HAL
open science

Résolution exacte du problème de l'optimisation des flux de puissance

Hadrien Godard

▶ **To cite this version:**

Hadrien Godard. Résolution exacte du problème de l'optimisation des flux de puissance. Optimisation et contrôle [math.OC]. Conservatoire national des arts et metiers - CNAM, 2019. Français. NNT : 2019CNAM1258 . tel-02463203v2

HAL Id: tel-02463203

https://theses.hal.science/tel-02463203v2

Submitted on 23 Jan 2020

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

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

le cnam

ÉCOLE DOCTORALE EDITE
CEDRIC

THÈSE DE DOCTORAT

présentée par : **Hadrien Godard**
soutenue le : **17 décembre 2019**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Discipline : **Informatique**

Résolution exacte du problème de l'Optimisation des Flux de Puissance

THÈSE dirigée par

Madame Elloumi Sourour	*Professeure, ENSTA*

et co-encadrée par

Madame Lambert Amélie	*Maître de conférence, CNAM*
Monsieur Maeght Jean	*Docteur, RTE*

RAPPORTEURS

Monsieur Anjos Miguel	*Professeur, Université d'Édimbourg*
Monsieur Van Hentenryck Pascal	*Professeur, Georgia Tech*
Monsieur Wehenkel Louis	*Professeur, Université de Liège*

PRÉSIDENT DU JURY

Madame Zidani Hasnaa	*Professeure, ENSTA*

EXAMINATEUR

Monsieur Le Ruyet Didier	*Professeur, CNAM*

Remerciements

Je tiens tout d'abord à remercier très fortement ma directrice de thèse Sourour Elloumi pour son soutien et sa disponibilité dans la réalisation de ce travail. Je la remercie également d'avoir accepté de m'encadrer pour cette thèse. Je remercie également très fortement Amélie Lambert, co-encadrante de cette thèse, pour toute l'aide qu'elle m'a apportée, notamment à travers ses nombreuses relectures qui ont permis d'améliorer grandement la qualité du présent manuscrit. Je remercie Jean Maeght, mon encadrant chez RTE, pour la confiance dont il a fait preuve en me prenant en tant que stagiaire puis doctorant. Je le remercie également pour son aide et pour le plaisir pris à travailler au quotidien à ses côtés. Je remercie Manuel Ruiz qui m'a également encadré chez RTE pour tous ses conseils et son aide dans l'élaboration des outils algorithmiques de cette thèse ainsi que pour la préparation des présentations que j'ai pu faire.

Je remercie Miguel Anjos, Pascal Van Hentenryck et Louis Wehenkel d'avoir été rapporteurs de cette thèse. Je les remercie également pour l'intérêt qu'ils ont porté à mon travail dans les discussions que j'ai pu avoir avec eux au sein de RTE ou en conférence. Je remercie Hasnaa Zidani d'avoir accepté d'être présidente de mon jury. Je remercie également Didier Le Ruyet d'avoir participé à mon jury de thèse.

Je remercie les chefs successifs du pôle au sein duquel je travaillais à RTE, à savoir Lucian Balea puis Gwilherm Poullennec. Je remercie Patrick Panciatici, conseiller scientifique de la Direction R&D de RTE, pour ses nombreux propos passionnants.

Je remercie Marie-Christine Costa et Christophe Picouleau pour leur rôle de chef de l'équipe OC du laboratoire CEDRIC. Je les remercie également pour leurs cours auxquels j'ai assistés en tant qu'étudiant avant de démarrer cette thèse.

Je remercie Cédric Josz, qui m'a précédé en tant que doctorant au sein de RTE sur un sujet très

proche du mien, pour ses conseils. Je le remercie également de m'avoir invité à intervenir à la conférence INFORMS qui eut lieu à Phoenix en novembre 2019.

Je remercie Pierre Carpentier, professeur à l'ENSTA Paris de m'avoir encouragé lors de cette thèse et de m'avoir fait confiance pour présenter mes travaux dans le cadre de la promotion de la recherche scientifique au sein de l'ENSTA Paris.

Je remercie Dominique Orban de m'avoir donné l'opportunité de travailler avec lui à Toulouse au cours de cette thèse.

Je tiens à remercier Édouard Dumon aux côtés duquel ce fut un plaisir de travailler lors du stage qui a précédé à cette thèse. Je remercie également Julie Sliwak et Clémentine Straub, collègues doctorantes au sein de RTE pour leur bonne humeur quotidienne. Je remercie spécialement Xavier Blanchot pour l'aide qu'il a apporté à cette thèse, et notamment dans le développement de l'algorithme de *branch-and-bound* spatial. Je lui souhaite bonne chance pour la poursuite de sa thèse.

Je remercie mes collègues de RTE Fabiola Aravena et Sébastien Finet avec qui se fut un plaisir de se retrouver les midis.

Je remercie mes collègues doctorants du CEDRIC, Marie de Faverges, Arnaud Lazare et Antoine Houdayer avec qui se fut toujours un plaisir de discuter.

Je remercie mes amis compagnons de route de doctorat pendant ces trois années. Tout d'abord Benoit Bonnet que je félicite pour son mariage, Émilien Flayac que je remercie pour l'éloge qu'il m'a fait dans ses propres remerciements, Rémi Feuillet pour son inflexible bonne humeur et enfin Rémi Lucas à qui je souhaite bonne chance pour sa dernière année de thèse. Je remercie également mes compagnons de hobby Thomas Bourguenolle et Antoine Berthezene avec qui j'ai passé d'excellents moments lors de cette dernière année.

Je remercie également ma cousine Fanny Lacoste d'être venue à ma soutenance, ma grand-mère Madeleine Lacoste à qui je dois beaucoup, Patrice Perey pour sa présence et son aide permanente ainsi que ma mère Nadine Godard à qui je dois tout.

Mes derniers mots vont à Marion Pichoud que je remercie pour cette merveilleuse année 2019.

Résumé

Cette thèse a pour objet la résolution exacte d'un problème d'optimisation des flux de puissance (OPF) dans un réseau électrique. Dans l'OPF, on doit planifier la production et la répartition des flux de puissances électriques permettant de couvrir, à un coût minimal, la consommation en différents points du réseau. Trois variantes du problème de l'OPF sont étudiées dans ce manuscrit. La première d'entres elles, $(OPF - L)$, correspond au cas où les coûts de production de puissance dans le réseau sont modélisés avec un polynôme de degré 1, on parle de coûts linéaires par abus de langage. Dans la seconde variante, $(OPF - Q)$ ces mêmes coûts sont modélisés avec un polynôme de degré 2. On dira que ces coûts sont quadratiques. Enfin, la dernière variante, $(OPF - UC)$, correspond au cas où les unités de production peuvent être allumées ou éteintes. Cette variante introduit des variables binaires de décision.

Nous nous concentrerons principalement sur la résolution exacte des deux problèmes $(OPF - L)$ et $(OPF - Q)$, puis nous montrerons comment notre approche peut naturellement s'étendre à la troisième variante $(OPF - UC)$. Les problèmes $(OPF - L)$ et $(OPF - Q)$ sont des problèmes d'optimisation non-convexe à variables complexes. Cette thèse propose de résoudre ces derniers à l'aide d'une méthode de reformulation que l'on appelle `RC-OPF`. Pour ce faire, nous proposons dans un premier temps de reformuler le problème initial en un nouveau problème quadratique possédant certaines propriétés de convexité. Notre reformulation est optimale dans le sens où elle fournit la meilleure borne inférieure parmi une famille de reformulations. Dans un second temps, la reformulation construite est résolue à l'aide d'un algorithme de *branch-and-bound*.

La contribution principale de cette thèse réside dans l'étude, le développement et l'utilisation de notre méthode de résolution exacte `RC-OPF` sur les trois variantes d'OPF. Nous montrons notamment

que l'étape de reformulation optimale de `RC-OPF` est plus efficace que celle d'autres méthodes de résolution par reformulation comme MIQCR, tout en gardant les mêmes propriétés. `RC-OPF` utilise également des techniques de contractions de bornes, et nous montrons comment ces techniques classiques peuvent être renforcées en utilisant des résultats issus de notre reformulation optimale. Des expérimentations numériques permettent de prouver l'efficacité de cette méthode dans le cadre d'instances d'OPF dont la résolution exacte est « difficile » avec les outils à disposition dans l'état de l'art.

Mots-clés : Réseaux-Électriques Énergie, Optimisation des Flux de Puissance, Optimisation non-Linéaire, Reformulation Quadratique Convexe, Programmation Quadratique, Programmation Semi-Définie, Réduction de bornes.

Abstract

Alternative Current Optimal Power Flow (ACOPF) is naturally formulated as a non-convex problem. In that context, solving (ACOPF) to global optimality remains a challenge when classic convex relaxations are not exact. We use semidefinite programming to build a quadratic convex relaxation of (ACOPF). We show that this quadratic convex relaxation has the same optimal value as the classical semidefinite relaxation of (ACOPF) which is known to be tight. In that context, we build a spatial branch-and-bound algorithm to solve (ACOPF) to global optimality that is based on a quadratic convex programming bound. An advantage of our approach is that we capture the strength of semidefinite programming bounds into a quadratic convex program that is known to be faster to solve.

Keywords : Electrical-Network Energy, Optimal Power Flow, Non-Linear Optimization, Quadratic Convex Reformulation, Quadratic Programming, Semi Definite Programming, Bound Tightening.

Table des matières

Remerciements **5**

Résumé **9**

Abstract **13**

Liste des tableaux **21**

Liste des figures **25**

Notations **27**

Introduction **31**

0.1 Présentation du problème de l'optimisation des flux de puissance (OPF) 32

0.1.1 Généralités sur la gestion des réseaux électriques 32

0.1.2 Le problème OPF 32

0.2 Modélisation mathématique de l'OPF 33

0.2.1 Représentation du réseau électrique par un graphe 34

0.2.2 Variables du problème 35

0.2.3 Données du problème 37

0.2.4 Fonction objectif et contraintes 40

0.2.5 Les variantes de l'OPF étudiées : $(OPF - L)$, $(OPF - Q)$ et $(OPF - UC)$ . . 43

0.2.5.1 OPF avec fonction de coût linéaire : $(OPF - L)$ 44
0.2.5.2 OPF avec fonction de coût quadratique : $(OPF - Q)$ 45
0.2.5.3 OPF avec Unit Commitment : $(OPF - UC)$ 45
0.2.6 Instances étudiées 46
0.3 Objectifs visés 47

1 État de l'art **53**
1.1 État de l'art sur l'OPF 54
1.1.1 Complexité de l'OPF 54
1.1.2 Calculs de solutions réalisables pour l'OPF : bornes supérieures 56
1.1.3 Relaxations de QCQP applicables à l'OPF : bornes inférieures 56
1.2 État de l'art sur la résolution exacte des QCQP 61
1.2.1 La hiérarchie de Lasserre pour résoudre des QCQP 63
1.2.2 Méthodes par algorithmes de *branch-and-bound* pour résoudre un QCQP . . . 65
1.2.3 La méthode MIQCR 71

2 La méthode `RC-OPF` **73**
2.1 Passage en variables réelles 75
2.2 Phase 1 : Reformulation quadratique optimale 77
2.2.1 Famille de reformulations quadratiques exactes 77
2.2.2 La relaxation du rang 78
2.2.3 Reformulation optimale 81
2.2.4 Extension aux contraintes de limitation du flux d'intensité 84
2.3 Comparaison avec la méthode MIQCR 85
2.4 Phase 2 : Algorithme de *branch-and-bound* spatial 89
2.4.1 Présentation générale de l'algorithme 89
2.4.2 Choix de la variable de branchement 91

2.4.3 En quelle valeur couper ? 93
2.4.4 Choix du nœud suivant 93
2.4.5 Autres paramètres 94
2.4.6 Notre méthode de résolution : `RC-OPF` 94
2.5 Illustration de `RC-OPF` sur l'instance WB2 96
2.5.1 Présentation de l'instance 96
2.5.2 Substitution de variables 98
2.5.3 Formulation du problème en variables complexes 98
2.5.4 Phase 0 : Réécriture en variables réelles 100
2.5.5 Phase 1.1 : Résolution de la relaxation du rang 101
2.5.6 Phase 1.2 : Construction de la reformulation quadratique optimale 102
2.5.7 Phase 2 : Résolution à l'aide de l'algorithme de *branch-and-bound* 102
2.6 Extension au problème $(OPF - Q)$ 103
2.6.1 Reformulation quadratique 103
2.6.2 Relaxation du rang 105
2.6.3 Reformulation optimale 108
2.6.3.1 Définition de la notion de reformulation optimale 108
2.6.3.2 Construction d'une reformulation optimale 109
2.6.4 Ajout des inégalités de McCormick 112
2.6.5 Conclusion sur la résolution de $(OPF - Q)$ avec `RC-OPF` 116

3 Contractions de bornes dans la méthode `RC-OPF` 119

3.1 Contractions fondées sur l'optimalité en phase 2 120
3.1.1 Contraction de bornes par résolution de problèmes convexes 121
3.1.2 Synergie avec la reformulation quadratique optimale 122
3.1.3 Contractions fondées sur les coûts réduits 122

3.1.4 Algorithme 122

3.2 Contractions fondées sur la faisabilité en phase 2 123

3.2.1 Contractions linéaires [1] 123

3.2.2 Contractions non-linéaires 123

3.2.3 Algorithme 125

3.3 Extension des contractions de bornes à $(OPF - Q)$ 128

4 Liens avec la méthode de la hiérarchie de Lasserre 129

4.1 Introduction 131

4.2 Bref rappel de la Hiérarchie de Lasserre 132

4.3 Reformulation et relaxation lagrangienne 134

4.3.1 Reformulation d'ordre d 134

4.3.2 Relaxation lagrangienne 136

4.3.3 Conditions pour une absence de saut de dualité 137

4.3.4 Équivalence primale-duale 138

4.4 Principal résultat 138

4.4.1 Reformulation de (P_d^2) 138

4.4.2 Le résultat 139

4.5 Conséquences 139

4.5.1 Reformulation « optimale » 139

4.5.2 Reformulation *Sum of Squares* 140

5 Expérimentations numériques avec `RC-OPF` 143

5.1 Caractéristiques des instances étudiées 144

5.2 Évaluation de `RC-OPF` sur des instances $(OPF - L)$ 146

5.2.1 Comparaison numérique de `RC-OPF` sans contractions de bornes avec le logiciel BARON 146

5.2.2 Impact des contractions de bornes 148

5.2.3 Impact du choix du paramètre δ 149

5.2.4 Résultats de `RC-OPF` avec contractions 151

5.3 Résultats obtenus avec `RC-OPF-Q` sur $(OPF - Q)$ 153

5.4 Génération d'instances difficiles 157

Conclusion **165**

Bibliographie **171**

Liste des annexes **176**

A Recherche d'instances difficiles **177**

Liste des tableaux

5.1 Caractéristiques des 42 instances étudiées. n est le nombre de nœuds du réseau électrique. n_p est le nombre de nœuds producteurs. 145

