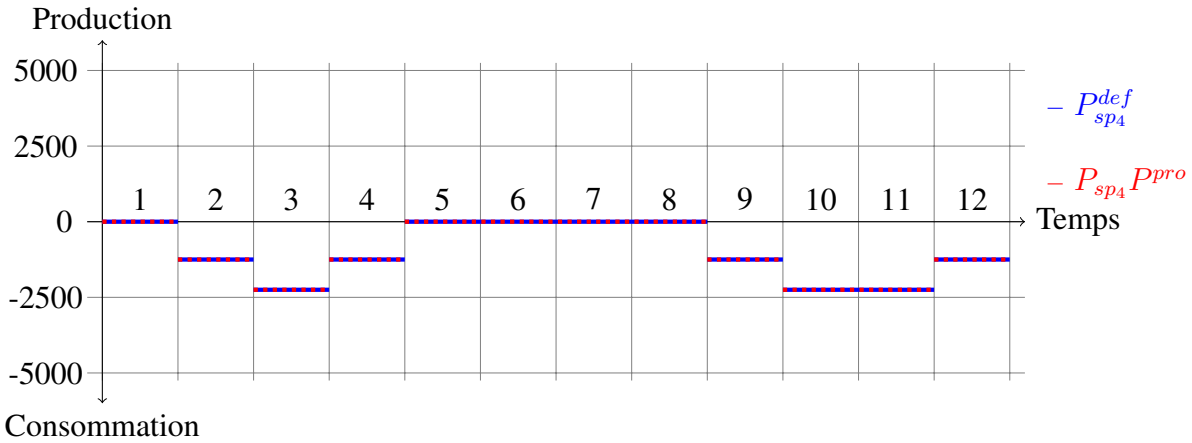

FIGURE 5.5 – Plan de production du système piloté numéro 4

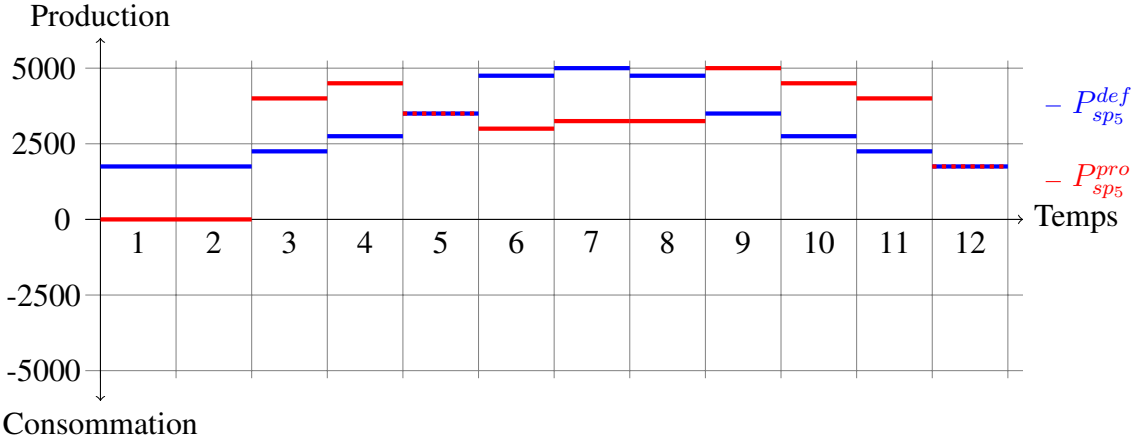

FIGURE 5.6 – Plan de production du système piloté numéro 5

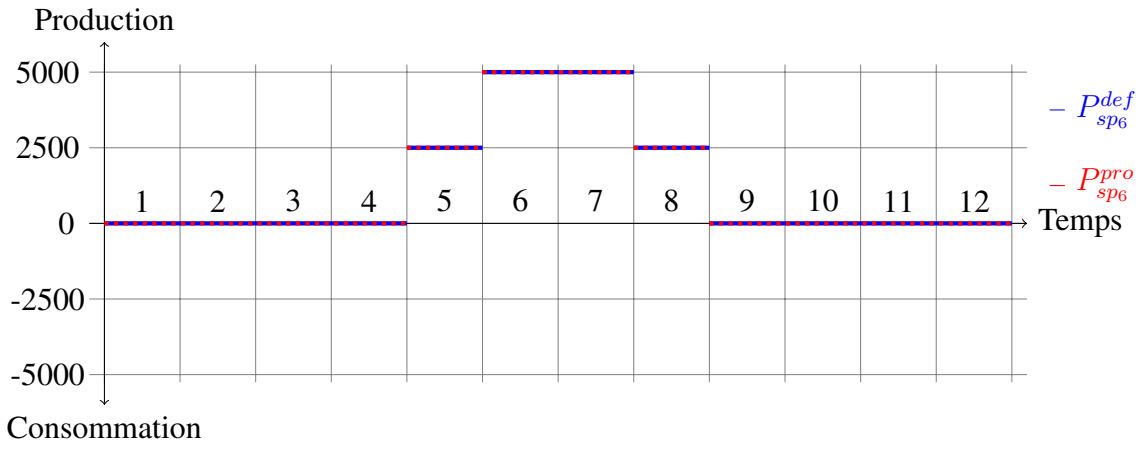

FIGURE 5.7 – Plan de production du système piloté numéro 6

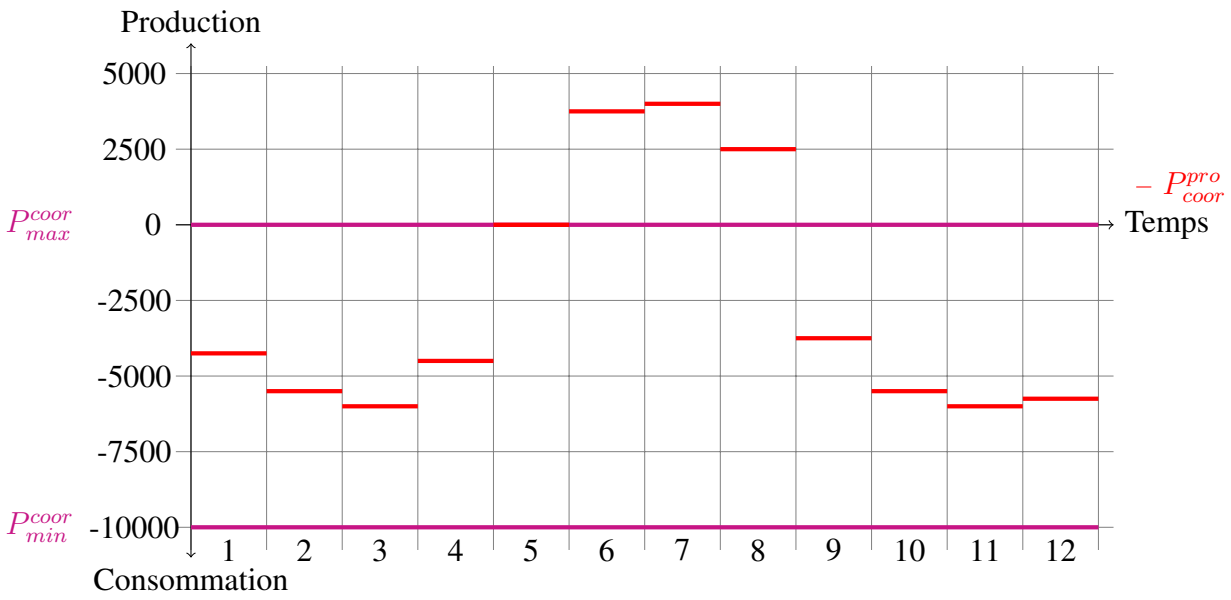

FIGURE 5.8 – Puissances proposées par le coordinateur si aucun système piloté n'est sélectionné

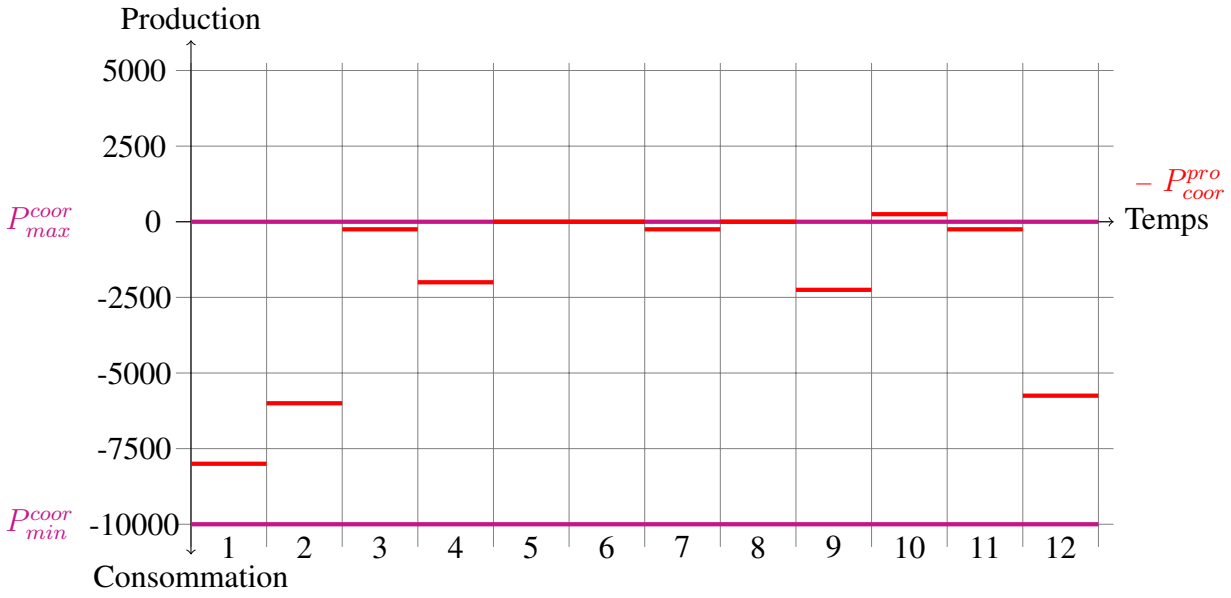

FIGURE 5.9 – Puissance proposée par le coordinateur si tous les systèmes pilotés sont sélectionnés

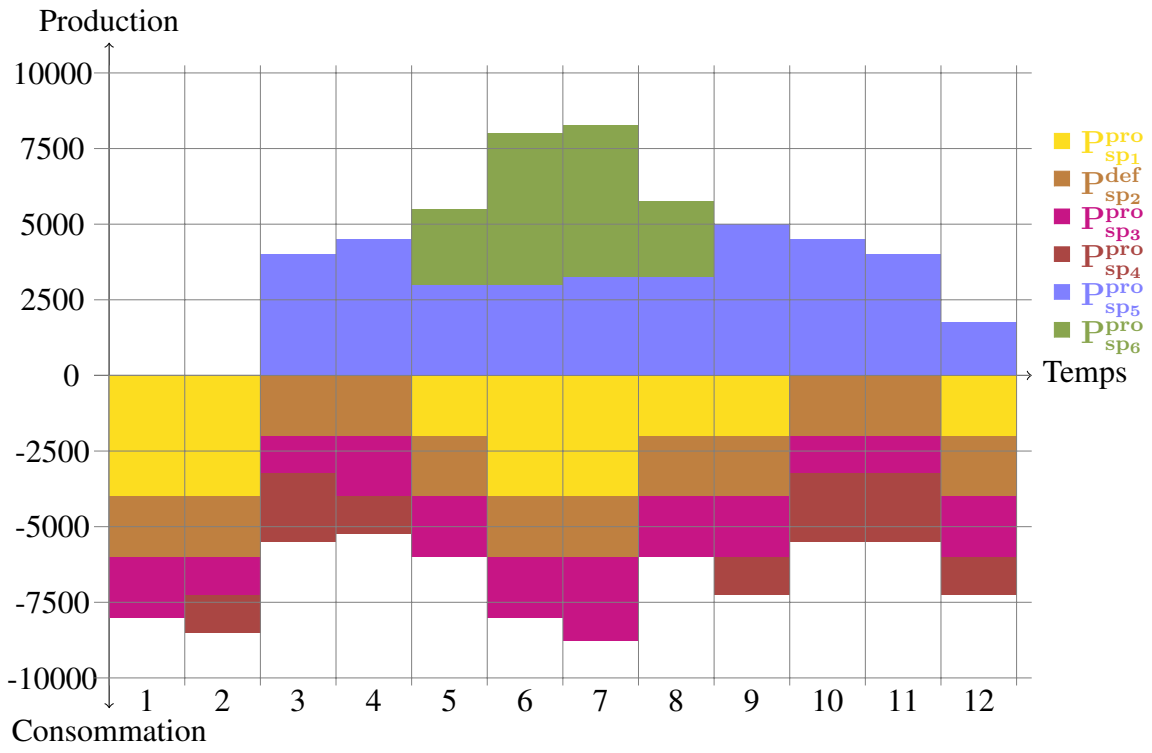

FIGURE 5.10 – Combinaison optimale des puissances

Et pour chaque système piloté sp le paramètre $\Delta_{sp}(t)$ comme la différence entre la puissance proposée par sp et sa puissance par défaut au pas de temps t :

$$\forall sp \in \mathcal{SP}, \forall t \in \mathcal{T}, \Delta_{sp}(t) = P^{sp}_{pro}(t) - P^{sp}_{def}(t) \tag{5.11}$$

De plus nous définissons la puissance minimale modifiée au pas de temps t, $\Delta_{min}(t)$ comme étant la puissance minimale d'échange du coordinateur avec le client moins la somme des puissances par défaut des systèmes pilotés :

$$\forall t \in \mathcal{T}, \Delta_{min}(t) = P^{coor}_{min} - \sum_{sp \in \mathcal{SP}} P^{sp}_{def}(t) \tag{5.12}$$

De même, nous définissons la puissance maximale modifiée au pas de temps t, $\Delta_{max}(t)$ comme étant la puissance maximale d'échange du coordinateur avec le client moins la somme des puissances par défaut des systèmes pilotés :

$$\forall t \in \mathcal{T}, \Delta_{max}(t) = P^{coor}_{max} - \sum_{sp \in \mathcal{SP}} P^{sp}_{def}(t) \tag{5.13}$$

Si nous reprenons notre exemple, à partir des puissances proposées et des puissances par défaut, nous pouvons calculer facilement la différence de puissance pour tous les systèmes pilotés et pour tous les pas de temps. Par exemple, au pas de temps 1, nous avons :

$$\begin{aligned} \Delta_{sp_1}(1) &= P^{pro}_{sp_1}(1) - P^{sp}_{def}(1) && (5.14)\\ &= -4000 - (-2000) && (5.15)\\ &= -2000 && (5.16) \end{aligned}$$

Et à partir des puissances par défaut des systèmes pilotés et des puissances minimale et maximale d'échange, nous pouvons également calculer les puissances minimale et maximale modifiées du coordinateur. Par exemple, au pas de temps 1, nous avons :

$$\begin{aligned} \Delta_{max}(1) &= P^{coor}_{max} - \left(P^{def}_1(1) + P^{def}_2(1) + P^{def}_3(1) + P^{def}_4(1) + P^{def}_5(1) + P^{def}_6(1)\right) && (5.17)\\ &= 0 - (-2000 - 2000 - 2000 + 0 + 1750 + 0) && (5.18)\\ &= 4250 && (5.19) \end{aligned}$$

et :

$$\begin{aligned} \Delta_{min}(1) &= P^{coor}_{min} - \left(P^{def}_1(1) + P^{def}_2(1) + P^{def}_3(1) + P^{def}_4(1) + P^{def}_5(1) + P^{def}_6(1)\right) && (5.20)\\ &= -10000 - (-2000 - 2000 - 2000 + 0 + 1750 + 0) && (5.21)\\ &= -5750 && (5.22) \end{aligned}$$

La figure 5.12 représente la combinaison optimale de variations de puissances pour cet exemple. Il est à noter que les systèmes 4 et 6 n'apparaissent pas sur la figure car ils sont fatals (la différence entre leur puissance proposée et leur puissance par défaut est nulle).