5.2 Comparaison entre `RC-OPF` sans contractions de bornes et BARON. Le gap d'optimalité relatif ϵ vaut 10^{-3}. Le temps limite est de 5 minutes. 147

5.3 Contractions des domaines des variables pour deux instances $(OPF - L)$ 149

5.4 Résultats pour 42 instances $(OPF - L)$ de `RC-OPF` avec OBBT-OPF et FBBT-OPF, 1h de temps de calcul et un gap d'optimalité relative valant 10^{-3}. 152

5.5 Résultats pour 42 instances $(OPF - Q)$ de `RC-OPF-Q` avec OBBT-OPF et FBBT-OPF et 1h de temps de calcul 155

5.6 Gaps de dualité fournis par la relaxation SDP pour 12 instances d'OPF 157

5.7 Intervalle des coefficients linéaires de production active 159

5.8 Résultats de `RC-OPF` avec les contractions de bornes sur 10 instances difficiles et un temps limite de 120h. 160

5.9 Mes contributions. 169

A.1 Résultats par instance. 1000 tirages par instance. 179

A.2 Résultats pour 9 instances aléatoires obtenues à partir de 39mod1. 185

A.3 Résultats pour 10 instances aléatoires obtenues à partir de 300mod. 190

Table des figures

1 Graphe représentant le réseau WB5 34

2 Le réseau WB5 avec les variables de puissance 36

3 Le réseau WB5 avec les variables de tension et d'intensité 37

4 Le réseau WB5 avec les données d'admittances 38

5 Le réseau WB5 avec les données de production et de consommation 39

6 Diagramme en Π de la ligne (i, j). 41

1.1 Graphe à 5 nœuds 60

1.2 Extension cordale 60

1.3 Décomposition en cliques de l'extension cordale 61

1.4 Exemple d'un algorithme de *branch-and-bound* spatial 67

2.1 Enveloppe convexe définie par les inégalités de McCormick de l'espace non-convexe (2.28) pour des bornes $\ell_i = \ell_j = -1$ et $u_i = u_j = 1$. 86

2.2 Schéma des actions effectuées en chaque nœud de l'arbre de l'algorithme 90

3.1 Anneau contenant les variables v_i et v_{i+n} 126

3.2 Exemple de réduction de bornes avec les réductions non linéaires 126

5.1 Nombre de nœuds nécessaires pour atteindre l'optimalité dans le cadre de la résolution de WB5 avec `RC-OPF`, en fonction de δ. On utilise FBBT-OPF comme contractions. . 150

5.2 Réduction du gap au cours des itérations pour les 10 instances difficiles créées à partir de l'instance 57 161

5.3 Gap final moyen pour les 10 instances difficiles à 57 nœuds après 12h de calculs 162

5.4 Synthèse des contributions principales 167

A.1 Evolution des bornes sur l'instance aléatoire 137 depuis WB5 182

A.2 Evolution des bornes sur l'instance aléatoire 195 depuis WB5 182

A.3 Evolution des bornes sur l'instance aléatoire 236 depuis 22loop 183

A.4 Evolution des bornes sur l'instance aléatoire 30 depuis 39mod1 184

A.5 Evolution des bornes sur l'instance aléatoire 211 depuis 39mod1 186

A.6 Evolution des bornes sur l'instance aléatoire 437 depuis 39mod1 186

A.7 Evolution des bornes sur l'instance aléatoire 537 depuis 39mod1 186

A.8 Evolution des bornes sur l'instance aléatoire 643 depuis 39mod1 187

A.9 Evolution des bornes sur l'instance aléatoire 712 depuis 39mod1 187

A.10 Evolution des bornes sur l'instance aléatoire 766 depuis 39mod1 187

A.11 Evolution des bornes sur l'instance aléatoire 776 depuis 39mod1 188

A.12 Evolution des bornes sur l'instance aléatoire 924 depuis 39mod1 188

A.13 Evolution des bornes sur l'instance aléatoire 537 depuis 39mod1 189

A.14 Evolution des bornes sur l'instance aléatoire 628 depuis 57 191

A.15 Evolution des bornes sur l'instance aléatoire 31 depuis 300mod 191

A.16 Evolution des bornes sur l'instance aléatoire 45 depuis 300mod 192

A.17 Evolution des bornes sur l'instance aléatoire 242 depuis 300mod 192

A.18 Evolution des bornes sur l'instance aléatoire 437 depuis 300mod 192

A.19 Evolution des bornes sur l'instance aléatoire 520 depuis 300mod 193

A.20 Evolution des bornes sur l'instance aléatoire 536 depuis 300mod 193

A.21 Evolution des bornes sur l'instance aléatoire 625 depuis 300mod 193

A.22 Evolution des bornes sur l'instance aléatoire 650 depuis 300mod 194

A.23 Evolution des bornes sur l'instance aléatoire 948 depuis 300mod 194

A.24 Evolution des bornes sur l'instance aléatoire 980 depuis 300mod 194

Notations

La liste suivante présente les notations générales que l'on retrouve tout au long de ce manuscrit.

— $\mathbb{N}$ est l'ensemble des entiers naturels,

— $\mathbb{N}^*$ est l'ensemble des entiers naturels non nuls,

— $\mathbb{R}$ est l'ensemble des nombres réels,

— $\mathbb{R}^+$ est l'ensemble des nombres réels positifs,

— $|x|$ est la valeur absolue du nombre réel x,

— $\min(a, b)$ est le minimum entre les réels a et b,

— $\max(a, b)$ est le maximum entre les réels a et b,

— v^t est le transposé du vecteur v de $\mathbb{R}^n$,

— $\mathcal{M}_n(\mathbb{R})$ est l'ensemble des matrices carrées de taille n à coefficients dans $\mathbb{R}$,

— M^t est la transposée de la matrice carrée M de $\mathcal{M}_n(\mathbb{R})$,

— $\mathcal{S}_n(\mathbb{R})$ est l'ensemble des matrices carrées symétriques de taille n de $\mathcal{M}_n(\mathbb{R})$,

— $rg(M)$ est le rang de la matrice M de $\mathcal{M}_n(\mathbb{R})$,

— $\mathrm{vp}_{\min}(M)$ est la valeur propre minimale de la matrice carrée M de $\mathcal{S}_n(\mathbb{R})$,

— $M \succeq 0$ signifie que la matrice M de $\mathcal{S}_n(\mathbb{R})$ est semi-définie positive,

— $< A, B >$ est le produit scalaire canonique entre deux matrices de $\mathcal{M}_n(\mathbb{R})$,

— $\mathcal{I}_n$ est la matrice identité de taille n de $\mathcal{M}_n(\mathbb{R})$,

— 0_n est la matrice nulle de taille n de $\mathcal{M}_n(\mathbb{R})$,

— $\mathbb{C}$ est l'ensemble des nombres complexes,

— $\mathbf{i}$ est le nombre complexe de partie réelle nulle et de partie imaginaire égale à 1,

— $Re(z)$ est la partie réelle du nombre complexe z,

— $Im(z)$ est la partie imaginaire du nombre complexe z,

— $ang(z)$ est la partie angulaire du nombre complexe z,

— z^* est le conjugué du nombre complexe z,

— $|z|$ est le module du nombre complexe z,

— v^* est le conjugué du transposé du vecteur v de $\mathbb{C}^n$,

— $\mathcal{M}_n(\mathbb{C})$ est l'ensemble des matrices carrées de taille n à coefficients dans $\mathbb{C}$,

— M^* est la conjuguée de la transposée de la matrice carrée M de $\mathcal{M}_n(\mathbb{C})$,

— $val(P)$ est la valeur optimale du problème d'optimisation (P).

— Γ_n^d est l'ensemble des monômes de degré d ou moins dans $\mathbb{R}^n$.

Le tableau suivant est un aide-mémoire contenant l'ensemble des notations adoptées pour les variables et les données d'un problème OPF.

Nom	Type	Description
$\mathcal{G}$	Donnée	Graphe représentant le réseau électrique
$\mathcal{N}$	Donnée	Ensemble des nœuds du réseau électrique
$\mathcal{P}$	Donnée	Ensemble des nœuds producteurs du réseau électrique
$\mathcal{C}$	Donnée	Ensemble des nœuds consommateurs du réseau électrique
$\mathcal{E}$	Donnée	Ensemble des arêtes du réseau électrique
$\mathbf{Y}_{i,j}$	Donnée	Admittance de la ligne (i,j)
$\mathbf{Y}_{i,j}^{gr}$	Donnée	Admittance à la masse de la ligne (i,j) au nœud i
$\mathbf{Y}_i$	Donnée	Admittance du nœud i
$\mathbf{Y}_i^M$	Donnée	Matrice fournissant la puissance injectée au nœud i en fonction des tensions V
$\mathbf{Y}_i^R$	Donnée	Matrice fournissant la puissance active injectée au nœud i en fonction des tensions v
$\mathbf{Y}_i^I$	Donnée	Matrice fournissant la puissance réactive injectée au nœud i en fonction des tensions v
$\mathbf{S}_i^c$	Donnée	Puissance électrique soutirée au nœud i
$\mathbf{p}_i^c$	Donnée	Puissance électrique active soutirée au nœud i
$\mathbf{q}_i^c$	Donnée	Puissance électrique réactive soutirée au nœud i
$\underline{\mathbf{p}_i}$	Donnée	Puissance active minimale produite au nœud producteur i
$\overline{\mathbf{p}_i}$	Donnée	Puissance active maximale produite au nœud producteur i
$\underline{\mathbf{q}_i}$	Donnée	Puissance réactive minimale produite au nœud producteur i
$\overline{\mathbf{q}_i}$	Donnée	Puissance réactive maximale produite au nœud producteur i
$\underline{\mathbf{v}_i}$	Donnée	Module de tension minimal au nœud i
$\overline{\mathbf{v}_i}$	Donnée	Module de tension maximal au nœud i
$\mathbf{I}_{i,j}^{max}$	Donnée	Intensité électrique maximale sur la ligne (i,j)
$\mathbf{S}_{i,j}^{max}$	Donnée	Puissance électrique maximale sur la ligne (i,j)
$\mathbf{P}_{i,j}^{max}$	Donnée	Puissance électrique active maximale sur la ligne (i,j)
$\mathbf{v}_{i,j}^{max}$	Donnée	Module de différence de tension électrique maximal sur la ligne (i,j)
$\boldsymbol{\theta}_{i,j}^{max}$	Donnée	Différence angulaire maximale sur la ligne (i,j)
$\mathbf{c}_{0,i}$	Donnée	Coût fixe sur la puissance active injectée au nœud producteur i
$\mathbf{c}_{1,i}$	Donnée	Coût linéaire sur la puissance active injectée au nœud producteur i
$\mathbf{C}_{2,i}$	Donnée	Coût quadratique sur la puissance active injectée au nœud producteur i
$\mathbf{c}_0$	Donnée	Coût fixe total du réseau, utilisée pour $(OPF-L)$
$\mathbf{C}_1$	Donnée	Matrice de coût linéaire, utilisée pour $(OPF-L)$
$\mathbf{c}_0^q$	Donnée	Coût fixe total du réseau, utilisée pour $(OPF-Q)$
$\mathbf{c}_1$	Donnée	Vecteur des coûts linéaires, utilisé pour $(OPF-Q)$
$\mathbf{C}_2$	Donnée	Matrice de coût quadratique, utilisée pour $(OPF-Q)$
S_i	Variable	Puissance électrique injectée au nœud i par l'extérieur du réseau
$S_{i,j}$	Variable	Puissance électrique partant du nœud i vers le nœud j
p_i	Variable	Puissance active produite au nœud producteur i
p	Variable	Vecteur des $(p_i)_{i\in\mathcal{P}}$
I_i	Variable	Intensité électrique fournie par ce qui est extérieur au nœud i
$I_{i,j}$	Variable	Intensité électrique partant du nœud i vers le nœud j
V_i	Variable	Tension électrique au nœud i
V	Variable	Vecteur des $(V_i)_{i\in\mathcal{N}}$
v	Variable	Vecteur des parties réelle et imaginaire des $(V_i)_{i\in\mathcal{N}}$
b_i	Variable	Binaire représentant l'allumage ou non du générateur au nœud producteur i

Introduction

0.1 Présentation du problème de l'optimisation des flux de puissance (OPF)

0.1.1 Généralités sur la gestion des réseaux électriques

RTE (Réseau de Transport d'Électricité), en tant que gestionnaire du réseau de transport d'électricité français, est responsable du bon fonctionnement de ce dernier. Les différentes grandeurs physiques intervenant dans un réseau électrique sont la tension, l'intensité, ou courant, et la puissance. Comme un réseau de transport haute tension fonctionne en courant alternatif, toutes ces grandeurs sont des fonctions sinusoïdales du temps en régime permantent, c'est-à-dire une fois qu'elles se sont stabilisées. Quand un événement entraine un important changement sur le réseau, le régime permanent est perturbé et ces grandeurs entrent en régime transitoire. Un tel événement se produit par exemple quand une ligne haute tension est mise hors service à la suite d'un incident comme un orage.

Le problème de la stabilité consiste, pour RTE, à ne pas se retrouver dans une situtation où un régime transitoire ne se stabiliserait pas en un nouveau régime permanent. Le problème de la tenue du « N-1 » consiste à s'assurer que la perte de l'une des lignes du réseau ne va pas entrainer une perte de stabilité. Dans ce manuscrit, nous ne nous intéressons pas plus aux problèmatiques de stabilité du réseau. Nous restons dans l'étude et dans le cadre du régime permanent en considérant que notre réseau saura se stabiliser sur ce dernier.

0.1.2 Le problème OPF

Le problème de l'Optimisation des Flux de Puissance est appelé en anglais « Optimal Power Flow » et abrégé en OPF. C'est sous ce sigle que l'on désignera le problème de l'Optimisation des Flux de Puissance dans ce manuscrit. L'OPF est un problème d'optimisation qui intervient dans un réseau électrique. Il consiste à déterminer la puissance à produire, ainsi que les flux de puissance à transporter sur le réseau, pour couvrir, à un coût minimal, la puissance consommée.

À travers l'expression réseau électrique, on entend une infrastructure regroupant des postes électriques reliés entre eux par des lignes électriques. Ces dernières sont des ouvrages transportant un courant électrique sous haute tension, et les postes sont des édifices vers lesquels convergent plusieurs

lignes. Ainsi, dans un réseau électrique, le courant électrique se déplace entre les postes au moyen des lignes. Au niveau d'un poste on trouve divers équipements permettant notamment d'abaisser ou de réhausser la tension pour passer par exemple d'une ligne 400kV à une ligne 220kV.

RTE cherche, entre autres, à minimiser les pertes de puissance sur le réseau de transport d'électricité français. La minimisation des pertes de puissance sur un réseau électrique se représente comme une instance du problème OPF. Dans la suite on s'intéresse à la résolution du problème OPF dans un réseau de transport d'électricité. Le cas des réseaux de transport se démarque des réseaux de distribution par le fait que le niveau de tension considéré pour les différents ouvrages est plus élevé. De plus, dans un réseau de transport, les lignes bouclent sur les postes électriques du réseau, formant des cycles. Ceci découle de la règle connue sous le nom de règle du N-1, stipulant que le réseau doit continuer à fonctionner en cas de perte d'une des lignes. Les réseaux de distribution d'électricité ont au contraire une structure arborescente. Cette propriété de bouclage augmente fortement la difficulté du problème OPF.

Dans le cadre de ce manuscrit, on étudie des réseaux de transport d'électricité, c'est-à-dire des réseaux contenant des cycles. Un poste est le point de rencontre entre plusieurs lignes du réseau, c'est aussi un point d'échange de puissance avec l'extérieur du réseau. L'extérieur du réseau peut représenter des producteurs ou des consommateurs d'électricité. La puissance consommée en chaque poste est connue et est une donnée du problème OPF. La puissance produite est elle une variable du problème d'optimisation. L'objectif est de déterminer la puissance à produire par les acteurs de production, ainsi que les flux de puissance sur les lignes entre les postes, pour couvrir la consommation en chaque poste. Tout cela en minimisant les coûts de production pour lesdits acteurs.

0.2 Modélisation mathématique de l'OPF et variantes $(OPF - L)$, $(OPF - Q)$ et $(OPF - UC)$

Cette section présente une modélisation mathématique de l'OPF. Dans la suite de ce manuscrit on travaille sur différentes variantes de l'OPF. La nature et les modélisations de ces variantes sont également présentées dans ce chapitre.

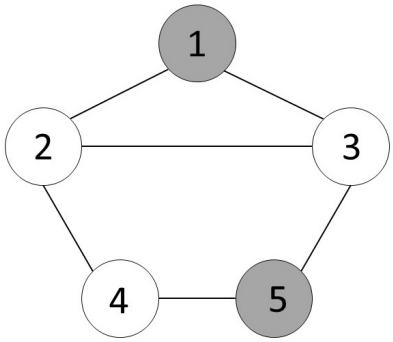

FIGURE 1 – Graphe représentant le réseau WB5

0.2.1 Représentation du réseau électrique par un graphe

Tout d'abord, en chaque poste du réseau, on peut définir des nœuds qui sont les points de jonction entre plusieurs lignes. Cette modélisation du réseau permet de schématiser ce dernier comme un graphe $\mathcal{G}(\mathcal{N}, \mathcal{E})$. Les nœuds de ce graphe sont alors les nœuds du réseau, et des nœuds différents peuvent se trouver à l'intérieur d'un même poste électrique. Les arêtes du graphe sont les lignes reliant ces nœuds.

On divise les nœuds du réseau en deux catégories en fonction de la nature de leurs échanges avec l'extérieur : ceux qui soutirent de la puissance électrique depuis un producteur d'énergie (centrales thermiques, parcs éoliens, etc.) tout en fournissant une puissance à d'éventuels consommateurs. et ceux qui ne font que fournir de la puissance à l'extérieur pour alimenter des consommateurs d'énergie électrique (par exemple un réseau de distribution). Par abus de langage, on parle de nœuds producteurs ou de nœuds reliés à des producteurs pour ceux appartenant à la première catégorie. Les nœuds appartenant à la seconde catégorie sont qualifiés de nœuds consommateurs ou de nœuds reliés à des consommateurs. $\mathcal{P}$ est l'ensemble des nœuds producteurs du réseau, tandis que $\mathcal{C}$ est l'ensemble des nœuds consommateurs. On notera bien que les nœuds producteurs peuvent avoir une activité de consommation en plus de leur production, mais que les nœuds consommateurs n'ont pas d'activité de production.

La Figure 1 représente un réseau fictif à 5 nœuds appelé WB5, créé par Wakash Bukhsh *et al.* [2]. Ce réseau est constitué de 5 nœuds, numérotés de 1 à 5, ainsi que de 6 lignes. Les nœuds 1 et 5 (en gris) sont reliés à des producteurs et injectent de la puissance dans le réseau. Les nœuds 2, 3 et 4 (en

blanc) sont reliés à des consommateurs et soutirent de la puissance au réseau.

0.2.2 Variables du problème

Pour présenter les variables du problème on introduit dans un premier temps certaines grandeurs physiques des réseaux électriques : la puissance, l'intensité et la tension. Dans le cadre de cette thèse, nous recherchons une solution en régime permanent et en courant alternatif. Ces grandeurs sont donc des fonctions sinusoïdales du temps oscillant à une fréquence de 50Hz. Elles se caractérisent par leur amplitude et leur déphasage à l'origine. Ceci amène à représenter les différentes grandeurs du réseau par des nombres complexes dont le module représente l'amplitude, et l'argument le déphasage à l'origine. Dans ce qui suit on utilise souvent les termes actif et réactif au sujet de la puissance. Le terme actif fait référence à la partie réelle d'une puissance complexe, tandis que le terme réactif fait référence à sa partie imaginaire.

L'OPF s'intéressant aux flux de puissance dans un réseau, on introduit en premier lieu les variables de puissance. Soit i et j deux nœuds du réseau reliés par une ligne. $S_{i,j} \in \mathbb{C}$ représente la puissance électrique partant du nœud i vers le nœud j. $S_{i,j}$ est *a priori* différent de $-S_{j,i}$ car il y a des pertes de puissance sur les lignes dues à de phénomènes physiques comme l'effet Joule. On remarque que les flux de puissance dans le réseau sont non-conservatifs. On définit aussi la puissance injectée en un nœud i que l'on note $S_i \in \mathbb{C}$. Elle correspond à la puissance fournie au nœud i par ce qui est extérieur au réseau. Ainsi pour les nœuds reliés à des producteurs, elle correspond à la puissance apportée par ces derniers, moins la puissance soutirée par les éventuels consommateurs présents en ce nœud. Pour les nœuds reliés à des consommateurs, elle correspond à l'opposé de la puissance soutirée par ceux-ci. On sous-entend dans ce manuscrit que le soutirage est l'action opposée à l'injection. Pour simplifier les notations, on introduit $(p_i)_{i\in\mathcal{P}}$ qui est la puissance active produite au nœud producteur i. C'est une autre variable du problème qui s'exprime directement à l'aide de la puissance S_i et des données du problème, voir la section 0.2.4 pour la formule précise. La Figure 2 représente le réseau WB5 avec les variables de puissance traversant les lignes ainsi que les puissances injectées.

Les deux autres grandeurs fondamentales des réseaux électriques sont la tension et l'intensité. Cette dernière représente le flux de charges électriques le long d'une ligne. De manière analogue à la puissance

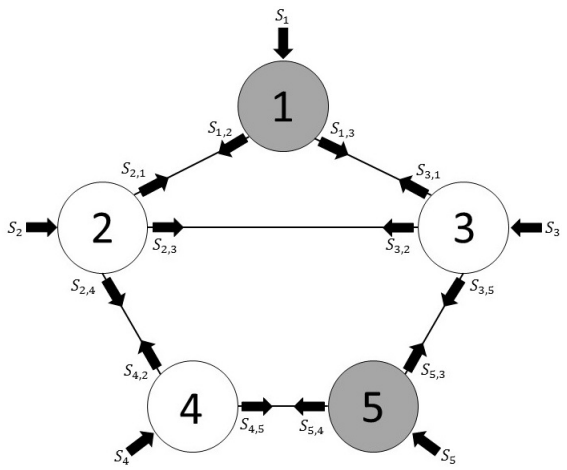

FIGURE 2 – Le réseau WB5 avec les variables de puissance

on définit $I_{i,j} \in \mathbb{C}$ représentant l'intensité partant du nœud i vers le nœud j sur la ligne (i, j). *A priori* $I_{i,j} \neq -I_{j,i}$, ceci provient des impédances à la masse d'une ligne qui peuvent être différentes de part et d'autre de celle-ci. On définit aussi l'intensité injectée en chaque nœud i, I_i comme l'intensité fournie par l'extérieur au réseau au nœud i. La tension est le potentiel électrique s'exerçant en un point. En chaque nœud i on définit une tension $V_i \in \mathbb{C}$. La Figure 3 représente le réseau WB5 avec les variables d'intensité traversant les lignes, les variables d'intensité injectées aux nœuds ainsi que les tensions aux nœuds.

En résumé, résoudre l'OPF consiste à déterminer les valeurs des grandeurs physiques suivantes qui sont les variables du problème :

— $S_{i,j} \in \mathbb{C} \quad \forall (i, j) \in \mathcal{E}$,

— $S_i \in \mathbb{C} \quad \forall i \in \mathcal{N}$,

— $p_i \in \mathbb{R} \quad \forall i \in \mathcal{P}$,

— $I_{i,j} \in \mathbb{C} \quad \forall (i, j) \in \mathcal{E}$,

— $I_i \in \mathbb{C} \quad \forall i \in \mathcal{N}$,

— $V_i \in \mathbb{C} \quad \forall i \in \mathcal{N}$.

Une fois ces valeurs déterminées, on connait l'état du réseau en régime permanent.

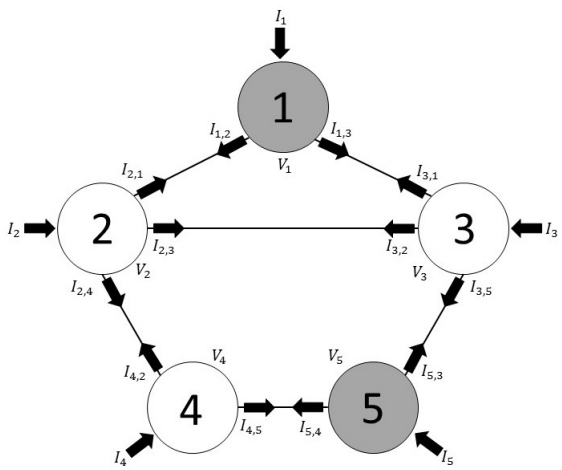

FIGURE 3 – Le réseau WB5 avec les variables de tension et d'intensité

0.2.3 Données du problème

Une instance d'un problème d'OPF est caractérisée par un certain nombre de données. Dans un premier temps, le réseau se caractérise par une topologie modélisée par le graphe $\mathcal{G}(\mathcal{N}, \mathcal{E})$. De plus, chaque ligne $(i, j) \in \mathcal{E}$ est caractérisée par une admittance $\mathbf{Y}_{i,j} \in \mathbb{C}$ et deux admittances à la masse $\mathbf{Y}^{gr}_{i,j} \in \mathbb{C}$ et $\mathbf{Y}^{gr}_{j,i} \in \mathbb{C}$. L'admittance est la généralisation complexe de la conductance, et représente la capacité à conduire l'électricité. L'admittance ne dépend que des propriétés de la ligne (longueur, matériaux la composant, etc.), donc $\mathbf{Y}_{i,j} = \mathbf{Y}_{j,i}$. *A priori* les deux admittances à la masse d'une ligne ne sont pas égales. Elles dépendent des interactions de la ligne avec le sol, au niveau des pylônes par exemple. Chaque nœud i dispose aussi d'une admittance $\mathbf{Y}_i \in \mathbb{C}$ modélisant les équipements que l'on trouve au niveau de ce nœud. La topologie et les admittances aux lignes et aux nœuds sont les données physiques du problème permettant de caractériser le réseau considéré. La Figure 4 représente le réseau WB5 avec les données d'admittance.

Un deuxième type de données correspond aux données de production et de consommation aux nœuds. Aux nœuds reliés à des consommateurs, on connait la puissance que ces consommateurs vont soutirer au réseau. Pour un nœud consommateur $i \in \mathcal{C}$, on note $\mathbf{S}^c_i \in \mathbb{C}$ la puissance soutirée par ce nœud au réseau. Aux nœuds reliés à des producteurs, on connait des bornes de fonctionnement sur les puissances active et réactive injectées au réseau. C'est-à-dire que l'on connait : $\underline{\mathbf{p}_i} \in \mathbb{R}$ et $\overline{\mathbf{p}_i} \in \mathbb{R}$ puissances actives respectivement minimale et maximale injectées au nœud $i \in \mathcal{P}$, ainsi que $\underline{\mathbf{q}_i} \in \mathbb{R}$ et

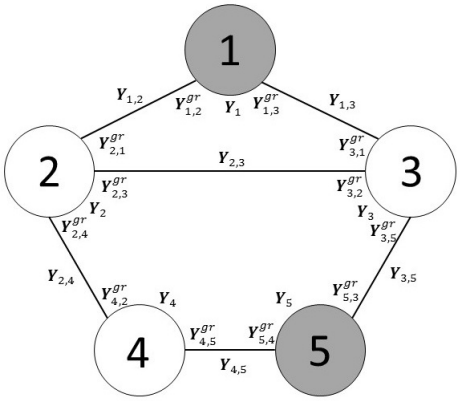

FIGURE 4 – Le réseau WB5 avec les données d'admittances

$\underline{\mathbf{q}_i} \in \mathbb{R}$ puissances réactives respectivement minimale et maximale injectées. De plus, on peut trouver des consommateurs raccordés aux nœuds de production. La puissance soutirée en un nœud producteur $i \in \mathcal{P}$ est notée $\mathbf{S}_i^c \in \mathbb{C}$.