La figure 5.13 représente les variations de puissance du coordinateur associées à cette combinaison. Sur cette figure, la puissance proposée pour le coordinateur est la somme des variations de puissances proposées par les systèmes pilotés 1, 3 et 5. Par exemple, la variation de puissance du coordinateur δ_{coor} au pas de temps 10 vaut :

$$\begin{aligned} \delta_{coor}(10) &= \Delta^{pro}_{sp_1}(10) + \Delta^{pro}_{sp_3}(10) + \Delta^{pro}_{sp_5}(10) && (5.23)\\ &= 2000 + 1250 + 1750 && (5.24)\\ &= 5000 && (5.25) \end{aligned}$$

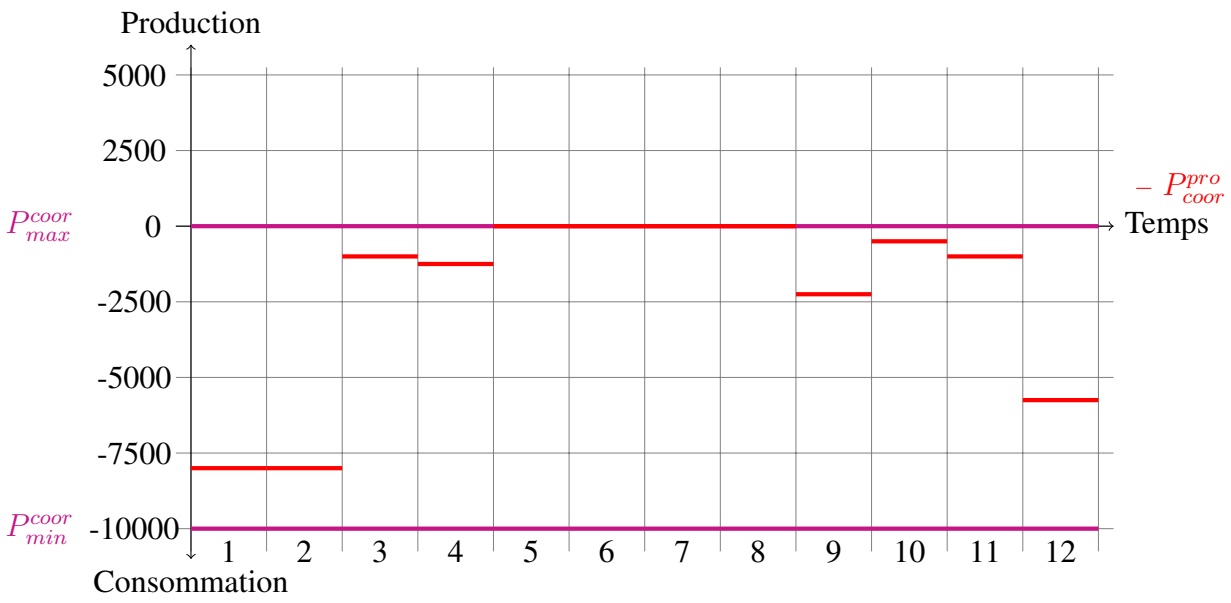

FIGURE 5.11 – Puissances proposées optimales pour le coordinateur

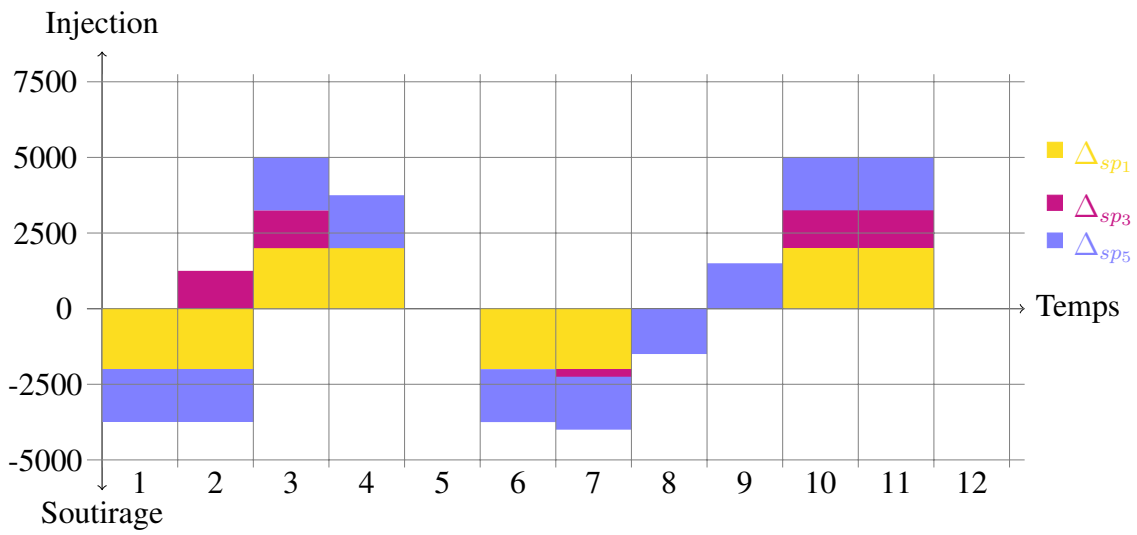

FIGURE 5.12 – Combinaison optimale des variations de puissances

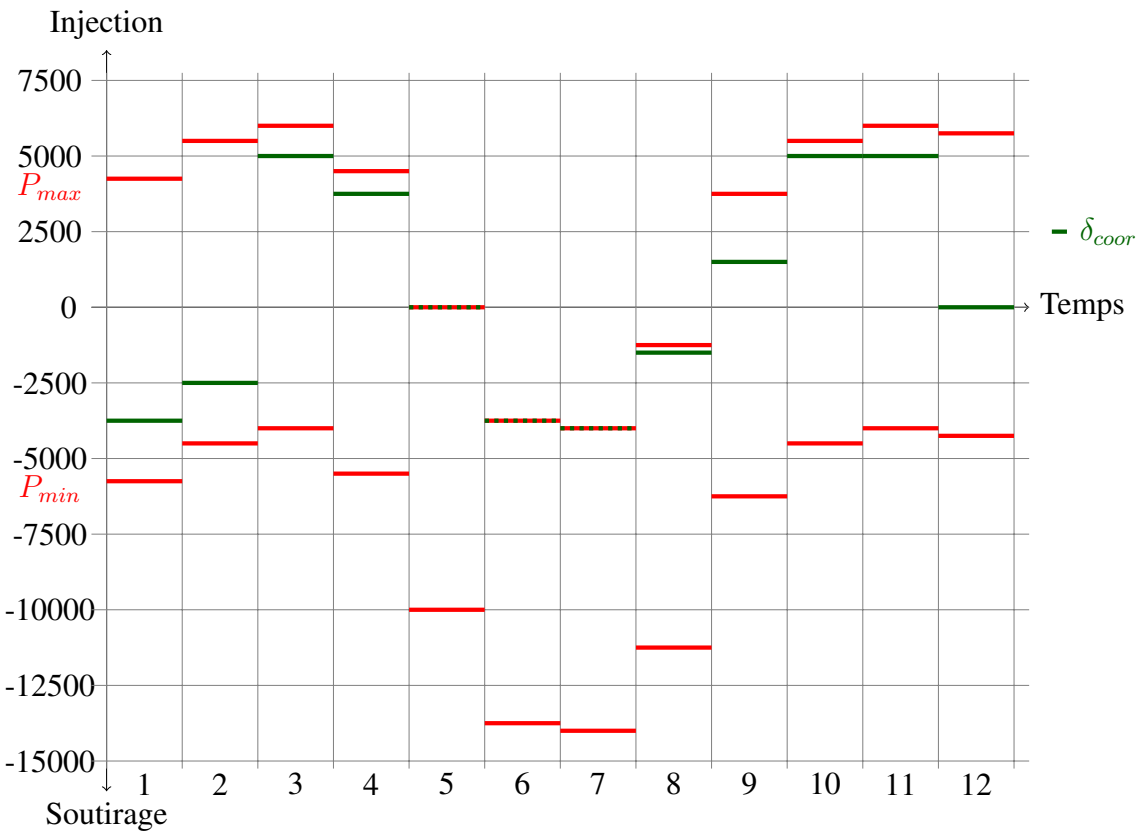

FIGURE 5.13 – Variations de puissances optimales

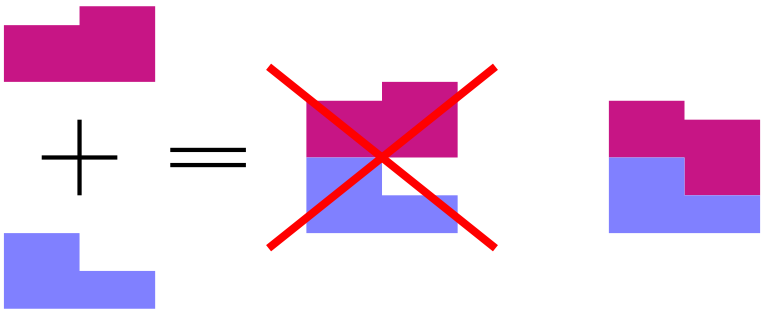

FIGURE 5.14 – Schéma illustrant la déformabilité des objets

5.2 Caractérisation scientifique et complexité

Si nous considérons que pour tous les temps t, les variations de puissances $\Delta_{sp}(t)$ des systèmes pilotés sont soit toutes positives (i.e. tous les systèmes pilotés injectent dans le réseau), soit toutes négatives (i.e. tous les systèmes pilotés soutirent depuis le réseau) alors le problème est équivalent à un problème de sac-à-dos multidimensionnel en 0-1 [Briley and McDonald, 1977]. En effet, à chaque système piloté nous pouvons associer un objet de dimension T de poids dans la dimension t $|\Delta_{sp}(t)|$ et de valeur $\sum_{t\in\mathcal{T}}(P^{sp}_{def}(t)+\Delta_{sp}(t))*(P^{coor}_r(t)-P^{sp}_r(t))$. La capacité du sac dans la dimension t est $\Delta_{max}(t)$ si tous les systèmes pilotés injectent dans le réseau ($|\Delta_{min}(t)|$ dans l'autre cas). L'objectif est de maximiser la somme des valeurs des objets emportés, en respectant la capacité du sac-à-dos dans chacune des dimensions. Notre problème général est donc une extension du problème de sac-à-dos multidimensionnel en 0-1 dans lequel le poids d'un objet peut être positive ou négative. De plus le poids d'un objet peut être de signe différent à deux périodes différentes, par exemple une batterie a une puissance négative quand elle se charge et une puissance positive quand elle se décharge. Et enfin les objets sont déformables (cf. Figure 5.14).

Le problème de sac-à-dos en 0-1 multidimensionnel étant NP-Difficile (ce problème est une extension du problème de sac-à-dos en 0-1 qui est NP-Difficile [Martello and Toth, 1990]), notre problème d'agrégation de plans de production l'est donc aussi.

Nous pouvons noter que notre problème ressemble au problème d'ordonnancement sous contraintes cumulatives [Garey and Johnson, 1979] qui consiste à ordonnancer un ensemble de tâches utilisant des ressources en respectant la capacité de ces ressources. La différence avec notre problème est que dans ce problème d'ordonnancement, toutes les tâches doivent être effectuées alors que dans notre problème d'agrégation, nous avons le choix de sélectionner ou non une tâche. De plus, dans le problème d'ordonnancement sous contraintes cumulatives, la consommation de ressource des tâches est positive ou nulle, alors que dans notre problème les consommations peuvent être négatives (puisque les procédés peuvent avoir des puissances négatives).

À notre connaissance, le problème d'agrégation que nous avons à résoudre n'a donc jamais été étudié dans la littérature.

5.3 Instances

Pour tester les méthodes que nous présentons par la suite, nous avons généré un ensemble de 100 instances.

Dans ces instances, l'horizon de temps $\mathcal{T}$ est fixé à 24 heures avec un pas temporel de 10 minutes (comme dans les instances du problème traité dans le chapitre précédent).

Dans un premier temps, nous avons créé 100 instances, contenant chacune un nombre de systèmes pilotés choisi aléatoirement entre 400 et 800 (dans la réalité, le nombre de systèmes pilotés ne dépasse pas 400). Ces systèmes pilotés ont été sélectionnés aléatoirement parmi les 1000 systèmes pilotés générés sur la base de données réelles dans le chapitre 4 de ce rapport, et leurs plans de production correspondent aux plans de production calculés par les méthodes présentées dans ce même chapitre.

La puissance P_{coor}^{min} est choisie aléatoirement entre $\sum_{sp\in\mathcal{SP}} \min(P_{pro}^{sp}(t), P_{def}^{sp}(t), 0)$ et 0. La puissance P_{coor}^{max} est choisie aléatoirement entre 0 et $\sum_{sp\in\mathcal{SP}} \max(P_{pro}^{sp}(t), P_{def}^{sp}(t), 0)$.

Nous avons ensuite associé à chaque instance un prix P_r^{coor} (prix envoyé par le client au coordinateur) choisi aléatoirement parmi l'historique des prix du marché spot de l'année 2011. Supposons que le prix $P_r^{sp}(t)$ envoyé à chaque système piloté sp vaut $0,9 * P_r^{coor}(t)$ (le coordinateur prend une marge de 10 %).

Afin de simuler les procédés fatals, nous avons commencé par choisir aléatoirement pour chaque instance une puissance P_{fatal} entre $\sum_{sp\in\mathcal{SP}} \min(P_{pro}^{sp}(t), P_{def}^{sp}(t), 0)$ et $\sum_{sp\in\mathcal{SP}} \max(P_{pro}^{sp}(t), P_{def}^{sp}(t), 0)$. Puis nous avons fait varier cette puissance fatal au cours de la journée. Pour cela, nous avons commencer par générer un profil de consommation pour chaque jour de l'année 2011 pour la France sur la base des historiques de la façon suivante : pour chaque jour j de l'année 2011, pour chaque période t, nous avons normalisé la puissance obtenant ainsi un profil $Prof_{ref}^j(t) = \frac{P_{conso}^j(t)}{\max_t(P_{conso}^j(t))}$ avec $P_{conso}^j(t)$ la puissance consommée au jour j à la période t. Nous avons ensuite choisi aléatoirement un profil $Prof$ parmi l'ensemble des profils $\{Prof_{ref}^j\}$. Nous avons donc ajouté à chaque instance un procédé sp_{fatal} suivant :

- $P_{def}^{sp}(t) = P_{fatal} * Prof(t)$: une puissance par défaut au pas de temps t ;
- $P_{pro}^{sp}(t) = P_{def}^{sp}(t)$: une puissance proposée au pas de temps t ;
- Le prix envoyé à ce système piloté est le même que pour les autres systèmes pilotés.

Nous avons donc 100 instances générées aléatoirement afin de tester nos méthodes.

5.4 Programmation dynamique

Le problème d'agrégation de plans de production étant une extension du problème de sac-à-dos en 0-1, nous nous sommes inspirés des méthodes développées pour ce dernier problème. Nous nous sommes dans un premier temps intéressés à l'adaptation à notre problème de l'algorithme de programmation dynamique proposé dans [Bellman, 1958].

Nous verrons tout d'abord comment l'adapter au problème d'agrégation noté AP_T dans lequel les systèmes pilotés ne peuvent qu'injecter dans le réseau, ainsi qu'au problème AN_T dans lequel les systèmes pilotés ne peuvent que soutirer. Puis nous présenterons sa généralisation au problème d'agrégation général que nous noterons A_T.

5.4.1 Adaptation à AP_T

5.4.1.1 Premier algorithme

Pour rappel, l'algorithme de programmation dynamique du problème de sac-à-dos en 0-1 [Bellman, 1954] est le suivant :

Pour chaque couple d'entiers m $(1 \leq m \leq n)$ et $\hat{w}$ $(0 \leq \hat{w} \leq W)$, considérons la sous-instance du problème de sac à dos contenant les m premiers objets et de capacité $\hat{w}$. Nous noterons $f_m(\hat{w})$ la solution optimale de cette sous-instance :

$$f_m(\hat{w}) = \max\left\{ \sum_{j=1}^{m} p_j x_j : \sum_{j=1}^{m} w_j x_j \leq \hat{w}.\ x_j \in \{0,1\}\ \forall j \in [1;m] \right\}$$

.

1. Trivialement, nous avons $f_1(\hat{w}) = \begin{cases} 0\ \forall \hat{w} \in [0, w_1 - 1] \\ p_1\ \forall \hat{w} \in [w_1, W] \end{cases}$.
2. Puis récursivement, nous avons $f_m(\hat{w}) = \begin{cases} f_{m-1}(\hat{w})\ \forall \hat{w} \in [0, w_m - 1] \\ \max(f_{m-1}(\hat{w}); f_{m-1}(\hat{w} - w_m) + p_m)\ \forall \hat{w} \in [w_m, W] \end{cases}$.

La solution optimale du problème de sac-à-dos en 0-1 complet correspond à $f_n(W)$.

Une fois la table des $f_m(\hat{w})$ construite, il suffit de démarrer de la case de $f_n(W)$ et de déduire l'état des objets en remontant jusqu'à une case $f_0(*)$.

La complexité temporelle et spatiale de cette approche est $O(nW)$.

Nous allons adapter cet algorithme à notre problème AP_T :

Pour chaque $T+1$-uples d'entiers m ($1 \leq m \leq |SP|$, p_1 ($0 \leq p_1 \leq \Delta_{max}(1)$), ... , p_T ($0 \leq p_T \leq \Delta_{max}(T)$), considérons la sous-instance du problème de AP_T avec les m premiers systèmes pilotés et la liste des capacités égale à $(p_1, \ldots, p_T)$. Nous noterons $f_m(p_1, ..., p_t, ..., p_T)$ la solution optimale de cette sous-instance :

$f_m(p_1, ..., p_T) = \max\left\{ \sum_{t\in\mathcal{T}} \left(\sum_{sp\in\mathcal{SP}} (\alpha_{sp} * (P_r^{coor}(t) - P_r^{sp}(t)) * \Delta_{sp}(t) \right) : 0 \leq \sum_{sp\in\mathcal{SP}} (\alpha_{sp} * \Delta_{sp}(t) \leq p_t, \forall t \in \mathcal{T} \,.\, \alpha_{sp} \in \{0,1\}\ \forall sp \in \mathcal{SP} \right\}$.

1. Trivialement, nous avons $f_1(p_1, ..., p_T) = \begin{cases} 0 \text{ Si } \exists t \in \mathcal{T} \text{ tel que } p_t < \Delta_{P_{sp_1}}(t) \\ \sum_{t\in\mathcal{T}} (P_r^{coor}(t) - P_r^{sp_1}(t)) * \Delta_{P_{sp_1}(t)}) \text{ Sinon.} \end{cases}$.

2. Nous pouvons définir $f_m(p_1, ..., p_T)$ récursivement ainsi :

   - Si $\exists t \in [|1; T|]$ tel que $p_t < \Delta_{sp_m}(t)$ alors $f_m(p_1, ..., p_T) = f_{m-1}(p_1, ..., p_T)$.
   - Sinon $f_m(p_1, ..., p_T) = \max(f_{m-1}(p_1, ..., p_T); f_{m-1}(p_1 - \Delta_{sp_m}(1), ..., p_T - \Delta_{sp_m}(T)) + \sum_{t\in\mathcal{T}}((P_r^{coor}(t) - P_r^{sp_m}(t)) * \Delta_{sp_m}(t)))$

La solution optimale correspond alors à $f_{|SP|}(\Delta_{max}(1), ..., \Delta_{max}(T))$. Il est alors possible de retrouver les procédés sélectionnés de la même façon que pour le problème de sac-à-dos en 0-1.

La complexité temporelle et spatiale de cette solution est $O\left(|SP| * \prod_{t\in\mathcal{T}}(\Delta_{max}(t))\right)$. Il existe donc un algorithme pseudo-polynomial pour résoudre AP_T. Par conséquent ce problème est NP-Difficile au sens faible.

Dans le cas du problèmes AN_T, l'algorithme est similaire mais il faut prendre les valeurs absolues des $\Delta_{sp}(t)$ et remplacer tous les $\Delta_{max}(t)$ par les $\Delta_{min}(t)$. Il est donc aussi NP-Difficile au sens faible.

5.4.1.2 Élimination des $T+1$-uplets dominés

Dans un problème de sac-à-dos en 0-1, le nombre d'états considérés à chaque étape peut être considérablement réduit en éliminant les couples $(m, \hat{w})$ dominés, c'est à dire les couples pour lesquels il existe un couple (m, ω) avec $f_m(\omega) \geq f_m(\hat{w})$ et $\omega < \hat{w}$ (le sac à dos est moins rempli pour un profit supérieur ou égal) [Martello and Toth, 1990].

Voici comment ces couples sont éliminés : soit $\mathcal{F}_m$, l'ensemble des valeurs des états dominants possédant jusqu'à m objets. Nous considérons que $\mathcal{F}_m$ est ordonné.

Trivialement, nous avons : $\begin{cases} \mathcal{F}_1 = \{f_1(0); f_1(w_1)\} \\ f_1(0) = 0 \\ f_1(w_1) = p_1 \end{cases}$.

Puis récursivement, nous déterminons $\mathcal{F}_m$ grâce à l'algorithme suivant :

Pour chaque $f_{m-1}(\hat{w})$ dans $\mathcal{F}_{m-1}$:

- Si $f_m(\hat{w})$ n'existe pas dans $\mathcal{F}_m$, alors nous ajoutons $f_{m-1}(\hat{w})$ dans $\mathcal{F}_m$.
- Sinon $\begin{cases} \text{Si} f_{m-1}(\hat{w} + w_m) \text{ existe dans } \mathcal{F}_{m-1}, \text{ alors nous ajoutons à } \mathcal{F}_m \text{ la valeur} \\ \qquad f_m(\hat{w} + w_m) = \max\{f_{m-1}(\hat{w} + w_m), f_{m-1}(\hat{w}) + p_m\}. \\ \text{Sinon Si } \hat{w} + w_m \leq W, \text{ alors nous ajoutons dans } \mathcal{F}_m, f_m(\hat{w} + w_m) = f_{m-1}(\hat{w}) + p_m. \end{cases}$.

La solution optimale du problème de sac-à-dos en 0-1 complet correspond à la dernière valeur dans $\mathcal{F}_n$. La complexité temporelle et spatiale de cette version est en $O\Big(min(2^n, nW)\Big)$: l'algorithme en $O(2^n)$ testant toutes les combinaisons possibles est limité parce qu'on ne peut pas avoir plus de W états de sac-à-dos différents lors de l'ajout d'un objet.

Voici comment ceci peut être adapté à notre problème AP_T en reprenant les mêmes notations :

1. Trivialement, nous avons : $\begin{cases} \mathcal{F}_1 = \{f_1(0, ..., 0); f_1(\Delta_{P_{sp_1}(1)}, ..., \Delta_{P_{sp_1}(T)})\} \\ f_1(0, ..., 0) = 0 \\ f_1(\Delta_{P_1(1)}, ..., \Delta_{P_1(T)}) = \sum_{t\in\mathcal{T}}(P_r^{coor}(t) - P_r^{sp_m}(t)) * \Delta_{P_{sp_1}(t)}) \end{cases}$.

2. Puis récursivement, nous déterminons $\mathcal{F}_m$ grâce à l'algorithme suivant :

   Pour chaque $f_{m-1}(p_1, ..., p_T)$ dans $\mathcal{F}_{m-1}$:

   - Si $f_m(p_1, ..., p_T)$ n'existe pas dans $\mathcal{F}_m$, alors nous ajoutons $f_{m-1}(p_1, ..., p_T)$ dans $\mathcal{F}_m$.
   - Sinon $\begin{cases} \text{Si } f_{m-1}(p_1 + \Delta_{sp_m}(1), ..., p_T + \Delta_{sp_m}(T)) \text{ est dans } \mathcal{F}_{m-1} \text{ alors nous ajoutons à } \mathcal{F}_m \\ \quad \text{la valeur } f_m(p_1 + \Delta_{sp_m}(1), ..., p_T + \Delta_{sp_m}(T)) \text{ qui est égale au} \\ \quad \text{maximum entre } f_{m-1}(p_1 + \Delta_{sp_m}(1), ..., p_T + \Delta_{sp_m}(T)) \text{ et} \\ \quad f_{m-1}(p_1, ..., p_T) + \sum_{t\in\mathcal{T}}((P_r^{coor}(t) - P_r^{sp_m}(t)) * \Delta_{sp_m}(t)). \\ \text{Sinon si } \forall t \in \mathcal{T}, p_t + \Delta_{sp_m}(t) \leq \Delta_{max}(t) \text{ alors nous ajoutons à } \mathcal{F}_m \text{ la valeur} \\ \quad f_m(p_1 + \Delta_{sp_m}(1), ..., p_T + \Delta_{sp_m}(T)) \\ \quad = f_{m-1}(p_1, ..., p_T) + \sum_{t\in\mathcal{T}}((P_r^{coor}(t) - P_r^{sp_m}(t)) * \Delta_{sp_m}(t)). \end{cases}$.

La solution optimale du problème AP_T complet correspond à la plus grande valeur dans $\mathcal{F}_{|SP|}$. La complexité temporelle et spatiale de cette solution est alors en $O\Big(min(2^{|SP|}, |SP| * \prod_{t\in\mathcal{T}}(\Delta_{max}(t)))\Big)$.

5.4.1.3 Un approche type "Meet-in-the-middle" pour le cas particulier AP_T

Horowitz and Sahni (1974) ont présenté un algorithme pour résoudre le problème du sac-à-dos en 0-1 basé sur la subdivision du problème original de n objets en deux sous-problèmes respectivement de $q = \lceil \frac{n}{2} \rceil$ et de $r = n - q$ objets. Pour chaque sous-problème, une liste contenant tous les couples dominants est construite. Les deux listes sont combinées pour trouver la solution optimale. La complexité temporelle et spatiale de cet algorithme est en $O\Big(min(2^{\frac{n}{2}}, nW)\Big)$.

Nous pouvons utiliser la même technique pour le problème AP_T. La complexité temporelle et spatiale de cette solution est alors en $O\Big(min(2^{\frac{|SP|}{2}}, |SP| * \prod_{t\in\mathcal{T}} \Delta_{max}(t))\Big)$.

5.4.2 Algorithme pour A_T

Voici maintenant une adaptation de l'algorithme au cas général du problème dans lequel les puissances peuvent être positives ou négatives. Nous allons reprendre le même principe que l'algorithme pour AP_T. La sous-instance du problème de AP_T avec les m premiers systèmes pilotés et une liste de capacités égale à $(p_1, \ldots, p_T)$, sera caractérisé par un $T + 1$-uples d'entiers m $(1 \leq m \leq |SP|)$, p_1 $(\Delta_{min}(1) - \sum_{sp\in[m+1,|SP|]} \max(\Delta_{sp}(1), 0) \leq p_1 \leq \Delta_{max}(1) - \sum_{sp\in[m+1,|SP|]} \min(\Delta_{sp}(1), 0))$, $\ldots$, p_T $(\Delta_{min}(T) - \sum_{sp\in[m+1,|SP|]} \max(\Delta_{sp}(T), 0) \leq p_T \leq \Delta_{max}(T) - \sum_{sp\in[m+1,|SP|]} \min(\Delta_{sp}(T), 0))$, la sous-instance du problème de A_T avec les m premiers systèmes pilotés et une liste de capacités égale à $(p_1, \ldots, p_T)$. En effet, lors de l'ajout du m^{me} système piloté, la capacité du pas de temps t est comprise entre la limite inférieure du sac-à-dos $\Delta_{min}(t)$ moins la somme sur les systèmes pilotés restants des variations de puissance positives $\sum_{sp\in[m+1,|SP|]} \max(\Delta_{sp}(t), 0)$ et la limite supérieure du sac-à-dos $\Delta_{max}(t)$ moins la somme sur les systèmes pilotés restants des variations de puissance négatives $\sum_{sp\in[m+1,|SP|]} \min(\Delta_{sp}(t), 0)$.