On connait aussi le coût de production associé à la puissance active injectée en un nœud. Cela regroupe, pour chaque nœud $i \in \mathcal{P}$, un coût fixe $\mathbf{c}_{0i} \in \mathbb{R}^+$, un coût linéaire $\mathbf{c}_{1i} \in \mathbb{R}^+$ et un coût quadratique $\mathbf{C}_{2i} \in \mathbb{R}^+$. La Figure 5 représente le réseau WB5 avec les données de production et de consommation en chaque nœud.

Le dernier type de données correspond aux limites de fonctionnement des infrastructures présentes sur le réseau. Premièrement en chaque nœud $i \in \mathcal{N}$ on trouve des bornes $\underline{\mathbf{v}_i} \in \mathbb{R}^+$ et $\overline{\mathbf{v}_i} \in \mathbb{R}^+$ sur le module de la tension en ce nœud. Deuxièmement sur chaque ligne $(i,j) \in \mathcal{E}$ on trouve une limite de flux, qui est la capacité de la ligne, et qui dépend de plusieurs paramètres physiques et géographiques, comme la température. Cette capacité est approchée de plusieurs manières par la communauté scientifique, qui ne sont pas équivalentes et n'ont pas la même signification physique :

— Une intensité maximale sur la ligne notée $\mathbf{I}_{i,j}^{max} \in \mathbb{R}^+$,

— Une puissance maximale sur la ligne notée $\mathbf{S}_{i,j}^{max} \in \mathbb{R}^+$,

— Une puissance active maximale sur la ligne notée $\mathbf{P}_{i,j}^{max} \in \mathbb{R}^+$,

— Un maximum du module de la différence de tension sur la ligne noté $\mathbf{v}_{i,j}^{max} \in \mathbb{R}^+$.

Selon le choix fait, on obtient une modélisation différente de l'OPF. Troisièmement, on peut aussi

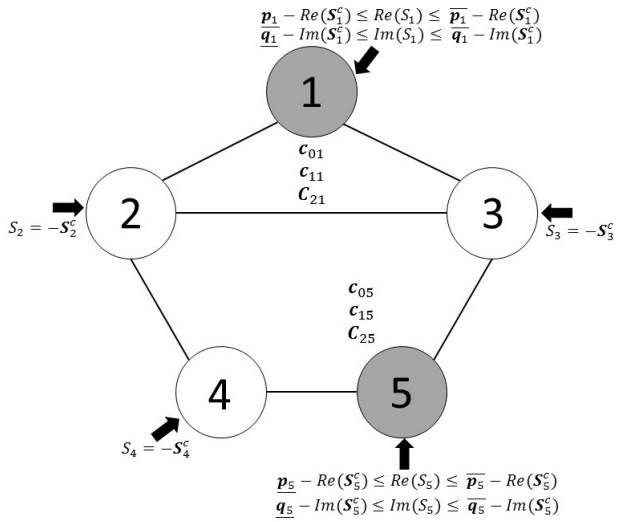

FIGURE 5 – Le réseau WB5 avec les données de production et de consommation

trouver une différence angulaire maximale entre les phases des tensions aux extrémités d'une ligne notée $\boldsymbol{\theta}_{i,j}^{max} \in \mathbb{R}$.

En résumé les données du problèmes sont :

— Le graphe $\mathcal{G}(\mathcal{N}, \mathcal{E})$,

— $\mathcal{P}$ et $\mathcal{C}$ deux sous ensembles de $\mathcal{N}$ tels que : $\mathcal{N} = \mathcal{P} \cup \mathcal{C}$,

— $\mathbf{Y}_{i,j} \in \mathbb{C} \quad \forall (i,j) \in \mathcal{E}$,

— $\mathbf{Y}_{i,j}^{gr} \in \mathbb{C} \quad \forall (i,j) \in \mathcal{E}$,

— $\mathbf{Y}_{j,i}^{gr} \in \mathbb{C} \quad \forall (i,j) \in \mathcal{E}$,

— $\mathbf{Y}_i \in \mathbb{C} \quad \forall i \in \mathcal{N}$,

— $\mathbf{S}_i^c \in \mathbb{C} \quad \forall i \in \mathcal{N}$,

— $\underline{\mathbf{p}_i} \in \mathbb{R} \quad \forall i \in \mathcal{P}$,

— $\overline{\mathbf{p}_i} \in \mathbb{R} \quad \forall i \in \mathcal{P}$,

— $\underline{\mathbf{q}_i} \in \mathbb{R} \quad \forall i \in \mathcal{P}$,

— $\overline{\mathbf{q}_i} \in \mathbb{R} \quad \forall i \in \mathcal{P}$,

— $\mathbf{c}_{0i} \in \mathbb{R}^+ \quad \forall i \in \mathcal{P}$,

— $\mathbf{c}_{1i} \in \mathbb{R}^+ \quad \forall i \in \mathcal{P}$,

— $\mathbf{C}_{2i} \in \mathbb{R}^+ \quad \forall i \in \mathcal{P}$,

— $\underline{\mathbf{v}_i} \in \mathbb{R}^+ \quad \forall i \in \mathcal{N}$,

— $\overline{\mathbf{v}_i} \in \mathbb{R}^+ \quad \forall i \in \mathcal{N}$,

— Pour chaque ligne $(i,j) \in \mathcal{E}$, une ou plusieurs des données de limite de flux sur cette ligne,

— De manière optionnelle, on peut trouver sur une ligne $(i,j) \in \mathcal{E}$: $\boldsymbol{\theta}_{i,j}^{max} \in \mathbb{R}$.

0.2.4 Fonction objectif et contraintes

Dans ce problème on cherche à minimiser les coûts de production permettant de couvrir la puissance consommée. La fonction objectif (fonction de coût) associée est une fonction quadratique de la puissance active injectée en chaque nœud générateur p_i. p_i est liée à S_i par la relation : $p_i = Re(S_i) + Re(\mathbf{S}_i^c)$.

$$f_{obj}(p) = \sum_{i \in \mathcal{P}} \mathbf{C}_{2i} p_i^2 + \mathbf{c}_{1i} p_i + \mathbf{c}_{0i}. \tag{1}$$

En prenant $\mathbf{C}_{2i} = 0$ et $\mathbf{c}_{1i} = 1$ pour tout i, on minimise les pertes sur le réseau électrique.

On retrouve trois grands types de contraintes dans le problème OPF. Les premières couplent les différentes grandeurs physiques. Ces contraintes proviennent des lois de la physique des réseaux électriques. Elles sont utilisées pour substituer des variables à d'autres, ou pour garder la cohérence physique entre les différentes variables utilisées dans la modélisation choisie d'un problème OPF. Le deuxième type de contraintes assure l'équilibre offre-demande, c'est-à-dire fixe la puissance consommée et encadre la puissance produite. Enfin le troisième type encadre le module de la tension en chaque nœud, limite le flux le long des lignes ainsi que l'écart angulaire.

Définition de la puissance On donne ici la définition physique des variables complexes de puissance afin de relier puissance, tension et intensité. La puissance s'exprime comme un produit entre tension et intensité. Plus précisément, les puissances $S_{i,j}$ et S_i sont définies comme suit :

$$S_{i,j} = V_i I_{i,j}^* \quad \forall (i,j) \in \mathcal{E}, \tag{2}$$

$$S_i = V_i I_i^* \quad \forall i \in \mathcal{N}. \tag{3}$$

Ce lien non linéaire entre ces trois quantités est au fondement du caractère non linéaire et non convexe du problème d'optimisation qu'est l'OPF.

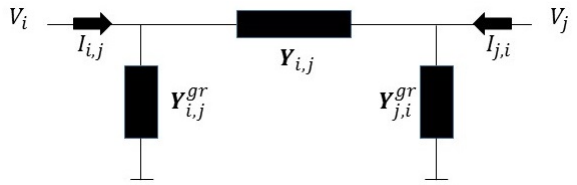

FIGURE 6 – Diagramme en Π de la ligne (i, j).

Lois des nœuds La loi des nœuds lie les intensités sur les lignes aux intensités injectées aux nœuds de la façon suivante : Soit $N_{\mathcal{G}}(i)$ l'ensemble des nœuds voisins des nœud i dans le graphe $\mathcal{G}$.

$$I_i = \sum_{j \in N_{\mathcal{G}}(i)} I_{i,j} + \mathbf{Y}_i V_i. \tag{4}$$

Puisque $S_i = V_i I_i^*$ on peut écrire la loi des nœuds sur S_i :

$$S_i = \sum_{j \in N_{\mathcal{G}}(i)} S_{i,j} + \mathbf{Y}_i^* V_i^* V_i. \tag{5}$$

Les lois des nœuds permettent donc d'écrire les quantités injectées aux nœuds à partir des quantités transitant sur les lignes et des tensions aux nœuds.

Loi d'Ohm La loi d'Ohm permet de relier les tensions aux intensités. Plus précisément elle indique que l'intensité sur une section est le produit de l'admittance de cette section par la différence entre les tensions aux extrémités de cette dernière. On modélise une ligne électrique $(i, j) \in \mathcal{E}$ selon le schéma de la Figure 6 avec ce qu'on appelle un diagramme en Π. Ceci nous fournit l'équation suivante :

$$I_{i,j} = \mathbf{Y}_{i,j}(V_i - V_j) + V_i \mathbf{Y}^{gr}_{i,j} \quad \forall (i, j) \in \mathcal{E}. \tag{6}$$

En utilisant la loi d'Ohm (6) et la loi des nœuds (4), on est capable de remplacer chaque occurence d'une variable I_i par son expression linéaire selon les variables de tension V. En ajoutant la définiton de la puissance $S_i = V_i I_i^*$, on peut remplacer totalement les occurences des variables de puissance S par leur expression en fonction des variables de tension V. Les puissances s'expriment de manière « quadratique », au sens complexe, en les variables de tension.

Équilibre offre-demande Les contraintes d'équilibre offre-demande expriment le fait que la puissance soutirée au nœud de consommation i est fixée et vaut $\mathbf{S}_i^c$, tandis qu'aux nœuds de production,la

puissance injectée est bornée. Les contraintes d'équilibre d'offre-demande se modélisent comme ceci :

$$S_i = -\mathbf{S}_i^c, \quad \forall i \in \mathcal{C}, \tag{7}$$

$$\underline{\mathbf{p}_i} \leq p_i \leq \overline{\mathbf{p}_i}, \quad \forall i \in \mathcal{P}, \tag{8}$$

$$\underline{\mathbf{q}_i} - Im(\mathbf{S}_i^c) \leq Im(S_i) \leq \overline{\mathbf{q}_i} - Im(\mathbf{S}_i^c), \quad \forall i \in \mathcal{P}. \tag{9}$$

Contraintes de limite technique La dernière famille de contraintes correspond aux contraintes de limite technique dont la première catégorie correspond aux bornes sur les modules de tension aux nœuds. Ces contraintes sont formalisées ainsi :

$$\underline{\mathbf{v}_i} \leq |V_i| \leq \overline{\mathbf{v}_i}, \quad \forall i \in \mathcal{N}. \tag{10}$$

La deuxième catégorie correspond aux limites de flux sur les lignes. Comme expliqué plus haut, cette limite peut être considérée de plusieurs manières. La liste suivante donne les contraintes obtenues avec chaque modélisation :

— Avec une intensité maximale :

$$|I_{i,j}| \leq \mathbf{I}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E}, \tag{11}$$

— Avec une puissance maximale :

$$|S_{i,j}| \leq \mathbf{S}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E}, \tag{12}$$

— Avec une puissance active maximale :

$$|Re(S_{i,j})| \leq \mathbf{P}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E}, \tag{13}$$

— Avec une différence de tension maximale :

$$|V_i - V_j| \leq \mathbf{v}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E}. \tag{14}$$

Enfin une troisième catégorie correspond aux limites de différence angulaire entre deux nœuds adjacents du réseau. Les contraintes de cette catégorie s'écrivent :

$$|ang(V_i) - ang(V_j)| \leq \boldsymbol{\theta}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E}. \tag{15}$$

0.2.5 Les variantes de l'OPF étudiées : $(OPF-L)$, $(OPF-Q)$ et $(OPF-UC)$

En rassemblant les variables, les données et les contraintes présentées ci-dessus, il est possible d'obtenir une formulation générique de l'OPF. Cependant, dans le cadre de ce manuscrit, on ne va considérer que certaines variantes de l'OPF, qui diffèrent de cette formulation générique comme précisé ci-dessous.

Dans tout ce qui suit, on note n le cardinal de $\mathcal{N}$, le nombre de nœuds dans le réseau étudié, numérotés de 1 à n. On note également $V \in \mathbb{C}^n$ le vecteur des variables de tension. Pour toutes les variantes étudiées, les substitutions suivantes sont faites :

— Les variables d'intensité sont remplacées par celles de tension :

$$\forall (i,j) \in \mathcal{E}, I_{i,j} = (\mathbf{Y}_{i,j} + \mathbf{Y}_{i,j}^{gr})V_i - \mathbf{Y}_{i,j}V_j, \tag{16}$$

$$\forall i \in \mathcal{N}, I_i = \left[\sum_{j \in N_{\mathcal{G}}(i)} (\mathbf{Y}_{i,j} + \mathbf{Y}_{i,j}^{gr}) + \mathbf{Y}_i \right] V_i - \sum_{j \in N_{\mathcal{G}}(i)} \mathbf{Y}_{i,j}V_j, \tag{17}$$

— Les variables représentant les puissances s'écoulant d'un nœud vers un autre nœud sont remplacées par celles de tension :

$$\forall (i,j) \in \mathcal{E}, S_{i,j} = (\mathbf{Y}_{i,j}^* + \mathbf{Y}_{i,j}^{gr*})V_i^*V_i - \mathbf{Y}_{i,j}^*V_j^*V_i, \tag{18}$$

— Les variables représentant les puissances échangées avec l'extérieur en chaque nœud consommateur sont remplacées par celles de tension :

$$\forall i \in \mathcal{C}, S_i = \left[\sum_{j \in N_{\mathcal{G}}(i)} (\mathbf{Y}_{i,j}^* + \mathbf{Y}_{i,j}^{gr*}) + \mathbf{Y}_i^* \right] V_i^*V_i - \sum_{j \in N_{\mathcal{G}}(i)} \mathbf{Y}_{i,j}^*V_j^*V_i. \tag{19}$$

Notons que la puissance échangée avec l'extérieur aux nœuds producteurs s'exprime de la même manière que celle aux nœuds consommateurs. Pour chaque nœud $i \in \mathcal{N}$, on peut donc définir une matrice complexe $\mathbf{Y}_i^M \in \mathcal{M}_n(\mathbb{C})$, telle que $S_i = V^*\mathbf{Y}_i^M V$. $\mathbf{Y}_i^M$ contient des composantes nulles, à l'exception des entrées suivantes :

— $(i,i) : \sum_{j \in N_{\mathcal{G}}(i)} (\mathbf{Y}_{i,j}^* + \mathbf{Y}_{i,j}^{gr*}) + \mathbf{Y}_i^*$,

— $\forall j \in N_{\mathcal{G}}(i), (j,i) : \mathbf{Y}_{i,j}^*$

On remarquera également que pour toute matrice $M \in \mathcal{M}_n(\mathbb{C})$ et pour tout $V \in \mathbb{C}^n$:

$$Re(V^*MV) = V^* \left(\frac{M + M^*}{2} \right) V, \tag{20}$$

$$Im(V^*MV) = V^*\left(\frac{M - M^*}{2\mathbf{i}}\right)V. \tag{21}$$

En outre, les matrices $\frac{M+M^*}{2}$ et $\frac{M-M^*}{2\mathbf{i}}$ sont hermitiennes [1]. On en déduit que :

$$\forall i \in \mathcal{N}, Re(S_i) = Re(V^*\mathbf{Y}_i^M V) = V^*\left(\frac{\mathbf{Y}_i^M + \mathbf{Y}_i^{M*}}{2}\right)V, \tag{22}$$

$$\forall i \in \mathcal{N}, Im(S_i) = Im(V^*\mathbf{Y}_i^M V) = V^*\left(\frac{\mathbf{Y}_i^M - \mathbf{Y}_i^{M*}}{2\mathbf{i}}\right)V. \tag{23}$$

De plus, on ne considère aucune contrainte de limite de flux sur les lignes, ni de différence angulaire maximale entre des nœuds adjacents. Ceci est pertinent parce que dans les cas tests de RTE, ces contraintes sont absentes. En effet, les contraintes de limite de flux sont souvent inactives dans les réseaux de transport car ceux-ci sont dimensionnés pour que ce type de contraintes ne soient pas actives en fonctionnement normal. Ceci découle de la nécessité de maintenir la stabilité du réseau en cas de pertes d'une des lignes. Cette règle est appelée la règle du « N-1 ».

0.2.5.1 OPF avec fonction de coût linéaire : $(OPF - L)$

La première variante est celle où la fonction de coût considérée est une fonction affine en les variables de puissance active p, par abus de langage on parle de coût linéaire, en opposition à un coût quadratique. Dans le cadre de cette variante, on remplace les variables de puissance échangée avec l'extérieur en chaque nœud producteur par les variables de tension. Les seules variables considérées sont alors celles de tension, qui permettent, une fois connues, de calculer toutes les autres grandeurs du réseau. On obtient le programme quadratique en variables complexes suivant pour $(OPF - L)$:

$$(OPF - L)\begin{cases} \min\limits_{V} \quad \sum\limits_{i\in\mathcal{P}} \mathbf{c}_{1i}V^*\left(\frac{\mathbf{Y}_i^M + \mathbf{Y}_i^{M*}}{2}\right)V + \mathbf{c}_{1i}Re(\mathbf{S}_i^c) + \mathbf{c}_{0i} \\ \text{s.t.} \\ V^*\mathbf{Y}_i^M V = -\mathbf{S}_i^c, \quad i \in \mathcal{C} & (24) \\ \underline{\mathbf{p}_i} - Re(\mathbf{S}_i^c) \leq V^*\left(\frac{\mathbf{Y}_i^M + \mathbf{Y}_i^{M*}}{2}\right)V \leq \overline{\mathbf{p}_i} - Re(\mathbf{S}_i^c), \quad i \in \mathcal{P} & (25) \\ \underline{\mathbf{q}_i} - Im(\mathbf{S}_i^c) \leq V^*\left(\frac{\mathbf{Y}_i^M - \mathbf{Y}_i^{M*}}{2\mathbf{i}}\right)V \leq \overline{\mathbf{q}_i} - Im(\mathbf{S}_i^c), \quad i \in \mathcal{P} & (26) \\ (10) \\ V_i \in \mathbb{C}, \quad i \in \mathcal{N}. \end{cases}$$

1. Une matrice M de $\mathcal{M}_n(\mathbb{C})$ est dite hermitienne si et seulement si $M^* = M$

0.2.5.2 OPF avec fonction de coût quadratique : $(OPF - Q)$

La deuxième variante ajoute des coûts quadratiques à la précédente. Un coût quadratique en plus d'un coût linéaire permet d'obtenir une meilleure approximation du coût réel de production de puissance active pour le détenteur du moyen de production. La fonction objectif est alors un polynôme de degré 2 en les variables de puissance active p et correspond à la fonction f_{obj} décrite en section 0.2.4. Dans cette variante, on remplace les variables de puissance réactive échangée avec l'extérieur en chaque nœud producteur par les variables de tension, mais on garde les variables de puissance active injectée par chaque générateur. Cette modélisation avec des variables complexes n'est pas utilisée telle quelle dans le cadre de ce travail, mais est transformée en une modélisation en variables réelles dans la section 2.6. On obtient le programme mathématique suivant pour $(OPF - Q)$:

$$(OPF - Q) \begin{cases} \min\limits_{V,p} \quad f_{obj}(p) \\ \text{s.t.} \\ \quad (24), (8), (26), (10) \\ \quad V^* \left(\dfrac{\mathbf{Y}_i^M + \mathbf{Y}_i^{M*}}{2} \right) V = p_i - Re(\mathbf{S}_i^c), \quad i \in \mathcal{P} \\ \quad V_i \in \mathbb{C}, \quad i \in \mathcal{N} \\ \quad p_i \in \mathbb{R}, \quad i \in \mathcal{P}. \end{cases} \tag{27}$$

0.2.5.3 OPF avec Unit Commitment : $(OPF - UC)$

La dernière variante introduit des variables binaires dans la précédente. Pour chaque nœud producteur $i \in \mathcal{P}$, on va ajouter une variable $b_i \in \{0, 1\}$, modélisant le fait que le producteur du nœud i est soit allumé ($b_i = 1$) ou éteint ($b_i = 0$). Ainsi un nouveau degré de liberté est donné à l'optimiseur, celui d'allumer ou non un générateur. Cela a un impact quand des générateurs ont une production active minimale strictement positive. On appelle ce problème, le problème OPF avec *unit commitment*. La

modélisation mathématique que l'on étudie est la suivante :

$$
(OPF-UC)\begin{cases}
\min\limits_{V,p,b} \quad f_{obj}(p) \\
\text{s.t.} \\
\quad (24),(27),(10) \\
\quad \underline{\mathbf{p}_i} b_i \leq p_i \leq \overline{\mathbf{p}_i} b_i, \quad i \in \mathcal{P} & (28) \\
\quad \underline{\mathbf{q}_i} b_i - Im(\mathbf{S}_i^c) \leq V^* \left(\dfrac{\mathbf{Y}_i^M - \mathbf{Y}_i^{M*}}{2\mathbf{i}} \right) V \\
\quad \leq \overline{\mathbf{q}_i} b_i - Im(\mathbf{S}_i^c), \quad i \in \mathcal{P} & (29) \\
\quad b_i \in \{0,1\}, \quad i \in \mathcal{P} \\
\quad V_i \in \mathbb{C}, \quad i \in \mathcal{N} \\
\quad p_i \in \mathbb{R}, \quad i \in \mathcal{P}.
\end{cases}
$$

0.2.6 Instances étudiées

Dans cette section nous introduisons les différentes instances d'OPF considérées dans ce manuscrit. Il existe une communauté importante d'équipes de recherche étudiant principalement des sujets relatifs au problème OPF. Il existe aussi des outils logiciels spécifiques à l'étude de l'OPF. C'est le cas de Matpower [3], qui est un modeleur de problèmes OPF utilisable avec MatLab. Les instances d'OPF étudiées le plus souvent par la communauté sont écrites dans un format lu par Matpower qui est le plus utilisé en pratique. Ces instances décrivent les données du problème, et particulièrement les coûts linéaires et quadratiques en puissance active. En mettant les coûts quadratiques à zéro, on peut construire un modèle $(OPF-L)$ pour cette instance. Si on laisse l'instance telle quelle, on peut construire un modèle $(OPF-Q)$ pour cette instance. Enfin, en utilisant une variable de décision en chaque nœud de production, on obtient le modèle $(OPF-UC)$.

Les instances étudiées dans ce manuscrit sont celles présentes dans le modeleur Matpower. En plus de ces dernières on trouve des instances décrites par Bukhsh *et al.* dans [2].. On étudie également les instances de la bibliothèque PGLIB [4], créées à partir de celles de Matpower. On peut se référer à la Table **??** pour obtenir des informations sur les caractéristiques propres à chacune de ces instances.

0.3 Objectifs visés

Motivations

L'OPF est étudié chez RTE notamment dans des outils d'étude pour simuler un fonctionnement optimal du système électrique, « comme si » le système production-réseau était géré par un acteur unique. RTE utilise également des variantes d'OPF pour préparer des données d'étude, par exemple pour définir des plans de tension pour le développement de réseau à 10 ans, ou encore des OPF avec variables entières pour calculer des états de réseau dans une chaine de simulation d'aléas sur la production renouvelable (projet iTesla [5]). D'autres OPF avec variables discrètes cherchent à optimiser la topologie des réseaux électriques [6].

Dans certains cas on peut se contenter d'obtenir des optimums locaux, c'est-à-dire sans garantie d'optimalité globale. Mais pour certains travaux, il est nécessaire d'obtenir des garanties d'optimalité pour les OPF. Ce sera par exemple le cas pour intégrer des OPF dans des méthodes bi-niveau comme celle de Mitsos [7] ; un projet de recherche est actuellement en cours à RTE avec l'université d'Aix-la-Chapelle. Un problème bi-niveau est un problème d'optimisation dont l'une des données est le résultat d'un autre problème d'optimisation. La résolution du problème secondaire doit en général être exacte afin de connaitre avec certitude la donnée intervenant dans le premier problème. Certaines applications de RTE sont des problèmes bi-niveaux où l'OPF joue le rôle du problème secondaire, on peut citer par exemple le cas des études de sécurité.

Objectifs généraux

L'objectif de ce manuscrit est double : d'une part continuer à progresser dans la résolution locale des OPF, et d'autre part de les résoudre de façon globale. La principale contribution de cette thèse est le développement de l'algorithme `RC-OPF` qui résout de façon exacte les problèmes $(OPF - L)$ et $(OPF - Q)$. Cet algorithme s'appuie sur une reformulation quadratique convexe qui est construite à partir de la solution duale d'une relaxation SDP. Dans notre approche, on cherche notamment à profiter de la structure particulière du problème OPF, ainsi que des travaux antérieurs, notamment la relaxation semi-définie positive bien et connue et efficace, appelée relaxation du rang. Notre algorithme de résolution est comparé à la méthode dite MIQCR pour *Mixed-Integer Quadratic Convex Reformulation*

initialement développée pour résoudre des problèmes quadratiques en variables mixtes à l'optimum global. Ensuite, cette adaptation pourra s'étendre au problème $(OPF - UC)$ qui contient des variables entières.

Structure du manuscrit

Le premier chapitre de cette thèse fait office d'introduction. Avant toute chose, il contient une liste de notations utilisées tout au long de ce manuscrit, puis un aide mémoire auquel le lecteur pourra se référer afin d'obtenir le sens physique de chacune des variables et données présentes dans les modèles d'optimisation. Ensuite ce chapitre introduit le problème OPF et le contextualise dans la gestion d'un réseau de transport d'électricité. La modélisation du problème OPF est donnée par un programme quadratique en variables complexes. On prècise quelles sont les variables, données et contraintes du problème. Les trois variantes de l'OPF étudiées dans ce manuscrit sont expliquées dans ce premier chapitre : la variante $(OPF - L)$ qui modélise les coûts comme une fonction linéaire, la variante $(OPF - Q)$ qui préfère une modélisation quadratique et enfin la variante $(OPF - UC)$ qui introduit des variables binaires. On trouve aussi le détail des instances étudiées par la suite dans les expérimentations numériques. La fin du chapitre présente les objectifs de cette thèse.

Le deuxième chapitre décrit l'état de l'art. Il commence par celui spécifique à l'OPF et introduit son caractère NP-difficile. Puis nous évoquons le calcul de solutions réalisables pour l'OPF, c'est-à-dire l'obtention de bornes supérieures sur la valeur optimale du problème, ainsi que le calcul de bornes inférieures à l'aide de relaxations convexes. La deuxième section de ce chapitre traite de la résolution de problèmes quadratiques non-convexes à l'optimum global. Nous introduisons la hiérarchie de Lasserre puis les algorithmes de *branch-and-bound*. Enfin, nous concluons ce chapitre avec la présentation de la méthode de résolution exacte MIQCR, destinée à des QCQP non-convexes, à laquelle nous comparerons notre méthode `RC-OPF` dans la suite de ce manuscrit.