Nous noterons $f_m(p_1, ..., p_t, ..., p_T)$ la solution optimale de cette sous-instance :

$f_m(p_1, ..., p_T) = \max \Bigg\{ \sum_{t\in\mathcal{T}} \Bigg(\sum_{sp\in\mathcal{SP}} (\alpha_{sp} * (S_{coor}(t) - S_{sp}(t)) * \Delta_{sp}(t) \Bigg) : \Delta_{min}(t) - \sum_{sp\in[m+1,|SP|]} (\max(\Delta_{sp}(t), 0) \leq \sum_{sp\in\mathcal{SP}} (\alpha_{sp} * \Delta_{sp}(t) \leq \Delta_{max}(t) - \sum_{sp\in[m+1,|SP|]} (\min(\Delta_{sp}(t), 0), \forall t \in \mathcal{T} \,.\, \alpha_{sp} \in \{0, 1\} \, \forall sp \in \mathcal{SP} \Bigg\}$.

Dans cet algorithme, comme les puissances peuvent être positives et négatives, la "hauteur" au pas de temps t du "sac-à-dos" intermédiaire m devra être comprise entre $\Delta_{min}(t)$ moins la somme des variations de puissance positives des objets restants $\sum_{sp\in\{m+1,|SP|\}} (\max(\Delta_{sp}(t), 0)$ et $\Delta_{max}(t)$ moins la somme des variations de puissance négatives des objets restants $\sum_{sp\in\mathcal{SP}} (\min(\Delta_{sp}(T), 0)$. Par exemple pour l'instance de A_3 suivante :

- $\Delta_{max} = \{10; 20; 5\}$;
- $\Delta_{min} = \{-5; -10; -7\}$;
- $\Delta_1 = \{5; 0; 5\}$;
- $\Delta_2 = \{-7; 2; 6\}$;
- $\Delta_3 = \{-10; -7; 7\}$;
- $\Delta_4 = \{20; -5; -10\}$.

La "hauteur" du "sac-à-dos" intermédiaire 2 devra être comprise :

- au temps 1, entre $\Delta_{min}(1) - \Delta_4(1) = -5 - 20 = -25$ et $\Delta_{max}(1) - \Delta_3(1) = 10 - (-10) = 20$;
- au temps 2, entre $\Delta_{min}(2) = -10$ et $\Delta_{max}(2) - (\Delta_3(2) + \Delta_4(2)) = 20 - (-7 - 5) = 32$;
- au temps 3, entre $\Delta_{min}(3) - \Delta_3(3) = -7 - 7 = -14$ et $\Delta_{max}(1) - \Delta_4(3) = 5 - (-10) = 15$.

Notre algorithme sera donc le suivant :

1. Trivialement, nous avons :
   $f_1(p_1, ..., p_T) = \begin{cases} \sum_{t\in\mathcal{T}} (S_{coor}(t) - S_{sp_1}(t)) * \Delta_{P_{sp_1}(t)}) \text{ si } \forall t \in \mathcal{T}, p_t \geq \Delta_{P_{sp_1}}(t). \\ 0 \text{ sinon.} \end{cases}$
2. Nous pouvons définir pour tous les $f_{m-1}(p_1, ..., p_T)$ qui existent les $f_m(p_1, ..., p_T)$ suivants :
   - $f_m(0, ..., 0) = 0$;
   - $f_m(p_1, ..., p_T) = f_{m-1}(p_1, ..., p_T)$;
   - Si $\forall t \in \mathcal{T}, \Delta_{min}(t) - \sum_{sp\in\{m+1,|SP|\}} \max(\Delta_{sp}(T), 0) \leq p_t + \Delta_{sp_m}(t) \leq \Delta_{max}(t) - \sum_{sp\in[m+1,|SP|]} (\min(\Delta_{sp}(t), 0) \sum_{sp\in\{m+1,|SP|\}} \min(\Delta_{sp}(T), 0)$ alors $f_m(p_1 + \Delta_{sp_m}(1), ..., p_T + \Delta_{sp_m}(T)) = f_m(p_1, ..., p_T) + \sum_{t\in\mathcal{T}} ((S_{coor}(t) - S_{sp_m}(t)) * \Delta_{sp_m}(t))$.

La solution optimale correspond alors à la plus grande valeur de $f_{|SP|}(p_1, ..., p_T)$. La complexité temporelle et spatiale de cet algorithme est en :

$O\Bigg(min(2^{|SP|}, |SP| * \prod_{t\in\mathcal{T}} (\Delta_{max}(t) - \sum_{sp\in\mathcal{SP}} (\min(\Delta_{sp}(t), 0)) - (\Delta_{min}(t) - \sum_{sp\in\mathcal{SP}} (\max(\Delta_{sp}(t), 0))) \Bigg)$.

Puisqu'il existe un algorithme pseudo-polynomial pour résoudre A_T, A_T est NP-Difficile au sens faible.

5.4.3 Résultats

Nous avons testé ces algorithmes sur les instances que nous avons générées aléatoirement. Les calculs ont été réalisés sur un ordinateur avec un processeur Intel®Xeon®X5670 et une carte graphique NVIDIA®GeForce ®GTX 780. Aucune des instances, qui, nous le rappelons, sont composées de 400 à 800 procédés, ne peut être résolue avec cette approche. En effet, le nombre d'états ($> 10^120$) est trop grand pour notre puissance de calcul. Nous avons généré d'autres instances plus petites afin de déterminer la taille approximative des problèmes pouvant être résolus par cette approche : nous avons pu résoudre des problèmes jusqu'à 32 procédés. Nous sommes donc bien loin des 350 procédés des problèmes de taille réelle, même en utilisant l'approche type "Meet-in-the-middle" pour le cas particulier AP_T. Cette approche n'est donc pas utilisable.

Nous proposons d'autres approches dans les sections suivantes.

5.5 Programmation linéaire en nombres entiers

5.5.1 Formulation mathématique

Nous allons dans cette section nous intéresser à la modélisation du problème par un programme linéaire en nombres entiers.

Pour tout système piloté sp nous introduisons les variables suivantes :

$$\alpha_{sp} = \begin{cases} 1 \text{ si le coordinateur sélectionne la puissance proposée par le système piloté } sp. \\ 0 \text{ si le coordinateur décide de conserver la puissance par défaut du système piloté } sp. \end{cases}$$

La seule contrainte du problème est que la puissance échangée entre le coordinateur et son client est bornée inférieurement et supérieurement à chaque instant t :

$$\forall t \in \mathcal{T}, \Delta_{min}(t) \leq \sum_{sp \in \mathcal{SP}} (\alpha_{sp} * \Delta_{sp}(t)) \leq \Delta_{max}(t) \tag{5.26}$$

L'objectif est de maximiser le bénéfice du coordinateur. Ce bénéfice correspond au gain provenant de la vente au client de la puissance fournie par le coordinateur (le client achète à chaque instant t chaque unité de puissance au prix $P_r^{coor}(t)$), moins le coût d'achat par le coordinateur des puissances des systèmes qu'il pilote (il achète chaque unité d'un système piloté sp au prix $P_r^{sp}(t)$)) :

$$\max \sum_{sp \in \mathcal{SP}} \sum_{t \in \mathcal{T}} \alpha_{sp} * \Delta_{sp}(t) * (P_r^{coor}(t) - P_r^{sp}(t)) \tag{5.27}$$

En résumé, le problème se modélise donc de la façon suivante :

$$\begin{array}{lll} \text{maximize} & \sum_{sp \in \mathcal{SP}} \sum_{t \in \mathcal{T}} \alpha_{sp} * \Delta_{sp}(t) * (P_r^{coor}(t) - P_r^{sp}(t)) & \\ \text{subject to} & \Delta_{min}(t) \leq \sum_{sp \in \mathcal{SP}} \alpha_{sp} * \Delta_{P_{sp}}(t) \leq \Delta_{max}(t) & \forall t \in \mathcal{T} \\ \text{and} & \alpha_{sp} \in \{0, 1\} & \forall sp \in \mathcal{SP} \end{array}$$

5.5.2 Résultats

Le modèle a été testé sur un processeur Intel®Xeon®X5670 en utilisant différents solveurs du marché : des solveurs payants (CPLEX, XPRESS-MP et GUROBI) et des solveurs gratuits (LPSolve, GLPK et Coin-OR). Les solveurs payants donnaient des résultats semblables et les résultats des les solveurs gratuits étaient équivalents. Nous avons donc choisi de faire nos tests en utilisant CPLEX 12.2 comme solveur payant, et LPSolve 5.5 comme solveur gratuit. Le tableau 5.1 contient les résultats obtenus sur les 100 instances. La partie supérieure du tableau contient le nombre d'instances résolues en fonction du temps d'exécution pour chacun des solveurs. CPLEX résout optimalement 26 instances en moins d'une minute contre 0 pour

TABLE 5.1 – Résultats de la programmation linéaire en nombres entiers sur les 100 instances

Nombre d'instances pour lesquelles le temps de résolution est				
Méthodes	$< 1\,min$	$1\,min \leq < 5\,min$	$5\,min \leq < 30\,min$	$> 30\,min$
LPSolve	0	0	0	100
CPLEX	26	9	14	51
Gap au bout d'une minute				
Méthodes	moyenne	min	max	95ème centile
LPSolve	8,88 %	0,08 %	26,63 %	20,49 %
CPLEX	0,20 %	0,00 %	0,91 %	0,70 %

LPSolve. Nous avons comparé les solutions trouvées par la méthode arrêtée au bout d'une minute avec l'optimum (trouvé en laissant tourner CPLEX plusieurs heures). L'écart entre les solutions trouvées au bout d'une minute par chacun des solveurs et l'optimum figure dans la partie inférieure du tableau. Les résultats montrent que les solutions obtenues par CPLEX après une minute sont très satisfaisantes (écart moyen de 0,20 % avec un écart minimal à 0 % et un écart maximal à 0,91 %) alors que le écart moyen obtenu par LPSolve est de 8,88 % avec un écart minimal à 0,01 % et un écart maximal à 26,63 %.

Les résultats de la programmation linéaire en nombres entiers sont satisfaisants pour le logiciel payant (CPLEX), mais ils ne le sont pas pour le logiciel gratuit (LPSolve). Cette approche est donc adaptée à la résolution de problèmes réels, mais elle n'est pas utilisable si nous ne voulons pas utiliser de logiciel payant.

Dans le but de disposer d'une méthode gratuite performante, nous présentons dans la section suivante l'adaptation d'une approche heuristique initialement développée pour résoudre le problème de sac-à-dos en 0-1 : une méthode de Monte-Carlo sur les chaînes de Markov.

5.6 Méthode de Monte Carlo sur les chaines de Markov

Afin de disposer d'une approche gratuite, nous avons cherché à développer une méthode approchée en nous inspirant des méthodes développées pour le problème de sac-à-dos multidimensionnel en 0-1 regroupées au sein de cette étude [Fréville, 2004]. Diverses méthodes approchées ont été proposées pour ce problème : des algorithmes génétiques, des méthodes de Monte Carlo sur les chaines de Markov, des particle swarm optimization, ... La méthode de Monte-Carlo sur les chaines de Markov ayant déjà été utilisée de façon satisfaisante chez EDF pour résoudre par exemple le problème de recomposition de la courbe de charge d'un consommateur [El Guedri *et al.*, 2009], nous avons décidé de tester cette approche.

5.6.1 Méthode de Métropolis

Nous avons commencé par adapter la méthode proposée initialement par [Metropolis *et al.*, 1953] pour le problème de sac-à-dos en 0-1. Voici le principe de cette méthode pour notre problème :

- soit $A_k = (x_1^k, \ldots, x_N^k)$: la solution courante à l'itération k avec $x_i^k = 1$ si le procédé i est sélectionné dans la solution courante à l'itération k, 0 sinon, et soit $P_{A_k}(t)$ la puissance totale de la centrale d'agrégation au pas de temps t à l'itération k ;

- à partir de A_k, nous définissons ainsi A_{k+1} :

  1. nous choisissons j aléatoirement selon une loi uniforme dans l'intervalle $[1, |SP|]$;
  2. soit $Y = (x_1^k, \ldots, x_{j-1}^k, 1 - x_j^k, x_{j+1}^k, \ldots x_N^k)$;

3. si Y est une solution réalisable, (*i.e.* $\forall t \in \mathcal{T} \mid P_{A_k}(t)+(1-x_j^k)*\Delta_{P_{sp_j}}(t) \geq \Delta_{min}(t)\ et\ \Delta_{max}(t) \geq P_{A_k}(t) + (1 - x_j^k) * \Delta_{P_{sp_j}}(t)$), alors $A_{k+1} = Y$ sinon $A_{k+1} = A_k$.

- l'algorithme s'arrête au bout de K_{max} itérations et retourne le meilleur A_k.

5.6.2 Améliorations

Nous avons apporté différentes améliorations à cet algorithme : nous avons tout d'abord adapté celles proposées par Hastings pour la méthode de Métropolis, puis nous avons modifié la loi de probabilité permettant de sélectionner le procédé j à faire entrer ou sortir de la solution à chaque itération, et enfin nous avons élargi le voisinage de recherche.

5.6.2.1 Hastings

[Hastings, 1970] a montré que l'algorithme de Métropolis converge lentement. Il l'a donc amélioré en y ajoutant un probabilité d'acceptation. Ainsi, dans l'étape 3, si la solution Y est réalisable, elle a une probabilité β d'être acceptée. Pour définir β pour le problème de sac-à-dos en 0-1, Hastings propose d'utiliser le principe du recuit simulé. Ainsi, $\beta = \min(1, C * \exp(\log(k) * \sum_{k=1}^{N} V_k * (x_i^Y - x_i^k))$ où :

- k est le numéro de l'itération courante ;
- C est une constante ;
- N est le nombre d'objets ;
- V_k est la valeur de l'objet k.

Nous avons adapté β à notre problème en le définissant de la façon suivante :

$$\beta = \min(1, C * \exp(\log(k) * \sum_{sp \in \mathcal{SP}} \sum_{t=1}^{T} (P_r^{sp}(t) - P_r^{sp}(t)) * (x_{sp}^Y - x_{sp}^k) * \Delta_{P_{sp}}(t))$$

5.6.2.2 Pénalisation

Afin d'améliorer l'algorithme, nous avons également essayé de modifier la loi de probabilité permettant de sélectionner le procédé à faire entrer ou sortir de la solution à l'étape 1 :

- en pénalisant l'inversion d'objet qui dégrade la solution ;
- en favorisant l'inversion d'objet qui améliore la solution.