Le troisième chapitre commence avant tout par expliquer comment reformuler un problème OPF en un programme quadratique à variables complexes. Puis, ce chapitre présente notre contribution principale qui est la méthode de résolution `RC-OPF`. Cette méthode se décompose en deux phases, qui sont décrites dans ce chapitre. Pour expliquer la Phase 1 : nous commençons par introduire une

famille de reformulations quadratiques exactes de $(OPF - L)$. Puis la relaxation SDP dite du rang est rappelée. Ensuite, nous expliquons comment contruire une reformulation optimale au sein de notre famille de reformulations. Le théorème permettant de caractériser une telle reformulation optimale est donné. Enfin nous montrons que cette méthode s'étend naturellement aux variantes d'$(OPF - L)$ disposant de limites sur les flux d'intensité traversant les lignes du réseau électrique étudié. Dans la suite nous expliquons que notre reformulation est optimale même en considérant l'ajout des inégalités de McCormick dans celle-ci. Ceci nous permet d'obtenir une reformulation ayant le même niveau d'efficacité que celle de la méthode de reformulation MIQCR mais en résolvant un problème SDP plus petit, car contenant beaucoup moins de contraintes. Dans un second temps, ce troisième chapitre présente la Phase 2 de la méthode `RC-OPF`. Celle-ci consiste en un algorithme de *branch-and-bound* résolvant à l'exact la reformulation exacte optimale obtenue en Phase 1. Nous expliquons les détails de l'algorithme, et notamment comment choisir une variable et une valeur lors de l'étape de branchement. Puis nous indiquons comment choisir le nœud de l'arbre courant à traiter, nous donnons quelques autres paramètres génériques de résolution, et enfin un schéma de l'algorithme `RC-OPF`. Nous conclons sur comment résoudre $(OPF - L)$ à l'aide de `RC-OPF`. La troisième section illustre la méthode `RC-OPF` avec la résolution exacte de l'instance WB2. Nous commençons par présenter l'instance, puis par parcourir les différentes étapes de résolution : la substitution des variables, l'écriture du problème en variables complexes, le passage en variables réelles, la résolution de la relaxation du rang, la construction de la reformulation quadratique optimale, la détermination du nœud électrique de référence, le lancement de l'heuristique de recherche de bornes supérieures, et enfin la résolution avec l'arbre de *branch-and-bound*. La quatrième section de ce troisième chapitre traite de l'extension de `RC-OPF` à la résolution de la variante $(OPF - Q)$. Nous commençons par donner la famille de reformulations quadratiques que nous considérons pour cette variante. Puis nous introduisons la relaxation du rang dans le cas de $(OPF - Q)$. Enfin, nous présentons et démontrons le théorème caractérisant la reformulation optimale pour $(OPF - Q)$.

Le quatrième chapitre concerne les développements autour des contractions de bornes sur les variables implémentées dans l'algorithme `RC-OPF`. En effet les capacités de résolution de notre algorithme profitent grandement de contractions sur les bornes des variables. La première section s'attarde sur les contractions OBBT (*Optimality-Based Bound Tightening*). Elle présente d'abord comment ces

contractions sont utilisées dans `RC-OPF` puis explique comment elles synergisent avec notre reformulation optimale. On donne également les contractions RCBT (*Reduced-Costs Bound Tightening*) utilisées dans `RC-OPF`. Notre contribution réside ici dans l'utilisation conjointe de ces contractions avec la reformulation optimale, cette dernière renforce leur efficacité. Cette première section se conclue sur l'algorithme implémenté dans notre méthode `RC-OPF` et utilisant les techniques OBBT. La deuxième section est dédiée aux contractions FBBT (*Feasibility-Based Bound Tightening*) présentes dans `RC-OPF`. On explique dans un premier temps les contractions linéaires, puis les contractions non-linéaires. Ces dernières sont une autre contribution de cette thèse. Puis on donne l'algorithme mettant en oeuvre ces contractions dans `RC-OPF`. La troisième section explique comment utiliser ces techniques dans la résolution de la variante $(OPF - Q)$.

Dans le cinquième chapitre nous nous attachons à montrer les liens entre notre approche et celle de la hiérarchie de Lasserre. Nous commençons par introduire notre problèmatique et nos objectifs qui sont d'étendre l'approche de `RC-OPF` aux ordres supérieurs de la hiérarchie de Lasserre. Nous rappelons la hiérarchie de Lasserre, les formes prises par le primal et le dual de l'ordre d de cette hiérarchie. Ensuite, nous présentons notre contribution, à savoir une reformulation optimale d'un problème polynômial en utilisant un ordre d quelconque de cette hiérarchie. Tout d'abord nous présentons la famille de reformulations considérées, puis nous en décrivons la relaxation lagrangienne et nous donnons des conditions pour une absence de saut de dualité pour cette dernière. Nous établissons alors l'équivalence primale-duale. Enfin, nous discutons notre principal résultat, à savoir la reformulation optimale et le théorème afférent. Nous donnons la forme de la reformulation obtenue, et celle d'une variante possible.

Enfin dans le sixième chapitre, on trouve différentes expérimentations numériques réalisées pour nos variantes d'OPF, ainsi que la description des instances étudiées. Les premières d'entre elles comparent `RC-OPF` au logiciel BARON, et montrent que notre méthode permet de résoudre des instances où BARON est inopérant. Les secondes démontrent l'impact positif des contractions de bornes développées et implémentées dans `RC-OPF`. Nous étudions l'impact du paramètre δ intervenant dans l'étape de branchement de notre algorithme de résolution `RC-OPF`. Les expérimentations suivantes donnent les résultats obtenus lorsque l'on utilise `RC-OPF` sur les instances d'OPF connues de l'état de l'art. On traite d'abord le cas de la variante $(OPF - L)$, puis celui de la variante $(OPF - Q)$. On donne également une mise en perspective des résultats obtenus avec ceux de l'état de l'art sur la résolution exacte de l'OPF.

Ensuite ce chapitre se poursuit avec la recherche d'instances plus difficiles à résoudre de manière exacte. On donne alors les résultats obtenus par `RC-OPF` sur ces instances. Enfin, on exploite ces instances plus difficiles pour comparer les trois différentes méthodes de branchement envisagées dans ce manuscrit.

Chapitre 1

État de l'art

Contenu

1.1 État de l'art sur l'OPF 54

1.1.1 Complexité de l'OPF 54

1.1.2 Calculs de solutions réalisables pour l'OPF : bornes supérieures 56

1.1.3 Relaxations de QCQP applicables à l'OPF : bornes inférieures 56

1.2 État de l'art sur la résolution exacte des QCQP 61

1.2.1 La hiérarchie de Lasserre pour résoudre des QCQP 63

1.2.2 Méthodes par algorithmes de *branch-and-bound* pour résoudre un QCQP . . . 65

1.2.3 La méthode MIQCR 71

1.1 État de l'art sur l'OPF

On s'intéresse à l'OPF depuis 1962 [8] et ce problème est encore aujourd'hui difficilement résolu de manière exacte, même si de nouvelles approches ont émergé dans la dernière décennie, notamment autour des relaxations semi-définies positives [9, 10, 11, 12, 13]. Cette section décrit l'état des connaissances sur les propriétés de l'OPF, puis sur sa résolution de manière exacte. L'OPF étant un problème d'optimisation, on procède souvent par comparaison de bornes inférieures et supérieures pour le résoudre. Les pages suivantes présentent les techniques actuelles pour obtenir de telles bornes.

Comme on l'a vu dans le chapitre précédent, l'OPF s'exprime naturellement comme un problème d'optimisation avec des variables complexes. Cependant, la quasi intégralité des solutions logicielles de résolution de problèmes d'optimisation travaillent sur des problèmes en variables réelles. On trouve la formulation en variables réelles de l'OPF dans de multiples publications, citons, par exemple, [14] où le modèle obtenu est un QCQP (*Quadratically-Constrained Quadratic Program*). Les formulations complexes envisagées dans la littérature sont souvent identiques et correspondent à ce qui est décrit dans le chapitre précédent, en particulier les variables d'intensité sont remplacées par celles de tension. Les formulations réelles sont souvent différentes, on en distingue deux grandes familles selon que l'on choisisse les coordonnées polaires ou cartésiennes pour passer des variables complexes aux variables réelles. Citons aussi cette revue de littérature [15] en complément des informations détaillées sur l'OPF présentes dans les sections suivantes.

1.1.1 Complexité de l'OPF

Dans cette section, on présente les principales propriétés connues pour le problème OPF. On donne des références démontrant le caractère NP-difficile de l'OPF. On rappelle également que les modélisations de l'OPF sont non-convexes et on explicite les variantes particulières dont les modélisations sont polynomiales.

Les contraintes de l'OPF, et particulièrement celles définissant les variables de puissance comme le produit entre des variables de tension et d'intensité, induisent des contraintes non-convexes dans l'OPF. Ceci est connu depuis que le problème a été posé, mais est mis plus en avant dans l'étude menée par Bukhsh [2]. Dans cette étude il démontre l'existence de plusieurs points-selles dans des instances

d'OPF modélisées comme des problèmes d'optimisation en variables réelles. L'existence de plusieurs points-selles confirme le caractère non-convexe des formulations de l'OPF.

Le fait que l'OPF ne soit modélisé que comme un problème d'optimisation non-convexe le classe dans la catégorie des problèmes difficiles à résoudre de manière exacte. Dans [16], les auteurs présentent une preuve du caractère NP-difficile de l'OPF. Pour ce faire ils ne s'intéressent qu'au problème de décision consistant à déterminer si une version simplifiée de l'OPF est réalisable ou non. Les caractéristiques de la formulation utilisée sont :

— Absence de résistance sur les lignes, mais présence d'une réactance, c'est-à-dire que les entrées des matrices $\mathbf{Y}$ sont des imaginaires purs.
— Injections fixées : un seul noeud soutire de la puissance et un seul noeud en produit,
— Tension aux noeuds fixée à la tension de référence,
— Différences angulaires maximales entre les nœuds adjacents, mais pas de limite de flux.

Les auteurs montrent qu'il existe une transformation, polynomiale en la taille des données, qui réduit le problème NP-difficile *one-in-three 3-SAT* en cette formulation simplifiée. Ainsi la résolution de l'OPF dans un réseau présentant les caractéristiques ci-dessus est NP-difficile.

On connait un certain nombre de conditions qui, si elles sont vérifiées par le réseau étudié, permettent une résolution de l'OPF de manière polynomiale en la taille des données. On les liste ci-dessous :

— Réseaux arborescents [17] ($\mathcal{G}$ est un arbre),
— Possibilité de fournir plus que la puissance demandée en un noeud [18], c'est-à-dire que la contrainte de bilan de puissance devient une inégalité au lieu d'être une égalité, il faut fournir **au moins** la puissance consommée en chaque nœud.
— Pas de bornes inférieures sur la puissance générée [10],
— Réseaux sans pertes sur graphes cycliques [19], c'est-à-dire que l'on considère des simplifications des équations physiques du réseau, au lieu de considérer celles explicitées dans le chapitre sur la modélisation de l'OPF.

Malheureusement, pour chacune des trois variantes de notre cas d'étude, aucune des conditions listées ci-dessus n'est vérifiée. On ne se trouve donc pas dans un cas simple où un algorithme polynomial est connu pour résoudre l'OPF.

1.1.2 Calculs de solutions réalisables pour l'OPF : bornes supérieures

Cette section présente les différentes méthodes de l'état de l'art pour trouver des solutions réalisables pour l'OPF, et notamment les méthodes par points intérieurs. L'algorithme des points intérieurs est construit pour résoudre des problèmes d'optimisation convexe génériques de manière exacte. Pour ce faire, il cherche à trouver un point satisfaisant les conditions d'optimalité du premier ordre avec une gestion particulière des contraintes de complémentarité. On peut implémenter à l'aide de diverses méthodes cet algorithme. On peut également fournir un problème non-convexe à un algorithme de points intérieurs. Ce dernier renverra également un point satisfaisant les conditions d'optimalité du premier ordre. Cela ne permet pas de conclure quant au caractère d'optimum du point donné. Cependant, satisfaire les conditions du premier ordre implique que la solution trouvée est réalisable. Une fois qu'une formulation quadratique réelle de l'OPF est construite, des solveurs comme KNITRO [20] ou IPOPT [21] permettent de calculer une solution réalisable. Cet algorithme est utilisé avec succès sur des instances d'OPF représentant des réseaux européens de très grande taille (plus de 1000 nœuds dans le graphe $\mathcal{G}$) dans [22] à l'aide de la plateforme Matpower [3], présentée en Section 0.2.6.

D'autres heuristiques, plus spécifiques au problème OPF, peuvent être utilisées pour trouver des solutions réalisables. On trouve notamment des méthodes approximant les équations du réseau pour approcher pas à pas des solutions réalisables, et ce de manière linéaire ou quadratique. On trouve également des algorithmes génétiques, des algorithmes évolutifs ou des algorithmes à particules. La revue de littérature [15] fournit quelques exemples de telles heuristiques. De telles méthodes ne sont pas utilisées dans les expérimentations détaillées dans ce manuscrit. Pour obtenir des solutions réalisables, l'algorithme des points intérieurs est toujours préféré dans nos expérimentations, car il donne de manière satisfaisante et robuste de bonnes solutions réalisables pour les instances connues de la communauté de l'OPF.

1.1.3 Relaxations de QCQP applicables à l'OPF : bornes inférieures

La recherche de bonnes bornes inférieures pour les problèmes OPF est une problématique très présente dans la littérature. De nombreuses relaxations ont été imaginées pour y répondre. Celles-ci sont en général convexes afin d'assurer une résolution rapide à leur optimum global. On trouve des relaxations linéaires, quadratiques convexes, SOCP (*Second-Order Cone Programming*) et SDP

(*Semidefinite Programming*). Les principales références relatives à ces relaxations sont données dans cette section.

Comme vu au début de la Section 1.1, l'OPF s'exprime naturellement comme un QCQP en variables réelles. On donne ci-dessous sa forme standard :

$$(QCQP)\begin{cases} \min\limits_{v} \quad v^t A v + c^t v \\ \text{s.t.} \\ \quad v^t B_k v + d_k^t v \leq e_k, \quad k \in K \\ \quad v \in \mathbb{R}^n. \end{cases} \tag{1.1}$$

n est un entier strictement positif. v est le vecteur des variables, A et $(B_k)_{k\in K}$ sont des matrices carrées symétriques aux dimensions compatibles avec v. c et $(d_k)_{k\in K}$ sont des vecteurs de même dimension que v. Les $(e_k)_{k\in K}$ sont des réels. Si A est semi-définie positive, l'objectif est une fonction convexe. De même si B_k est semi-définie positive, la contrainte (1.1) définit un domaine convexe. La fonction objectif de l'OPF est convexe car les coefficients des coûts de production sont positifs.

Relaxations linéaires

La linéarisation d'un QCQP est une méthode bien connue dans la littérature. Prenons un problème d'optimisation dont les variables sont notées $(v_i)_{i\in I}$. Pour chaque produit de variables $v_i v_j$, on introduit une variable $W_{i,j}$. En remplaçant les produits de variables v par les variables W, on linéarise le problème. Cette linéarisation est une relaxation et elle peut être renforcée en ajoutant des inégalités valides linéaires dans le problème. C'est ce que proposent les auteurs de [23] qui linéarisent les équations de réseau de l'OPF afin d'obtenir une relaxation linéaire de l'OPF.

Un renforcement classique pour un QCQP consiste à ajouter les inégalités de McCormick [24]. L'enveloppe convexe de l'espace défini par les contraintes suivantes $W_{i,j} = v_i v_j, \quad \ell_i \leq v_i \leq u_i, \quad \ell_j \leq v_j \leq u_j$ est définie par les inégalités de McCormick :

$$(MCC)\begin{cases} W_{i,j} \leq u_j v_i + \ell_i v_j - \ell_i u_j, \quad \forall (i,j) \in \{1,\dots,n\}^2 \\ W_{i,j} \leq \ell_j v_i + u_i v_j - u_i \ell_j, \quad \forall (i,j) \in \{1,\dots,n\}^2 \\ W_{i,j} \geq u_j v_i + u_i v_j - u_i u_j, \quad \forall (i,j) \in \{1,\dots,n\}^2 \\ W_{i,j} \geq \ell_j v_i + \ell_i v_j - \ell_i \ell_j, \quad \forall (i,j) \in \{1,\dots,n\}^2 \end{cases}$$

En ce qui concerne le problème OPF, on est capable de construire des bornes inférieures ou supérieures sur toutes les variables représentant des quantités physiques. On peut donc utiliser ces inégalités.

Voici la forme que prend la relaxation linéaire de $(QCQP)$ renforcée avec les inégalités de McCormick :

$$(QCQP-LP)\begin{cases} \min\limits_{v,W} \quad < A, W > + c^t v \\ \text{s.t.} \\ \quad < B_k, W > + d_k^t v \leq e_k, \quad k \in K & (1.2) \\ \quad (v, W) \in (MCC), & (1.3) \\ \quad v \in \mathbb{R}^n, \\ \quad W \in \mathcal{M}_n(\mathbb{R}). \end{cases}$$

Relaxations quadratiques convexes

En ne conservant que certaines contraintes convexes, et en procédant à la linéarisation des expressions quadratiques, on peut obtenir des relaxations quadratiques convexes de meilleure qualité qu'une simple linéarisation. On peut toujours renforcer ces relaxations avec des inégalités linéaires ou quadratiques convexes valides. Cette approche est menée dans [25] et fournit des relaxations très précises sur les cas tests emblématiques de l'OPF, qui se résolvent rapidement, pour des instances allant de quelques dizaines à quelques milliers de nœuds.

Relaxations SOCP

Des relaxations SOCP ont aussi été introduites pour l'OPF, notamment dans [26] et dans [27]. En partant de notre QCQP sous-forme standard, on peut introduire une matrice de variables W vérifiant $W = vv^t$. Dans le cas d'une linéarisation, on relaxe complètement la contrainte $W = vv^t$. Dans le cas SOCP, on peut conserver les contraintes de la forme $W_{i,j}^2 \leq W_{i,i}W_{j,j}$, qui peuvent s'exprimer comme des contraintes d'appartenance à un cône du second ordre après rotation du repère. Ceci constitue la relaxation SOCP standard, qui est très rapide à résoudre mais ne fournit pas nécessairement des bornes inférieures de qualité. Néanmoins cette relaxation peut être renforcée avec des contraintes valides, linéaires ou SOCP. Avec ces renforcements , les relaxations SOCP fournissent rapidement de bonnes bornes inférieures pour l'OPF ; même pour les plus grandes instances, représentant des réseaux de plusieurs milliers de nœuds [28, 27].

Relaxations SDP

Les relaxations SDP ont d'abord été utilisées avec beaucoup de succès sur des premières instances d'OPF dans [9] et [10]. Pour ces instances, la relaxation SDP appelée relaxation du rang fournit une borne inférieure exacte. C'est le cas de la plupart des instances de la bibliothèque Matpower. Depuis, de nouvelles instances ont été générées où cette relaxation est inexacte, mais c'est souvent celle donnant la meilleure borne inférieure, cependant c'est aussi la plus lente à calculer. C'est le cas des instances de la bibliothèque PGLib [4].

Cette relaxation repose sur le fait que la contrainte $W = vv^t$ est équivalente à :

$$W \succeq 0, \quad rg(W) = 1.$$

En relaxant la contrainte de rang égal à un [1], on obtient une relaxation semi-définie positive de l'OPF. Plus précisèment, la relaxation SDP de $(QCQP)$ sous forme standard s'écrit :

$$(QCQP - SDP) \begin{cases} \min\limits_{v,W} \quad < A, W > + c^t v \\ \text{s.t.} \\ \quad < B_k, W > + d_k^t v \leq e_k, \quad k \in K & (1.4) \\ \quad \begin{bmatrix} 1 & v^t \\ v & W \end{bmatrix} \succeq 0 & (1.5) \end{cases}$$

On remarquera que la relaxation SOCP est plus faible que la relaxation SDP car les contraintes $W_{i,j}^2 \leq W_{i,i} W_{j,j}$ sont incluses dans la contrainte SDP.

Depuis la première introduction de ces relaxations pour trouver des bornes inférieures à des OPF, d'autres études ont été menées. On peut citer [29] qui présente une relaxation SDP renforcée pour l'OPF. D'autres études se concentrent sur une accélération de la résolution des relaxations SDP en profitant de la structure très creuse du problème. Elles exploitent une décomposition en cliques de l'extension cordale du graphe du réseau [30, 22]. Pour un problème sous la forme $(QCQP - SDP)$, on peut définir sa matrice de parcimonie P de même taille que la matrice A. Une entrée de coordonnée (i, j) de P est non-nulle si et seulement s'il existe une entrée non nulle de coordonnées (i, j) dans A ou dans l'une des matrices B_k. Si la matrice P a un nombre d'entrées nulles relativement très important par rapport à son nombre d'entrées non-nulles, elle est dite creuse. Un problème $(QCQP - SDP)$ ayant une structure creuse correspond au cas où la matrice P associée à ce problème est creuse.

1. d'où le nom de relaxation du rang

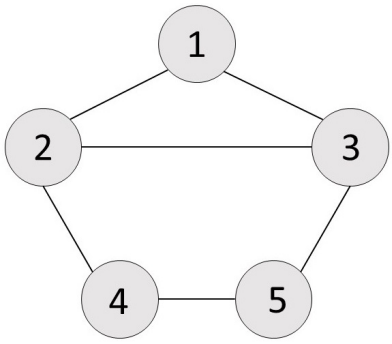

FIGURE 1.1 – Graphe à 5 nœuds

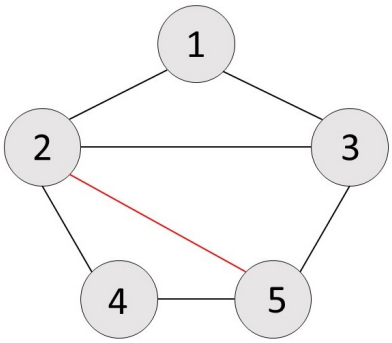

FIGURE 1.2 – Extension cordale

Résolution efficace de relaxations SDP

Dans [31, 32], les auteurs proposent une méthode de décomposition permettant de résoudre plus rapidement un problème SDP creux. L'idée est de décomposer la contrainte SDP en plusieurs contraintes SDP sur des matrices de plus petite dimension. Pour ce faire, il faut construire le graphe de parcimonie du problème. Ce graphe associe un nœud à chaque entrée i de la matrice SDP W. Une arête relie les nœuds i et j s'il existe un coefficient non nul associé à la variable $W_{i,j}$ dans l'objectif ou dans une contrainte du problème. Une fois ce graphe construit, il faut en calculer une extension cordale, c'est-à-dire ajouter des arêtes au graphe pour que chaque cycle de taille supérieur à 4 contienne au moins une corde [33]. Finalement, on peut décomposer la contrainte SDP sur chaque clique maximale de l'extension cordale. Un exemple illustratif est proposé ci-dessous.

La Figure 1.1 représente un graphe à 5 nœuds. La Figure 1.2 donne une extension cordale de ce graphe. Ici on ajoute une arête (en rouge) entre les nœuds 2 et 5, de sorte que le cycle 2,3,5,4 comporte

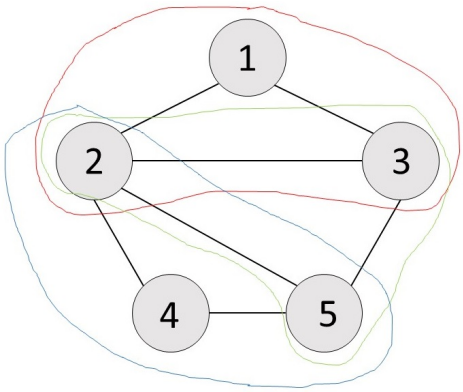

FIGURE 1.3 – Décomposition en cliques de l'extension cordale

une corde. La Figure 1.3 donne la décomposition en cliques maximales de l'extension cordale. Cette décomposition se résume en trois cliques de taille 3. La contrainte SDP se décompose alors, pour cet exemple, en trois contraintes SDP de taille 3, au lieu d'une de taille 5.

Cette méthode de résolution par décomposition permet d'obtenir beaucoup plus rapidement une bonne borne inférieure, mais des problèmes numériques remettent en question sa stabilité pour les plus grandes instances, c'est-à-dire celles pour lesquelles le réseau comporte plusieurs milliers de nœuds électriques [22].

1.2 État de l'art sur la résolution exacte des programmes quadratiques non-convexes applicable à l'OPF

L'OPF peut s'exprimer comme un QCQP en variables réelles à partir de sa formulation complexe présentée en introduction. Une manière d'obtenir cette formulation réelle est détaillée dans [14]. On montre aussi comment passer de la formulation complexe de l'introduction à une formulation réelle au début du chapitre 2. Dans ce chapitre, on s'intéresse aux techniques présentes dans l'état de l'art pour résoudre de manière exacte des problèmes QCQP non-convexes et applicables à l'OPF.

Les méthodes de résolution exacte d'un QCQP reposent sur le calcul de bornes supérieures et inférieures sur le problème. L'objectif est de faire converger ces deux bornes afin de pouvoir conclure sur l'optimalité de la solution trouvée. À ce propos, dans le cadre de ce manuscrit, l'optimalité considérée

est une optimalité à une certaine précision près, c'est-à-dire que si l'écart relatif entre une borne supérieure et inférieure du problème est inférieur à notre précision, le problème est considéré résolu. En général, le calcul d'une borne supérieure n'est pas un souci en ce qui concerne l'OPF (voir Section 1.1.2). La difficulté réside généralement dans l'obtention de bornes inférieures augmentant progressivement vers la valeur optimale du problème. Pour ce faire, deux familles de méthodes ont été principalement étudiées pour l'OPF : la hiérarchie de Lasserre et les algorithmes de *branch-and-bound*.

Pour présenter et comparer ces différentes techniques, on considère un problème QCQP type mis sous la forme standard suivante :

$$(QCQP)\begin{cases} \min\limits_{v} \quad v^tAv + c^tv \\ \text{s.t.} \\ \quad v^tB_kv + d_k^tv \leq e_k, \quad k \in K \\ \quad v \in \mathbb{R}^n. \end{cases} \tag{1.1}$$

De plus, on note $n \in \mathbb{N}^*$ la dimension de v.

La résolution des QCQP convexes de manière exacte est plus facile que celle des QCQP non-convexes. En effet, dans ce cas là, la satisfaction des conditions du premier ordre de KKT, suffit à garantir l'optimalité globale. On peut résoudre un QCQP convexe à l'aide d'un algorithme de points intérieurs par exemple. Les algorithmes privilégiés pour résoudre efficacement un QCQP convexe de manière exacte ne permettent pas de résoudre de manière exacte des QCQP non-convexes. Pour ces derniers, la satisfaction des conditions du premier ordre de KKT ne permet pas de conclure quant à l'optimalité globale. Il faut donc utiliser d'autres formes d'algorithmes.

Les techniques présentées ci-après pour résoudre un QCQP non-convexe de manière exacte sont : la hiérarchie de Lasserre et les méthodes par algorithme de *branch-and-bound*. Dans la première partie de cette section, on détaille la hiérarchie de Lasserre. Dans la seconde partie, on présente des méthodes de résolution utilisant des algorithmes de *branch-and-bound*. Dans cette seconde partie, on s'intéresse aussi aux techniques de contraction de bornes sur les variables. Dans la dernière partie, on présente plus spécifiquement la méthode de reformulation quadratique appelée MIQCR.

1.2.1 La hiérarchie de Lasserre pour résoudre des QCQP

La hiérarchie de Lasserre [34, 35, 36] est une hiérarchie de relaxations SDP d'un problème d'optimisation polynomial dont les valeurs tendent vers la valeur optimale du problème. À chaque relaxation est associée un ordre $d \in \mathbb{N}^*$ appelé ordre de la relaxation. Quand cet ordre tend vers l'infini, la relaxation tend à être exacte. Cependant, les problèmes grossissent de manière exponentielle, et il est déjà difficile de résoudre les ordres les plus petits. Le caractère creux du problème OPF rend ces résolutions possibles en construisant une hiérarchie plus faible avec des ordres partiels [37]. Cela revient à n'élever à un ordre supérieur que certaines contraintes du problème, relatives à certains nœuds du réseau électrique. Une présentation générique de cette méthode sur des problèmes en variables réelles et complexes se trouve dans [38]. Une adaptation de la hiérarchie de Lasserre au problème OPF a aussi été introduite dans [39]. Cependant cette méthode échoue à résoudre de manière exacte des instances d'OPF relatives à des réseaux électriques de plus de quelques dizaines de nœuds.

L'ordre 1 de cette hiérarchie correspond à la relaxation SDP $(QCQP - SDP)$ présentée dans la Section 1.1.3. Dans ce qui suit, on se limite à présenter la hiérarchie de Lasserre dans le cas où on l'applique à la résolution exacte d'un QCQP, puisque c'est ce qui nous intéresse dans ce manuscrit.