Soit G_j le gain de l'inversion de l'objet j : $G_j = \sum_{t=1}^{T} (P_r^{sp}(t) - SP_{sp_j}(t)) * (1 - \alpha_j) * \Delta_{P_{sp_j}}(t)$. Dans l'algorithme *Pénalisé*, pour choisir le procédé j à faire sortir ou entrer dans la solution, nous proposons d'utiliser la loi suivante $\mathcal{P}, \forall j \in \mathcal{SP}$:

- si $G_j > 0 : P(j) = \frac{1}{|SP|} * (1 + \frac{G_j}{\sum_{j|G_j>0} G_j})$;
- si $G_j = 0 : P(j) = \frac{1}{|SP|}$;
- si $G_j < 0 : P(j) = \frac{1}{|SP|} * (1 + \frac{G_j}{\sum_{j|G_j<0} |G_j|})$.

5.6.2.3 Élargir le voisinage de recherche

Dans cette version de base, l'algorithme utilise un voisinage de recherche limité (nous ne faisons sortir ou entrer qu'un objet à la fois). De plus, nous avons remarqué que l'algorithme rejetait beaucoup de solutions non réalisables. Nous avons donc proposé deux autres voisinages :

- d'une part un voisinage permettant soit de faire sortir ou entrer un objet j de la solution, soit de faire sortir un objet j_{in} de la solution et de faire entrer un objet j_{out} qui n'était pas dans la solution. Ainsi à chaque itération, le voisinage utilisé est sélectionné aléatoirement.
- d'autre part un voisinage autorisant les solutions non réalisables. L'idée est d'autoriser le passage par des solutions infaisables en espérant atteindre une meilleure solution réalisable par la suite. Il est à noter que nous utilisons pour les solutions irréalisables la même probabilité d'acceptation que pour les solutions réalisables.

5.6.3 Résultats

Nous avons testé les différentes versions de cet algorithme sur un ordinateur Intel®Xeon®X5670. Le tableau 5.2 montre les résultats obtenus sur les 100 instances. Ce tableau contient la moyenne sur toutes les instances des écarts par rapport à l'optimum pour la version de base (*Métropolis*), la version de base améliorée incluant l'amélioration de *Hastings*, puis cette dernière version avec la nouvelle loi de sélection des procédés (*Pénalisé*), puis la version contenant ces deux premières adaptations avec en plus l'utilisation du voisinage autorisant l'échange de deux procédés (*Échange*), et pour terminer la version contenant toutes les améliorations (*AvecSolInfaisable*).

L'écart moyen obtenu par l'algorithme de base *Metropolis* est de 9,53 % avec un minimum de 7,58 % et un maximum de 32,78 %, alors que la version contenant toutes les améliorations permet d'obtenir un écart moyen de 0,86 % avec un minimum de 0,00 %, un maximum de 8,85 %. Chaque amélioration proposée permet de réduire l'écart des solutions par rapport à l'optimum. Cependant, ce sont les modifications de voisinages qui permettent les plus grandes réductions de cet écart.

Le tableau 5.3 contient la moyenne sur toutes les instances des écarts par rapport à l'optimum pour la PLNE (avec Cplex) et pour la meilleure version de la méthode de Monte-Carlo sur les chaines de Markov (*MCMC*). Ce tableau montre que les résultats obtenus par *MCMC* sont comparables à ceux de Cplex (0,86 % en moyenne contre 0,20 %).

Étant donné que *MCMC* utilise un tirage aléatoire, nous avons exécuté 100 fois cet algorithme sur chaque instance afin de nous assurer de la stabilité de la méthode. La moyenne des écarts moyens obtenus par *MCMC* est de 0,96 % avec un minimum à 0,80 % et un maximum à 1,50%. Afin de limiter ces effets indésirables, nous avons décidé de lancer un algorithme *MCMC* par cœur disponible sur la machine et de conserver la meilleure solution obtenue par ces différentes exécutions.

Les résultats de cette approche sont donc satisfaisants.

5.7 Impact du signal de prix sur la solution

Si nous observons attentivement la solution optimale de l'exemple donné précédemment (cf. Figure 5.11), nous constatons que les contraintes de puissances minimale et maximale ne sont saturées qu'aux pas de temps 5, 6 et 7. Il serait ainsi possible d'augmenter le gain en produisant plus pendant les pas de temps dont le signal de prix est positif, et en consommant plus pendant les pas de temps dont le signal de prix est négatif. Ainsi, dans une solution idéale, nous souhaiterions injecter un maximum (et si possible saturer la contrainte P_{max}^{coor}) pendant les périodes de temps ayant un signal de prix positif, et consommer un maximum (et si possible saturer la contrainte P_{min}^{coor}) pendant les périodes de temps ayant un signal de prix négatif. Il faudrait, pour atteindre une telle solution à partir de la solution de la Figure 5.11, d'une part que tous les procédés ne soient pas sélectionnés dans la solution courante, et d'autre part que les procédés modifient

leurs puissances proposées. Compte tenu du mode de fonctionnement retenu, la modification des puissances proposées par les procédés n'est possible qu'en modifiant le signal de prix qui leur est envoyé.

Dans la solution de la figure 5.11, le procédé numéro 2 n'est pas sélectionné. Supposons que le prix qui lui est envoyé soit celui donné par la figure 5.15. Alors les puissances proposées par ce procédé deviennent celles données la figure 5.16. Dans ce cas, le coordinateur sélectionnera tous les procédés, aboutissant aux puissances proposées de la figure 5.17 de gain 173625, donc supérieur de 15 % à celui de la solution optimale précédente de la figure 5.11.

Nous avons comparé le gain des solutions optimales trouvées par Cplex au gain de la solution (pas forcément réalisable) dans laquelle à chaque pas de temps, la puissance proposée par la centrale est égale au maximum entre P_{min}^{coor} et la somme des puissances minimales P_{min}^{sp} des procédés sp qui soutirent si le prix est négatif, et est égale au minimum entre P_{coor}^{max} et la somme des puissances maximales P_{max}^{sp} des procédés sp qui injectent si le signal de prix est positif. En moyenne, l'écart entre ces deux solutions est de 13,66 %, avec un écart minimal de 0,68 % et un écart maximal de 42,59 %. Ces résultats montrent que les solutions obtenues pourraient certainement être améliorées. Nous pensons donc qu'il serait intéressant de travailler sur l'élaboration des signaux de prix. Une première proposition d'algorithme qui pourrait être utilisé pour modifier les signaux de prix est donnée dans l'annexe B.

5.8 Conclusion

Nous venons d'exposer trois approches différentes (programmation dynamique, programmation linéaire en nombres entiers et méthode de Monte Carlo sur les chaines de Markov) afin de résoudre notre problème. Le tableau 5.4 résume les avantages et les inconvénients de ces approches. L'avantage principal de la programmation dynamique est sa convergence rapide mais cette méthode ne peut pas aggréger suffisamment de procédés pour être utilisée sur des problèmes réels. La meilleure méthode est la PLNE avec un solveur payant (Cplex). L'avantage principal de la méthode de Monte Carlo sur les chaines de Markov est que c'est une méthode gratuite qui est utilisable sur toutes les instances. Mais cette méthode est non déterministe et les résultats sont un peu moins bons que pour la PLNE. La programmation linéaire en nombres entiers a comme avantage de converger rapidement vers l'optimum. Cependant, son utilisation requière un logiciel payant.

TABLE 5.2 – Résultats de la méthode de Monte-Carlo sur les chaines de Markov sur les 100 instances

Gap au bout d'une minute				
Méthodes	moyenne	min	max	95ème centile
Metropolis	9,53 %	7,58 %	32,78 %	25,56 %
+Hastings	8,08 %	6,08 %	30,85 %	24,47 %
+Pénalisé	6,85 %	3,67 %	30,04 %	22,35 %
+ Échange	3,54 %	0,00%	10,45 %	9,68 %
+AvecSolInfaisable	0,86 %	0,00 %	8,85 %	7,56 %

TABLE 5.3 – Comparaison de la programmation linéaire en nombres entiers et la méthode de Monte-Carlo sur les chaines de Markov sur les 100 instances

Gap au bout d'une minute				
CPLEX	0,20 %	0,00 %	0,91 %	0,70 %
MCMC	0,86 %	0,00 %	8,85 %	7,56 %

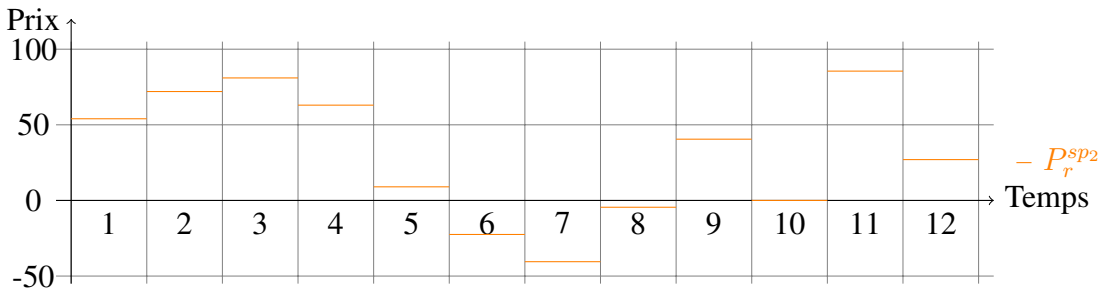

FIGURE 5.15 – Prix modifié pour le système piloté numéro 2

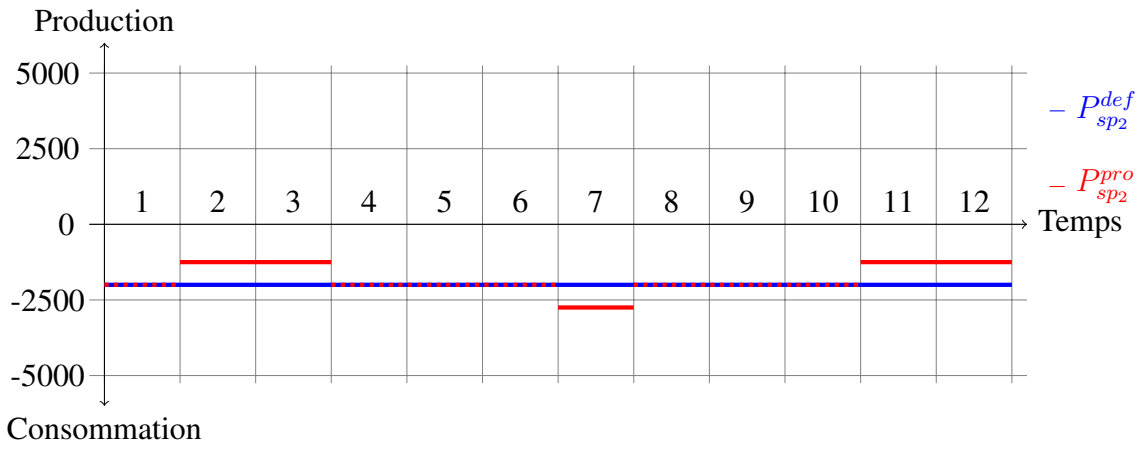

FIGURE 5.16 – Plan de production modifié du système piloté numéro 2

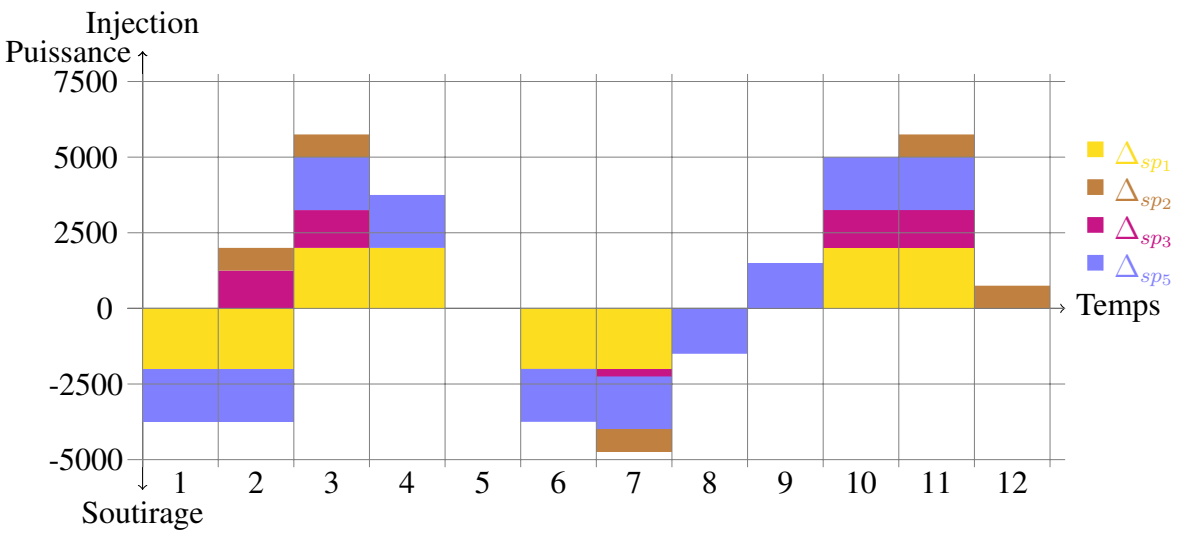

FIGURE 5.17 – Nouvelle combinaison optimale des variations de puissance

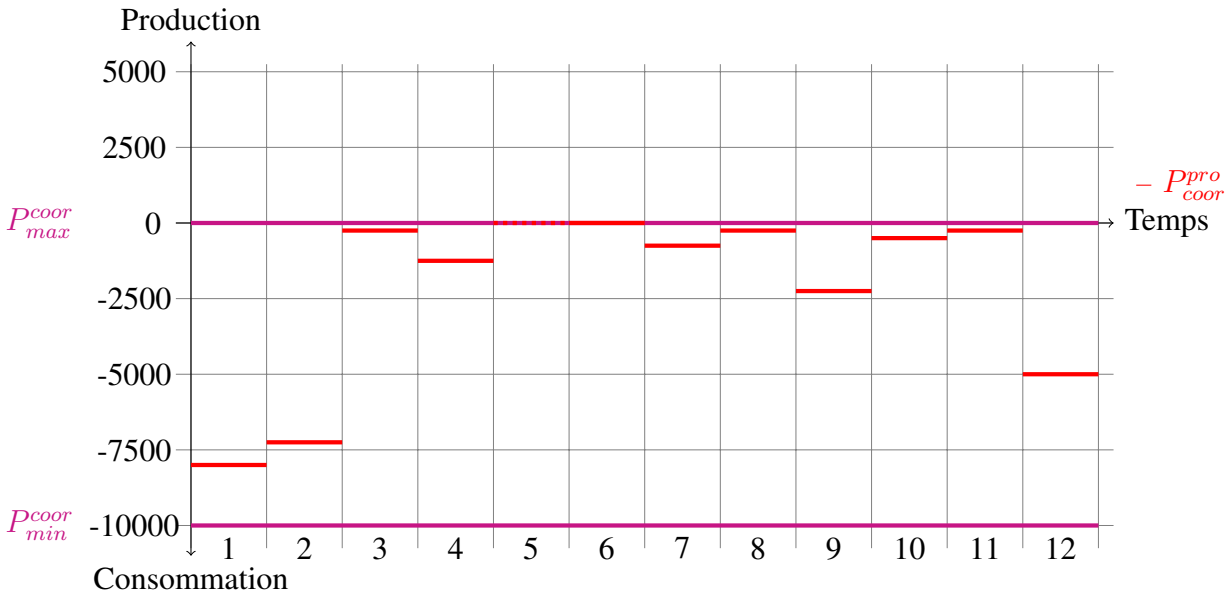

FIGURE 5.18 – Nouvellles puissances proposées par le coordinateur

TABLE 5.4 – Avantages et inconvénients des méthodes pour le problème d'agrégation des plans de production

Méthodes	Avantages	Inconvénients
Programmation dynamique	- gratuit - connaissance a priori du temps de calcul et du besoin en mémoire - convergence très rapide	- inutilisable sur des problèmes de taille réelle (limitée par la taille de la mémoire)
PLNE	- utilisable sur toutes les instances - convergence très rapide	- payant - aucune connaissance a priori du temps de calcul
MCMC	- gratuit - utilisable sur toutes les instances - convergence rapide	- non déterministe - aucune connaissance a priori du temps de calcul

6

Conclusions et perspectives

6.1 Conclusions

L'objectif général de cette thèse était de proposer une solution permettant d'augmenter la part des énergies renouvelables dans le mix énergétique. Nous avons vu qu'une façon de les intégrer au réseau de distribution dans le réseau électrique actuel est d'intégrer des procédés de consommation, de production et de stockage de petite taille, diffus (ils sont répartis sur tout le territoire) et hétéroclites. Nous avons proposé de les regrouper en six catégories :

- les procédés de production pilotable (e.g. les turbines à combustion) ;
- les procédés de consommation pilotable (e.g. les systèmes de gestion de l'éclairage public qui peuvent faire varier la luminosité des DEL pilotés) ;
- les procédés de déplacement de la production fatale (e.g. un système de production solaire thermique qui peut stocker le liquide chauffé dans un ballon et qui peut ensuite produire de l'électricité à partir de cette eau via une turbine à vapeur) ;
- les procédés de déplacement de la consommation fatale (e.g. un système qui contrôle la température peut arrêter le chauffage pendant quelques heures sans impacter fortement le confort de l'utilisateur) ;
- les procédés de stockage (e.g. les batteries) ;
- les procédés de déplacement d'une consommation (e.g. un système qui contrôle une machine à laver peut modifier l'heure de départ du cycle de la machine).

Ces différents procédés sont soumis à des contraintes spécifiques, qui sont soit des contraintes techniques, soit des contraintes de confort (pour ne pas user prématurément le matériel, ou pour ne pas gêner son utilisateur). Parmi les contraintes techniques, nous pouvons citer par exemple :

- une puissance minimale et maximale fournie ;
- une puissance fournie discrète.

Les contraintes de confort sont par exemple :

- une durée minimale et maximale d'appel ;

- une durée minimale entre un arrêt et le prochain départ ;
- un nombre maximal d'appels dans une journée.

Afin d'être rattachés au réseau de transport, ces procédés peuvent être rassemblés avec les moyens de production fatale au sein d'une centrale virtuelle (ou centrale d'agrégation de flexibilité). Afin de piloter une telle centrale comme une centrale classique et afin qu'elle puisse interagir avec un responsable d'équilibre, nous avons tout d'abord proposé une architecture multi-niveaux dans laquelle un coordinateur peut piloter plusieurs procédés directement et d'autres coordinateurs. Cette architecture permet d'intégrer tous les types de procédés, y compris d'autres centrales virtuelles, de tenir compte des contraintes liées aux réseaux et de l'impact de la centrale sur la courbe de consommation locale. Le mode de gestion proposé est le suivant : de haut en bas, chaque coordinateur reçoit de son opérateur amont un signal de prix, puis envoie un signal de prix à ses coordinateurs et aux procédés qu'il pilote directement. Dans un second temps, chaque procédé va déterminer son plan de production en fonction du signal de prix reçu. Finalement, ces plans de production sont collectés par chaque coordinateur, puis agrégés niveau par niveau, afin de calculer le plan de production global. Ces étapes peuvent être répétées avec un autre signal de prix si le plan de production obtenu n'est pas satisfaisant. Ce mode de gestion retenu a l'avantage de répondre à plusieurs types d'objectifs comme celui d'augmenter ou de diminuer la production, de faire un îlotage, de lisser la courbe de charge ou de satisfaire une demande précise.

Ce mode de fonctionnement fait apparaître deux principaux problèmes d'optimisation : la détermination (en moins d'une minute) du plan de production de chaque procédé piloté de façon à respecter les contraintes du procédé et à maximiser son gain en fonction du signal de prix reçu, et la détermination du plan de production de la centrale d'agrégation à partir des plans de production des procédés qu'elle contient. Cette architecture et ce mode de gestion ont été présentés à la conférence ISGT 2014 [Prelle *et al.*, 2014b]. Dans cette thèse, nous nous sommes intéressés au cas statique (pour le processus de répartition de l'offre sur la demande de la veille pour le lendemain) mais aussi au cas dynamique (pour le mécanisme d'ajustement) des deux principaux problèmes d'optimisation.

Nous avons dans un premier temps étudié la complexité du premier problème. Nous avons montré que le problème associé aux procédés de déplacement d'une consommation est polynomial. Par contre, bien que certains cas particuliers des problèmes des autres procédés soient polynomiaux, le problème général est NP-Difficile au sens faible. Ensuite, afin de pouvoir proposer des méthodes de résolution génériques, nous avons décrit une modélisation permettant de représenter n'importe quel type de procédé. Cette modélisation nous a permis de proposer un algorithme de programmation dynamique calculant le plan de production de n'importe quel procédé. Nous avons ensuite construit un modèle mathématique basé sur un modèle développé dans un projet précédent. Ce modèle a été amélioré par des reformulations de contraintes, l'ajout de contraintes « surrogates » et des bornes supérieures sur différentes variables. Il a ensuite été résolu avec un solveur payant (Cplex). Ce modèle a aussi été présenté à la conférence ISGT 2014 [Prelle *et al.*, 2014b]. Pour terminer, nous avons développé un modèle de programmation par contraintes pour lequel nous avons proposé une stratégie de branchement appropriée à notre problème. Ce modèle est ensuite résolu avec un solveur gratuit (Choco). Ce modèle de programmation par contraintes a été présenté à la conférence ROADEF 2014 [Prelle *et al.*, 2014a].

Afin de tester ces méthodes, nous avons généré deux groupes d'instances réalistes pour ce problème : des instances « standards » et des instances plus difficiles. Les expérimentations montrent que la méthode de programmation dynamique est utilisable pour certaines instances, mais qu'elle ne permet pas de résoudre toutes les instances pour raison d'occupation mémoire. Bien que le modèle de programmation linéaire en nombres entiers ne permette pas de résoudre optimalement toutes les instances difficiles en moins d'une minute, cette approche permet d'obtenir des solutions réalisables de bonne qualité dans le temps imparti. Concernant l'approche de programmation par contraintes, elle permet d'obtenir très vite (quelques secondes) une bonne solution pour toutes les instances, par contre la méthode peine à améliorer cette solution dans le temps imparti. La combinaison des deux dernières approches (calcul d'une première solution par programmation par contraintes, puis amélioration de cette solution par l'approche par programmation

linéaire en nombres entiers) permet d'obtenir de très bons résultats. C'est cette combinaison qui est préconisée si l'utilisation d'un solveur payant est autorisée.

Une fois les plans de production des procédés calculés, nous nous sommes attaqués au problème de calcul du plan de production de la centrale d'agrégation. Nous avons tout d'abord montré que ce problème est une extension d'un problème de sac-à-dos bidimensionnel en 0-1 dans lequel, entre autre, la hauteur des objets peut-être positive ou négative, et que ce problème est donc NP-Difficile. Un algorithme de programmation dynamique a été développé, démontrant du même coup que le problème est NP-Difficile au sens faible. Nous avons ensuite construit un modèle de programmation linéaire en nombres entiers. Enfin, nous avons développé plusieurs versions d'une méthode de Monte-Carlo sur les chaînes de Markov dans le but de disposer d'une méthode « gratuite ».

Des tests ont été réalisés à partir d'instances constituées de 400 à 800 plans de production issus de la résolution du premier problème d'optimisation. Les expérimentations ont montré que la méthode de programmation dynamique n'est pas adaptée car trop gourmande en mémoire. La programmation linéaire en nombres entiers permet d'obtenir de très bons résultats (0,2 % de l'optimum), mais nécessite de disposer d'un logiciel payant. Enfin, quelques améliorations de la méthode de Monte-Carlo sur les chaînes de Markov nous ont permis d'atteindre des résultats quasi-similaires à ceux de la programmation linéaire, avec des solutions à 0,8 % de l'optimum au bout d'une minute d'exécution. Ces méthodes ont été présentées à la conférence JOPT 2014 [Prelle *et al.*, 2014c].

Tous les travaux de cette thèse ont fait l'objet d'un dépôt de brevet [Delon *et al.*, 2014].

6.2 Perspectives

Comme nous l'avons vu dans la conclusion du chapitre 4, une piste de recherche qui nous parait intéressante concerne l'étude de l'impact du signal de prix sur les plans de production proposés par les procédés. Nous pensons en effet qu'il serait possible d'adapter le signal de prix à chaque procédé afin de mieux optimiser le plan de production de la centrale. Une proposition d'algorithme qui pourrait être utilisé pour modifier les signaux de prix est donnée dans l'annexe B.

De plus, tous ces travaux ont été réalisés sous l'hypothèse que le signal de prix envoyé par le client est un signal de prix linéaire. Un tel signal ne permet que d'augmenter la production en choisissant un coefficient directeur positif ou d'augmenter la consommation en choisissant un coefficient directeur négatif. Tous les autres objectifs comme par exemple une augmentation de la production limitée, un lissage ou un îlotage nécessitent d'avoir un signal de prix non linéaire. Un tel signal de prix aurait différents impacts sur les méthodes que nous avons développées :

- la programmation dynamique n'est pas impactée par ce changement de signal de prix mais nous n'avons préconisé cette méthode ni pour le premier problème, ni pour le second problème ;

- la méthode de Monte-Carlo sur les chaines de markov devrait être peu impactée car ce changement de signal de prix n'a pas d'impact sur la formulation de la méthode mais ce changement pourrait avoir un impact sur sa convergence ;

- la programmation linéaire en nombres entiers devrait être impactée car ce changement de signal de prix nécessite l'utilisation de contraintes de type "special order set de type 2" qui modélise le signal de prix linéaire par morceaux. Or ces contraintes augmentent significativement le nombre de contraintes et de variables binaires. Nous ne pouvons pas prédire la réaction des solveurs de programmation linéaire en nombres entiers à l'utilisation de ce type de contrainte. Et le signal de prix devra être approximé à un signal de prix linéaire par morceaux ;

- la programmation par contraintes devrait être la méthode la plus impactée par ce changement de signal de prix. Bien que le modèle de programmation par contraintes n'ait pas besoin d'être modifié

(en effet, la contrainte $costRegular$ permet de prendre en compte ce changement), il faudra trouver une nouvelle stratégie de branchement.

De plus, l'utilisation d'un tel signal de prix nécessite de développer un algorithme qui répercute le signal de prix reçu par le coordinateur sur les systèmes pilotés. En effet un signal de prix linéaire est facilement transférable du coordinateur vers les systèmes pilotés : il peut être directement envoyé aux systèmes pilotés sans modification. Par contre si un coordinateur reçoit un signal de prix qui l'incite à l'îlotage, quels signaux de prix va-t-il envoyer à ses systèmes pilotés afin que le coordinateur réalise son îlotage ?

Et enfin, les versions stochastiques de nos problèmes devront être étudié afin de garantir la robustesse des méthodes. En effet, nous supposons que la puissance interne produite/consommée est connue alors que dans la réalité nous ne connaissons que des prévisions de production et de consommation. Et divers pannes liées à la communication téléphonique ou internet ou aux systèmes pilotés devront être prises en compte.

A

Version désagrégée des contraintes des procédés

Dans cette annexe nous proposons la version désagrégée de toutes les contraintes liées uniquement à 4.16 ou à 4.17 ainsi que les contraintes combinant 4.16 et 4.17 :

- 4.16 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [t+1; min(T-1, t+T^{\alpha}_{minOn}-1)], s_{\alpha}(t)+e_{\alpha}(t') \leq 1 \quad \text{(A.1)}$$

- 4.17 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [t+1; min(T-1, t+T^{\alpha}_{minOff}-1)], e_{\alpha}(t)+s_{\alpha}(t') \leq 1 \quad \text{(A.2)}$$

- 4.30 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [t; min(T-1, t+T^{\alpha}_{minOn}-1)], s_{\alpha}(t) \leq p_{\alpha}(t') \quad \text{(A.3)}$$

- 4.31 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [max(0, t-T^{\alpha}_{minOn}), t-1], e_{\alpha}(t) \leq p_{\alpha}(t') \quad \text{(A.4)}$$

- 4.32 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [max(0, t-T^{\alpha}_{minOff}), t-1], s_{\alpha}(t)+p_{\alpha}(t') \leq 1 \quad \text{(A.5)}$$

- 4.33 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [t+1; min(T-1, t+T^{\alpha}_{minOff}-1)], e_{\alpha}(t)+p_{\alpha}(t') \leq 1 \quad \text{(A.6)}$$

- 4.36 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [max(0, t-T^{\alpha}_{minOff}+1); min(T-1, t+T^{\alpha}_{minOn}-1)], s_{\alpha}(t) \leq 1-e_{\alpha}(t') \quad \text{(A.7)}$$

- 4.37 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [max(0, t-T^{\alpha}_{minOn}+1); min(T-1, t+T^{\alpha}_{minOff}-1)], s_{\alpha}(t) \leq 1-e_{\alpha}(t') \quad \text{(A.8)}$$

- 4.38 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [t+1, min(T-1; t+T^{\alpha}_{minOn}+T^{\alpha}_{minOff}-1)] s_{\alpha}(t) \leq 1-s_{\alpha}(t') \tag{A.9}$$

- 4.39 :

$$\forall \alpha \in \{p^+; p^-; x^+; x^-\}, \forall t \in \mathcal{T}, \forall t' \in [t+1; min(T-1, t+T^{\alpha}_{minOn}+T^{\alpha}_{minOff}-1)] e_{\alpha}(t) \leq 1-e_{\alpha}(t') \tag{A.10}$$

B

Proposition d'un algorithme de modification d'un signal de prix linéaire

Afin de modifier un signal de prix linéaire, une voie serait d'utiliser l'algorithme suivant :

Phase 1 : Le coordinateur envoie aux procédés pilotés le signal de prix qu'il a reçu (en gardant une marge).

Phase 2 : Le coordinateur reçois un plan de production par systèmes pilotés. Il dispose pour chaque système piloté sp d'un ensemble $\mathcal{S}_{sp}$ de plan de production. Le coordinateur agrège les plans de production reçus. Si le plan est satisfaisant, le coordinateur envoie son plan à son client sinon il passe à la phase suivante.

Phase 3 : Afin d'améliorer son plan de production, le coordinateur envoie à tous les systèmes pilotés qui n'ont pas été choisis dans la Phase 2, un nouveau signal de prix. Pour tous t, plus la marge (rapport entre la puissance du coordinateur et la borne) sera petite plus la norme du signal de prix sera réduite :

$$\forall t \in \mathcal{T}, SP_{mod}(t) = \begin{cases} SP(t)(1 - \frac{\delta_{coor}(t)}{\max(P_{max}(t),1)}) \text{ si } SP(t) \text{ est positif} \\ SP(t)(1 - \frac{\delta_{coor}(t)}{\min(P_{min}(t),-1)}) \text{ si } SP(t) \text{ est négatif} \\ 0 \text{ sinon} \end{cases}$$

avec $SP(t)$ le signal de prix envoyé à l'itération précédente pour le temps t, $\delta_{coor}(t)$ la puissance du coordinateur calculée à l'itération précédente au temps t, et $P_{min}(t)$ ($P_{max}(t)$), la variation de puissance minimale (maximale) du coordinateur au temps t.

Phase 4 : Le coordinateur reçois un plan de production par systèmes pilotés qui n'avais pas été choisis à l'agrégation précédente. Il dispose donc pour chaque système piloté sp d'un ensemble $\mathcal{S}_{sp}$ de plan de production. Le coordinateur agrège tous les plans de production en ne choisissant au plus un plan de production par $\mathcal{S}_{sp}$.

Phase 5 : Si le plan est satisfaisant (par exemple si le plan de production du coordinateur n'a pas changé pendant cette itération, ou si tous les procédés sont dans la solution, ou si toutes les contraintes sont saturées) alors le coordinateur envoie son plan à son client. Sinon il retourne à la phase 3.

L'avantage de cet algorithme est qu'il ne peut dégrader la solution. Cependant le nombre de procédés agrégeables dans le pire des cas double à chaque itération.

Bibliographie

[ADDRESS, 2008] ADDRESS. Le projet ADDRESS. `http://www.addressfp7.org`, 2008. 49, 50

[Agnetis *et al.*, 2013] A. Agnetis, G. de Pascale, P. Detti, and A. Vicino. Load scheduling for household energy consumption optimization. *Smart Grid, IEEE Transactions on*, 4(4) :2364–2373, 2013. 49, 62

[Ahn *et al.*, 2011] C. Ahn, C. Li, and H. Peng. Decentralized charging algorithm for electrified vehicles connected to smart grid. In *American Control Conference (ACC), 2011*, pages 3924 –3929, 29 2011-july 1 2011. 61

[Al-Awami *et al.*, 2009] A.T. Al-Awami, E. Sortomme, and M.A. El-Sharkawi. Optimizing economic/environmental dispatch with wind and thermal units. In *Power Energy Society General Meeting, 2009. PES '09. IEEE*, pages 1 –6, july 2009. 62

[Alonso *et al.*, 2012] Monica Alonso, Hortensia Amaris, and Carlos Alvarez-Ortega. Integration of renewable energy sources in smart grids by means of evolutionary optimization algorithms. *Expert Systems with Applications*, 39(5) :5513 – 5522, 2012. 63

[Aoki *et al.*, 1987] K. Aoki, T. Satoh, M. Itoh, T. Ichimori, and K. Masegi. Unit commitment in a large-scale power system including fuel constrained thermal and pumped-storage hydro. *Power Engineering Review, IEEE*, PER-7(11) :51 –52, nov. 1987. 49, 62

[Asato *et al.*, 2010] B. Asato, T. Goya, K. Uchida, A. Yona, T. Senjyu, T. Funabashi, and Chul-Hwan Kim. Optimal operation of smart grid in isolated island. In *IPEC, 2010 Conference Proceedings*, pages 1100 –1105, oct. 2010. 62

[Barbato *et al.*, 2011] A. Barbato, A. Capone, G. Carello, M. Delfanti, M. Merlo, and A. Zaminga. House energy demand optimization in single and multi-user scenarios. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 345 –350, oct. 2011. 49, 62

[Becker and Yeh, 1974] L. Becker and W. W-G Yeh. Optimization of real time operation of a multiple reservoir system. *Water Resources Research*, 10 :1107–1112, June 1974. 61

[Bellman, 1954] Richard Bellman. The theory of dynamic programming. Technical report, DTIC Document, 1954. 125

[Bellman, 1958] Richard Bellman. *On a routing problem*, volume 16. Quarterly of Applied Mathematics, 1958. 125

[Bozchalui *et al.*, 2012] M.C. Bozchalui, S.A. Hashmi, H. Hassen, C.A. Canizares, and K. Bhattacharya. Optimal operation of residential energy hubs in smart grids. *Smart Grid, IEEE Transactions on*, 3(4) :1755–1766, 2012. 49, 62

[Briley and McDonald, 1977] WR Briley and H McDonald. Solution of the multidimensional compressible navier-stokes equations by a generalized implicit method. *Journal of Computational Physics*, 24(4) :372–397, 1977. 124

[Buck *et al.*, 2002] Reiner Buck, Markus Pfänder, Peter Schwarzbözl, Felix Tellez, Thomas Bräuning, and Thorsten Denk. Solar-hybrid gas turbine-based power tower systems (refos). *Journal of Solar Energy Engineering*, 124(1) :2–9, 2002. 49

[CEA, 2012] CEA. Energie et environnement. `http://www.cea.fr/jeunes/themes/l-energie/la-production-d-energie/energie-et-environnement/`, 2012. 23

[Chen *et al.*, 2011] Yen-Haw Chen, Yen-Hong Chen, and Ming-Che Hu. Optimal energy management of microgrid systems in taiwan. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1 –9, jan. 2011. 62

[Chen *et al.*, 2013] Z. Chen, L. Wu, and Z. Li. Electric demand response management for distributed large-scale internet data centers, 2013. 49, 62

[Cohen and Wan, 1985] A.I. Cohen and S.H. Wan. An algorithm for scheduling a large pumped storage plant. *Power Apparatus and Systems, IEEE Transactions on*, PAS-104(8) :2099 –2104, aug. 1985. 49, 62

[Delon and Normand, 2011] Pierre Etienne Delon and Olivier Normand. Ensemble de pilotage d'un système de production/consommation electrique, wo. patent 2 012 168 323, 10 06 2011. 63, 86

[Delon *et al.*, 2014] Pierre-Etienne Delon, Kengy Barty, Marc Le Du, and Thomas Prelle. Système de gestion d'un réseau de distribution électrique, fr n° de dépôt : 1451087, 12 02 2014. 139

[Demassey *et al.*, 2006] Sophie Demassey, Gilles Pesant, and Louis-Martin Rousseau. A cost-regular based hybrid column generation approach. *Constraints*, 11(4) :315–333, 2006. 95

[El Guedri *et al.*, 2009] Mabrouka El Guedri, Julien Bect, Christian Lajaunie, Gilles Fleury, Rédouane Seraoui, Thomas Garcia, Alexandre Girard, et al. Rjmcmc point process sampler for single sensor source separation : an application to electric load monitoring. In *Proceedings of the 17th European Signal Processing Conference*, 2009. 130

[EWG, 2007] EWG. COAL : RESOURCES AND FUTURE PRODUCTION. `http://energywatchgroup.org/wp-content/uploads/2014/02/EWG_Report_Coal_10-07-2007ms.pdf/`, 2007. 23

[Faria *et al.*, 2011] P. Faria, Z.A. Vale, J. Soares, and J. Ferreira. Particle swarm optimization applied to integrated demand response resources scheduling. In *Computational Intelligence Applications In Smart Grid (CIASG), 2011 IEEE Symposium on*, pages 1 –8, april 2011. 62

[Faria *et al.*, 2013] P. Faria, J. Soares, Z. Vale, H. Morais, and T. Sousa. Modified particle swarm optimization applied to integrated demand response and dg resources scheduling. *Smart Grid, IEEE Transactions on*, 4(1) :606–616, 2013. 62

[Fernandes *et al.*, 2011] F. Fernandes, T. Sousa, M. Silva, H. Morais, Z. Vale, and P. Faria. Genetic algorithm methodology applied to intelligent house control. In *Computational Intelligence Applications In Smart Grid (CIASG), 2011 IEEE Symposium on*, pages 1 –8, april 2011. 49, 62

[Fréville, 2004] Arnaud Fréville. The multidimensional 0–1 knapsack problem : An overview. *European Journal of Operational Research*, 155(1) :1–21, 2004. 130

[Garey and Johnson, 1979] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. Freeman New York, 1979. 83, 85, 124

[Georgievski *et al.*, 2012] I. Georgievski, V. Degeler, G.A. Pagani, Tuan Anh Nguyen, A. Lazovik, and M. Aiello. Optimizing energy costs for offices connected to the smart grid. *Smart Grid, IEEE Transactions on*, 3(4) :2273–2285, 2012. 49, 62

[Goya *et al.*, 2011a] T. Goya, T. Senjyu, A. Yona, K. Uchida, T. Funabashi, and Chul-Hwan Kim. Optimal operation of thermal generating units and smart houses. In *Power Electronics and Drive Systems (PEDS), 2011 IEEE Ninth International Conference on*, pages 418–423, 2011. 49, 62

[Goya *et al.*, 2011b] T. Goya, T. Senjyu, A. Yona, N. Urasaki, T. Funabashi, and Chul-Hwan Kim. Thermal units commitment considering voltage constraint based on controllable loads reactive control in smart grid. In *Power Electronics and Drive Systems (PEDS), 2011 IEEE Ninth International Conference on*, pages 793–798, 2011. 63

[Green and Sanduk, 2013] H. Green and M. Sanduk. Development of a wind turbine system with energy storage in the form of hydropower. In *Applications of Information Technology to Renewable Energy Processes and Systems (IT-DREPS), 2013 1st International Conference Exhibition on the*, pages 3–8, 2013. 49

[Gu *et al.*, 2010] W. Gu, Z. Wu, and X. Yuan. Microgrid economic optimal operation of the combined heat and power system with renewable energy. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1 –6, july 2010. 49, 62

[Guan *et al.*, 1994] X. Guan, P.B. Luh, H. Yen, and P. Rogan. Optimization-based scheduling of hydro-thermal power systems with pumped-storage units. *Power Systems, IEEE Transactions on*, 9(2) :1023 –1031, may 1994. 61

[Guangwen and Li, 1997] Ma Guangwen and Wang Li. Application of a genetic algorithm to optimal operation of hydropower station [j]. *Advances in water science*, 3, 1997. 61

[Gudi *et al.*, 2010] N. Gudi, W. Lingfeng, V. Devabhaktuni, and S.S.S.R. Depuru. Demand response simulation implementing heuristic optimization for home energy management. In *North American Power Symposium (NAPS), 2010*, pages 1 –6, sept. 2010. 49, 62

[Han *et al.*, 2010] Sekyung Han, Soohee Han, and K. Sezaki. Development of an optimal vehicle-to-grid aggregator for frequency regulation. *Smart Grid, IEEE Transactions on*, 1(1) :65 –72, june 2010. 63

[Hastings, 1970] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1) :97–109, 1970. 131

[Hermans *et al.*, 2011] Y. Hermans, B. Le Cun, and A. Bui. Modèle d'optimisation basé sur le vehicle-to-grid pour limiter l'impact des pics de consommation électrique sur la production. In *Roadef 2011*, 2011. 62

[Heussen *et al.*, 2010] K. Heussen, S. Koch, A. Ulbig, and G. Andersson. Energy storage in power system operation : The power nodes modeling framework. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pages 1 –8, oct. 2010. 63, 64

[Hopcroft *et al.*, 2001] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation, 2nd, 2001. 95

[Horowitz and Sahni, 1974] Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *J. ACM*, 21(2) :277–292, April 1974. 127

[Inoue *et al.*, 2014] T. Inoue, K. Takano, T. Watanabe, J. Kawahara, R. Yoshinaka, A. Kishimoto, K. Tsuda, S.-I. Minato, and Y. Hayashi. Distribution loss minimization with guaranteed error bound. *Smart Grid, IEEE Transactions on*, 5(1) :102–111, 2014. 63

[Jin *et al.*, 2013] Chenrui Jin, Jian Tang, and P. Ghosh. Optimizing electric vehicle charging with energy storage in the electricity market. *Smart Grid, IEEE Transactions on*, 4(1) :311–320, 2013. 61

[JORF, 2013] JORF. Journal officiel lois et décrets n 0089, 16 avril 2013. 31

[Kahrobaee *et al.*, 2013] S. Kahrobaee, S. Asgarpoor, and Wei Qiao. Optimum sizing of distributed generation and storage capacity in smart households. *Smart Grid, IEEE Transactions on*, 4(4) :1791–1801, 2013. 49, 62

[Kane *et al.*, 2003] M Kane, D Larrain, Daniel Favrat, and Yassine Allani. Small hybrid solar power system. *Energy*, 28(14) :1427–1443, 2003. 49

[Kim and Lavrova, 2013] Byungchul Kim and O. Lavrova. Optimal power flow and energy-sharing among multi-agent smart buildings in the smart grid. In *Energytech, 2013 IEEE*, pages 1–5, 2013. 49, 62

[Koutsopoulos *et al.*, 2011] I. Koutsopoulos, V. Hatzi, and L. Tassiulas. Optimal energy storage control policies for the smart power grid. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 475 –480, oct. 2011. 62

[Laaksonen *et al.*, 2011] H. Laaksonen, K. Kauhaniemi, and S. Voima. Microgrid voltage level management and role as part of smart grid voltage control. In *PowerTech, 2011 IEEE Trondheim*, pages 1 –8, june 2011. 63

[Li *et al.*, 2011] W. Li, J. Huang, G. Li, and Z. Wang. Research on optimizing operation of the single reservoir of hybrid pumped storage power station. In *Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2011 4th International Conference on*, pages 1389 –1394, july 2011. 61

[Lu *et al.*, 2011] Shuai Lu, N. Samaan, Ruisheng Diao, M. Elizondo, Chunlian Jin, E. Mayhorn, Yu Zhang, and H. Kirkham. Centralized and decentralized control for demand response. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1 –8, jan. 2011. 63

[Martello and Toth, 1990] Silvano Martello and Paolo Toth. *Knapsack Problems : Algorithms and Computer Implementations*. John Wiley & Sons, 1990. 124, 126

[Metropolis *et al.*, 1953] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21 :1087, 1953. 130

[Nguyen and Le, 2014] D.T. Nguyen and L.B. Le. Joint optimization of electric vehicle and home energy scheduling considering user comfort preference. *Smart Grid, IEEE Transactions on*, 5(1) :188–199, 2014. 49

[NiceGrid, 2011] NiceGrid. Le projet NiceGrid. `http://www.nicegrid.fr/`, 2011. 49

[Ochoa and Harrison, 2011] L. Ochoa and G. Harrison. Minimizing energy losses : Optimal accommodation and smart operation of renewable distributed generation. In *Power and Energy Society General Meeting, 2011 IEEE*, page 1, july 2011. 63

[Ölçer *et al.*, 1985] S Ölçer, C Harsa, and A Roch. Application of linear and dynamic programming to the optimization of the production of hydroelectric power. *Optimal Control Applications and Methods*, 6(1) :43–56, 1985. 61

[Oliveira *et al.*, 2011] Grégory De Oliveira, Mireille Jacomino, and Stéphane Ploix. Optimisation de l'énergie dans l'habitat : du smart home au smart grid. In *Roadef 2011*, 2011. 49, 62

[Oshiro *et al.*, 2012] M. Oshiro, Z. Ziadi, T. Senjyu, A. Yona, T. Funabashi, and Chul-Hwan Kim. Optimal scheduling method in distribution system considering controllable loads. In *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, pages 96–101, 2012. 49, 62

[Pang *et al.*, 2012] C. Pang, P. Dutta, and M. Kezunovic. Bevs/phevs as dispersed energy storage for v2b uses in the smart grid. *Smart Grid, IEEE Transactions on*, 3(1) :473 –482, march 2012. 61

[Pedrasa *et al.*, 2010] M.A.A. Pedrasa, T.D. Spooner, and I.F. MacGill. Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. *Smart Grid, IEEE Transactions on*, 1(2) :134–143, 2010. 49, 62

[PEGASE, 2013] PEGASE. Le projet PEGASE. `http://www.promes.cnrs.fr/pegase/`, 2013. 23, 49, 62

[Pesant, 2004] Gilles Pesant. A regular language membership constraint for finite sequences of variables. In *Principles and Practice of Constraint Programming–CP 2004*, pages 482–495. Springer, 2004. 95

[Pisinger, 1995] David Pisinger. *An O (nr) Algorithm for the Subset-sum Problem*. Datalogisk Institut, Københavns Universitet, 1995. 80

[Possemato *et al.*, 2013] F. Possemato, G.L. Storti, M. Paschero, A. Rizzi, and F.M.F. Mascioli. Two evolutionary computational approaches for active power losses minimization in smart grids. In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pages 401–406, 2013. 63

[Prelle *et al.*, 2014a] Thomas Prelle, Christelle Gueret, Thomas Prelle, Pierre-Etienne Delon, and Kengy Barty. Approches génériques pour calculer le plan de production des procédés d'un smart grid. In *ROADEF 2014*, 2014. 138

[Prelle *et al.*, 2014b] Thomas Prelle, Christelle Gueret, Thomas Prelle, Pierre-Etienne Delon, and Kengy Barty. Towards an approach to manage smart grids like any other power plant. In *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, pages 1–5, Feb 2014. 138

[Prelle *et al.*, 2014c] Thomas Prelle, Christelle Gueret, Thomas Prelle, Pierre-Etienne Delon, and Kengy Barty. Two approaches to aggregate smart grid's energy systems' production plan. In *Journée de l'Optimisation 2014*, 2014. 139

[PREMIO, 2007] PREMIO. Le projet PREMIO. `http://www.projetpremio.fr`, 2007. 49, 50

[Qudaih and Mitani, 2011] Yaser Soliman Qudaih and Yasunori Mitani. Power distribution system planning for smart grid applications using ann. *Energy Procedia*, 12(0) :3 – 9, 2011. <ce :title>The Proceedings of International Conference on Smart Grid and Clean Energy Technologies (ICSGCE 2011</ce :title>. 63

[RTE, 2012] RTE. éco2mix. `http://www.rte-france.com/fr/developpement-durable/eco2mix/`, 2012. 27, 31

[RTE, 2014a] RTE. Bilan électrique 2013. `http://www.rte-france.com/uploads/Mediatheque_docs/vie_systeme/annuelles/Bilan_electrique/bilan_electrique_2013.PDF`, 2014. 31

[RTE, 2014b] RTE. Les chemins de l'électricité. `http://www.rte-france.com/fr/actualites-dossiers/comprendre/les-chemins-de-l-electricite-1/les-chemins-de-l-electricite`, 2014. 27

[Ruomei *et al.*, 1995] L. Ruomei, C. Yunping, and G. Jianbo. An application of ann in scheduling pumped-storage. In *Energy Management and Power Delivery, 1995. Proceedings of EMPD '95., 1995 International Conference on*, volume 1, pages 85 –90 vol.1, nov 1995. 61

[Saber and Venayagamoorthy, 2012] A.Y. Saber and G.K. Venayagamoorthy. Resource scheduling under uncertainty in a smart grid with renewables and plug-in vehicles. *Systems Journal, IEEE*, 6(1) :103 –109, march 2012. 62

[Sanchez de la Nieta *et al.*, 2013] A.A. Sanchez de la Nieta, J. Contreras, and J.I. Munoz. Optimal coordinated wind-hydro bidding strategies in day-ahead markets. *Power Systems, IEEE Transactions on*, 28(2) :798–809, 2013. 49, 62

[Schwarzbözl *et al.*, 2006] Peter Schwarzbözl, Reiner Buck, Chemi Sugarmen, Arik Ring, Ma Jesús Marcos Crespo, Peter Altwegg, and Juan Enrile. Solar gas turbine systems : Design, cost and perspectives. *Solar Energy*, 80(10) :1231–1240, 2006. 49

[Sedgewick and Wayne, 2011] Robert Sedgewick and Kevin Wayne. *lgorithms, 4th Edition*. Addison-Wesley Professional, 2011. 77

[Soares *et al.*, 2011] J. Soares, T. Sousa, H. Morais, Z. Vale, and P. Faria. An optimal scheduling problem in distribution networks considering v2g. In *Computational Intelligence Applications In Smart Grid (CIASG), 2011 IEEE Symposium on*, pages 1 –8, april 2011. 61

[Sobu and Wu, 2012a] A. Sobu and Guohong Wu. Dynamic optimal schedule management method for microgrid system considering forecast errors of renewable power generations. In *Power System Technology (POWERCON), 2012 IEEE International Conference on*, pages 1–6, 2012. 49, 62

[Sobu and Wu, 2012b] A. Sobu and Guohong Wu. Optimal operation planning method for isolated micro grid considering uncertainties of renewable power generations and load demand. In *Innovative Smart Grid Technologies - Asia (ISGT Asia), 2012 IEEE*, pages 1–6, 2012. 49, 62

[Tanaka *et al.*, 2012] K. Tanaka, A. Yoza, K. Ogimi, A. Yona, T. Senjyu, T. Funabashi, and C. Kim. Optimal operation of dc smart house system by controllable loads based on smart grid topology. *Renewable Energy*, 39(1) :132 – 139, 2012. 49, 62

[Tian and Zhao, 2013] Y. Tian and C.Y. Zhao. A review of solar collectors and thermal energy storage in solar thermal applications. *Applied Energy*, 104(0) :538 – 553, 2013. 49

[Tushar *et al.*, 2014] M.H.K. Tushar, C. Assi, M. Maier, and M.F. Uddin. Smart microgrids : Optimal joint scheduling for electric vehicles and home appliances. *Smart Grid, IEEE Transactions on*, 5(1) :239–250, 2014. 49, 62

[Vachirasricirikul and Ngamroo, 2014] S. Vachirasricirikul and I. Ngamroo. Robust lfc in a smart grid with wind power penetration by coordinated v2g control and frequency controller. *Smart Grid, IEEE Transactions on*, 5(1) :371–380, 2014. 49, 63

[Viehweider *et al.*, 2010] A. Viehweider, H. Schichl, D.B. de Castro, S. Henein, and D. Schwabeneder. Smart robust voltage control for distribution networks using interval arithmetic and state machine concepts. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pages 1 –8, oct. 2010. 63

[Voice *et al.*, 2011] T. Voice, P. Vytelingum, S. Ramchurn, A. Rogers, and N. Jennings. Decentralised control of micro-storage in the smart grid. In *AAAI-11 : Twenty-Fifth Conference on Artificial Intelligence*, pages 1421–1426, 2011. 62

[Wang and Yang, 2013] Jidong Wang and Fan Yang. Optimal capacity allocation of standalone wind/solar/battery hybrid power system based on improved particle swarm optimisation algorithm. *Renewable Power Generation, IET*, 7(5) :443–448, 2013. 49, 62

[Wang *et al.*, 2006] W. C. Wang, C. T. Cheng, and D. M. Xu. The optimal operation model based on chaos genetic algorithm for hydropower station and its application. In *Journal of Hydroelectric Engineering*, volume 26, pages 7–11, 2006. 61

[Wang *et al.*, 2011] Zhu Wang, Rui Yang, and Lingfeng Wang. Intelligent multi-agent control for integrated building and micro-grid systems. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1 –7, jan. 2011. 63

[Wei *et al.*, 2010] C. Wei, H. Hu, Q. Chen, and G. Yang. Learning agents for storage devices management in the smart grid. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1 –4, dec. 2010. 62

[Wenqian, 2005] Q. Wenqian. Dynamic programming models for operation optimization of pumped storage power stations [j]. In *Hydropower automation and dam monitoring*, June 2005. 61

[Wille-Haussmann *et al.*, 2010] B. Wille-Haussmann, T. Erge, and C. Wittwer. Decentralised optimisation of cogeneration in virtual power plants. *Solar Energy*, 84(4) :604 – 611, 2010. International Conference CISBAT 2007. 49, 62

[Xiong *et al.*, 2011] G. Xiong, C. Chen, S. Kishore, and A. Yener. Smart (in-home) power scheduling for demand response on the smart grid. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1 –7, jan. 2011. 49, 62

[Xiwen *et al.*, 2010] Wei Xiwen, Qiu Xiaoyan, Xu Jian, and Li Xingyuan. Reactive power optimization in smart grid with wind power generator. In *Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific*, pages 1 –4, march 2010. 63

[Yoon *et al.*, 2014] J.H. Yoon, R. Baldick, and A. Novoselac. Dynamic demand response controller based on real-time retail price for residential buildings. *Smart Grid, IEEE Transactions on*, 5(1) :121–129, 2014. 49

[Zhang and Chow, 2011] Ziang Zhang and Mo-Yuen Chow. The leader election criterion for decentralized economic dispatch using incremental cost consensus algorithm. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pages 2730 –2735, nov. 2011. 62

[Zhao and Wu, 2014] Z. Zhao and L. Wu. Impacts of high penetration wind generation and demand response on lmps in day-ahead market. *Smart Grid, IEEE Transactions on*, 5(1) :220–229, 2014. 49, 62

[Zhao *et al.*, 2013] Zhuang Zhao, Won Cheol Lee, Yoan Shin, and Kyung-Bin Song. An optimal power scheduling method for demand response in home energy management system. *Smart Grid, IEEE Transactions on*, 4(3) :1391–1400, 2013. 49, 62

l'université nantes angers le mans
COMMUNAUTÉ D'UNIVERSITÉS ET ÉTABLISSEMENTS

MINES Nantes
Institut Mines-Télécom

Thèse de Doctorat

Thomas PRELLE

Gestion optimisée d'un modèle d'agrégation de flexibilités diffuses

Optimized management of a distributed demand response aggregation model

Résumé
Le souhait d'augmenter la part des énergies renouvelables dans le mix énergétique entraine une augmentation des parts des énergies volatiles et non pilotables, et rend donc l'équilibre offre-demande difficile à satisfaire. Une façon d'intégrer ces énergies dans le réseau électrique actuel est d'utiliser de petits moyens de production, de consommation et de stockage répartis sur tout le territoire pour compenser les sous ou sur productions. Afin que ces procédés puissent être intégrés dans le processus d'équilibre offre-demande, ils sont regroupés au sein d'une centrale virtuelle d'agrégation de flexibilité, qui est vue alors comme une centrale virtuelle. Comme pour tout autre moyen de production du réseau, il est nécessaire de déterminer son plan de production. Nous proposons dans un premier temps dans cette thèse une architecture et un mode de gestion pour une centrale d'agrégation composée de n'importe quel type de procédés. Dans un second temps, nous présentons des algorithmes permettant de calculer le plan de production des différents types de procédés respectant toutes leurs contraintes de fonctionnement. Et enfin, nous proposons des approches pour calculer le plan de production de la centrale d'agrégation dans le but de maximiser son gain financier en respectant les contraintes réseau.

Mots clés
Smart Grid, Pilotage de charge, Centrale virtuelle, Equilibre offre-demande, Effacement de la consommation, Intégration des énergies renouvelables, Optimisation, Recherche opérationnelle, Programmation linéaire en nombres entiers, Programmation dynamique, Programmation par contraintes

Abstract
The desire to increase the share of renewable energies in the energy mix leads to an increase in share of volatile and non-controllable energy and makes it difficult to meet the supply-demand balance. A solution to manage anyway theses energies in the current electrical grid is to deploy new energy storage and demand response systems across the country to counterbalance under or over production. In order to integrate all these energies systems to the supply and demand balance process, there are gathered together within a virtual flexibility aggregation power plant which is then seen as a virtual power plant. As for any other power plant, it is necessary to compute its production plan. Firstly, we propose in this PhD thesis an architecture and management method for an aggregation power plant composed of any type of energies systems. Then, we propose algorithms to compute the production plan of any types of energy systems satisfying all theirs constraints. Finally, we propose an approach to compute the production plan of the aggregation power plant in order to maximize its financial profit while complying with all the constraints of the grid.

Key Words
Smart Grids, Demand response, Supply-demand balance, Load Management, Renewable energies integration, Optimization, Operational research, Mixed integer linear programming, dynamic programming, Constraint programming

L'UNIVERSITÉ NANTES ANGERS LE MANS