Les relaxations de la hiérarchie de Lasserre linéarisent les monômes présents dans les polynômes du problème. La relaxation d'ordre d dans la hiérarchie de Lasserre représente les monômes de degré $2d$ ou moins. En utilisant un multi-indice $\alpha \in \mathbb{N}^n$, on représente un monôme. L'indice α_i est l'exposant associé à la coordonnée v_i de ce monôme. Ainsi, par définition, $v^\alpha = \prod_{i=1}^n v_i^{\alpha_i}$. Pour un polynôme f de degré $2d$ ou moins, on note f_α son coefficient relatif au monôme α de f. On note Γ_n^d l'ensemble des monômes de degré d ou moins dans $\mathbb{R}^n$. Il s'ensuit que $f(v) = \sum_{\alpha \in \Gamma_n^{2d}} f_\alpha v^\alpha$. Avec ces notations, on peut réécrire $(QCQP)$:

$$(QCQP) \begin{cases} \min\limits_{v} \sum\limits_{\alpha \in \Gamma_n^{2d}} g_{0,\alpha} v^\alpha & (1.6) \\ \text{s.t.} \\ \sum\limits_{\alpha \in \Gamma_n^{2d}} g_{k,\alpha} v^\alpha + h_k \geq 0, \quad k \in K & (1.7) \end{cases}$$

Avec les coefficients $g_{0,\alpha}, g_{k,\alpha}$ et h_k définis à partir de A, c, B_k, d_k et e_k.

Soit d un entier, introduisons la relaxation d'ordre d de $(QCQP)$. Pour la construire, on associe à chaque monôme α de degré $2d$ ou moins une variable y_α prenant la place de v^α. On note $|d, n|$ le nombre de monômes de degré d ou moins sur $\mathbb{R}^n$. On linéarise l'objectif et les contraintes en remplaçant les monômes v^α par y_α. Par exemple pour $n = 2$ les monômes de degré 1 sont : v_1, v_2. Ceux de degré 2 sont : v_1^2, v_2^2, v_1v_2. Ainsi $|2, 2| = 5$.

Soit $y_{d'}$ le vecteur contenant les variables y_α pour α représentant des monômes de degré d' ou moins. On remarque que $y_{d'}$ est un vecteur de $\mathbb{R}^{|d',n|}$.
Par exemple, pour $n = 2$ et $d' = 1$, $y_{d'} = \begin{pmatrix} y_{10} \\ y_{01} \end{pmatrix}$.
Les coefficients de la matrice $\begin{pmatrix} 1 \\ y_{d'} \end{pmatrix} \begin{pmatrix} 1 \\ y_{d'} \end{pmatrix}^t$ sont soit 1, soit une variable y_α, soit un produit de deux variables y_α.
Reprenons notre exemple : $\begin{pmatrix} 1 \\ y_{d'} \end{pmatrix} \begin{pmatrix} 1 \\ y_{d'} \end{pmatrix}^t = \begin{pmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{10}y_{10} & y_{10}y_{01} \\ y_{01} & y_{10}y_{01} & y_{01}y_{01} \end{pmatrix}$.
On remplace dans cette matrice chaque produit de deux variables $y_{\alpha,1} \times y_{\alpha,2}$, par la variable y_α représentant le monôme obtenu par le produit des deux monômes représentés par $y_{\alpha,1}$ et $y_{\alpha,2}$ On appelle cette matrice la matrice des moments y d'ordre d'. On la note $\mathcal{M}^n_{d'}(y)$.
Par exemple $\mathcal{M}^2_1(y) = \begin{pmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix}$.

Maintenant que l'on a introduit les variables de linéarisation y_α et les matrices de moment en y, nous construisons la relaxation de Lasserre d'ordre d.

1. L'objectif : La fonction objectif s'obtient en linéarisant les monômes v^α en y_α.

2. Les contraintes : Après avoir linéarisé chaque contrainte, on transforme (éventuellement) cette dernière en une contrainte SDP. Soit la contrainte k. On fixe d'_k à $d - 1$ si B_k est non-nul. Sinon, on fixe d'_k à d. Cela correspond au cas où B_k est nulle, et donc où la contrainte k est linéaire. Si d'_k vaut 0, on linéarise simplement la contrainte. Sinon la contrainte est remplacée par une contrainte SDP. La matrice de cette contrainte s'obtient à partir du produit entre le scalaire $\left(\sum_{\alpha \in \Gamma^{2d}_n} g_{k,\alpha} y_\alpha + h_k\right)$ et la matrice $\mathcal{M}^n_{d'_k}(y)$. Ce produit est une matrice carrée de même taille que $\mathcal{M}^n_{d'_k}(y)$ (i.e. $|d'_k, n| + 1$), dont les coefficients sont des sommes de produits de variables y_α. En réinterprétant ces produits comme des produits de monômes, on remplace chaque produit par

la variable y_α représentant le monôme produit. Par définition de d'_k, on est sûr que le monôme obtenu est de degré $2d$ au maximum. Notre contrainte devient alors la semi-définie positivité de la matrice obtenue par remplacement. Cette matrice peut se voir comme une combinaison linéaire de matrices carrées de taille $|d'_k, n| + 1$. Les coefficients de cette combinaison linéaire étant les variables y_α. On notera $B^d_{k,\alpha}$ la matrice relative à la variable y_α dans cette combinaison linéaire.

3. On ajoute une contrainte supplémentaire : $\mathcal{M}^n_d(y) \succeq 0$. Cette matrice peut aussi s'écrire comme combinaison linéaire de matrices que l'on notera $B^d_{0,\alpha}$.

La relaxation de Lasserre d'ordre d de $(QCQP)$ est alors :

$$(QCQP)_d \begin{cases} \min\limits_y \sum\limits_{\alpha \in \Gamma^{2d}_n} g_{0,\alpha} y_\alpha & (1.8) \\ \text{s.t.} \\ \sum\limits_{\alpha \in \Gamma^{2d}_n} y_\alpha B^d_{k,\alpha} + h_k B^d_{k,0} \succeq 0, \quad k \in K & (1.9) \\ \sum\limits_{\alpha \in \Gamma^{2d}_n} y_\alpha B^d_{0,\alpha} + B^d_{0,0} \succeq 0 & (1.10) \end{cases}$$

On notera que les contraintes linéarisées peuvent aussi se mettre sous la forme de combinaisons linéaires en considérant des matrices $B_{k,\alpha}$ de taille 1. De plus les contraintes d'égalités sont transformées en contraintes d'égalités matricielles et non en contraintes SDP, car de même que le seul réel a vérifiant $0 \leq a \leq 0$ est zéro, la seule matrice A vérifiant $A \succeq 0$ et $0 \succeq A$ est la matrice nulle.

La taille des relaxations de la hiérarchie de Lasserre augmente de manière exponentielle (regarder l'évolution de $|d, n|$). Dans le cadre de la résolution de l'OPF, on peut utiliser avantageusement le caractère creux du problème. Cependant, même en usant de ce caractère, il est numériquement trop long de résoudre ne serait-ce que l'ordre 2 de la hiérarchie de Lasserre pour des réseaux dépassant plusieurs dizaines de nœuds électriques. Ceci limite la portée de cette approche.

1.2.2 Méthodes par algorithmes de *branch-and-bound* pour résoudre un QCQP

Les algorithmes de *branch-and-bound* spatiaux se proposent de résoudre un problème générique d'optimisation non-convexe en variables réelles et bornées. L'objectif est de découper progressivement l'espace admissible de chaque variable en suivant une structure arborescente. Chaque nœud de cet arbre correspond à une portion de l'espace admissible du problème de départ. En chaque nœud, on

résout la relaxation obtenue en se limitant au problème de départ confiné dans la portion relative à ce nœud. On obtient alors une succession de bornes inférieures qui se précisent jusqu'à l'optimum. Lorsque l'on introduit des coupes admissibles aux relaxations au fur et à mesure que l'on évolue en profondeur dans l'arbre, on parle généralement d'algorithmes de *branch-and-cut*.

Dans cette section, on suppose qu'il existe des bornes ℓ et u sur les variables x de notre problème standard $(QCQP)$. Il existe de très nombreux algorithmes de *branch-and-bound* spatiaux pour résoudre des QCQP, on s'attache ici à en retranscrire les concepts-clés. De nombreux outils logiciels mettent en place ce genre de méthodes pour résoudre des problèmes d'optimisation non-convexes à l'optimum global. Citons ici BARON [40] et COUENNE [1]. L'idée générale est de résoudre une succession de relaxations convexes du problème de départ jusqu'à pouvoir conclure sur l'optimalité à une précision près. Ici, et contrairement à la hiérarchie de Lasserre, la taille des problèmes ne grandit pas forcément, et elle ne varie pas dans le cas le plus général. Un algorithme de *branch-and-bound* s'appuie normalement sur une relaxation. Le cas le plus simple à envisager serait une relaxation linéaire, mais on peut utiliser une relaxation quadratique convexe, SOCP ou SDP. On doit ensuite ajouter des contraintes valides pour $(QCQP)$ à la relaxation choisie afin d'assurer la convergence de l'algorithme de *branch-and-bound*. Le cas le plus courant pour un QCQP est d'utiliser les inégalités de McCormick [24]. Par exemple, en reprenant la forme standard $(QCQP)$, on peut proposer la relaxation linéaire $(QCQP - LP)$.

On a ainsi construit notre relaxation sur laquelle s'appuie l'algorithme de *branch-and-bound*. L'algorithme consiste ensuite à parcourir un arbre, et en chaque nœud de l'arbre, à en résoudre une relaxation. Le parcours de l'arbre commence à la racine de celui-ci, et l'arbre s'étend au fur et à mesure de son parcours. Chaque nœud n'est parcouru qu'une seule fois. Chaque nœud est caractérisé par un sous-espace du domaine $\ell \leq v \leq u$. La relaxation résolue en ce nœud utilise alors les bornes de ce sous-espace. En fonction du résultat de cette relaxation, on poursuit ainsi le parcours de l'arbre :

— Si la relaxation est infaisable, on quitte cette branche de l'arbre, et on repart d'un autre nœud pas encore exploré. Cela veut dire qu'il n'existe pas de solutions réalisables dans le sous-espace de ce nœud.

— Si la relaxation donne une valeur plus grande que celle d'une solution réalisable connue, on quitte cette branche et on repart d'un autre nœud pas encore exploré. Cela veut dire que dans ce sous-espace, il n'y a pas de meilleures solutions que la meilleure solution déjà connue, on

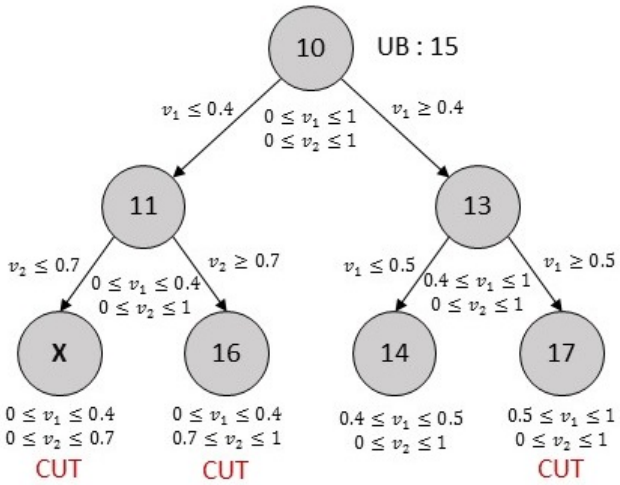

FIGURE 1.4 – Exemple d'un algorithme de *branch-and-bound* spatial

perd alors du temps à explorer cet espace.

— Si la relaxation donne une valeur plus basse que celle de la meilleure solution connue, on n'a pas encore assez d'informations pour conclure. On construit alors deux nœuds fils à partir de ce nœud. On le fait en coupant en deux le domaine d'une des variables. Une des parties revient à un fils, l'autre à l'autre fils. On repart ensuite d'un autre nœud pas encore exploré, cela peut être l'un des nœuds fils nouvellement créés.

Le parcours de cet arbre s'accompagne de recherches plus ou moins fréquentes de solutions réalisables, pour mettre à jour la meilleure borne supérieure connue sur le problème. Quand il n'y a plus de nœuds dans l'arbre à parcourir, l'algorithme se termine.

La Figure 1.4 présente un arbre de minimisation sur un problème contenant deux variables v_1 et v_2. Les nombres à l'intérieur des nœuds de l'arbre représentent la valeur de la relaxation calculée en les nœuds.

La convergence de ce type d'algorithme est assurée pour un QCQP car au fur et à mesure du parcours de l'arbre, la profondeur dans ce dernier augmente, et le volume correspondant à l'espace $\ell \leq v \leq u$ tend vers zéro. Or quand ce volume tend vers zéro, l'écart entre la borne inférieure donnée par la relaxation et la valeur optimale du problème restreint à l'espace $\ell \leq v \leq u$ tend vers zéro. Ainsi, une optimalité, à une certaine précision près, est atteignable en temps fini.

Dans le cas de l'OPF

Comme la relaxation SDP est la plus performante pour l'OPF, c'est elle qui a été choisie préférentiellement comme relaxation à résoudre aux nœuds d'un *branch-and-bound* spatial. On trouve un tel algorithme utilisé dans [12], ainsi que dans [41] avec une variante *branch-and-cut* spécialisée dans le traitement de problèmes en variables complexes. Un algorithme de *branch-and-cut* spatial utilisant des relaxations SOCP a aussi été développé pour résoudre l'OPF à l'optimum global [28]. Ces algorithmes de *branch-and-bound* donnent des résultats probants sur des instances de taille généralement réduite à quelques dizaines de nœuds dans le réseau électrique. Cependant, ces algorithmes échouent sur de plus grandes instances contenant plusieurs centaines de nœuds.

La réduction de bornes

Un point clé d'un algorithme de *branch-and-bound* réside dans la réduction de bornes. Ici on parle des bornes ℓ et u sur les variables. En effet si on est capable en chaque nœud de l'arbre de réduire les bornes sur un grand nombre de variables, on améliore la qualité de la relaxation et on diminue de fait le nombre de nœuds à parcourir avant de pouvoir conclure quant à l'optimalité. C'est ce qu'on appelle la réduction de bornes.

Les logiciels d'optimisation comme BARON ou COUENNE implémentent de nombreuses techniques de réduction de bornes. On en trouve plusieurs détaillées dans [40] et [1]. On se concentre ici sur les plus fondamentales qui seront réutilisées pour résoudre l'OPF. On distingue deux familles de réduction des bornes, comme c'est le cas dans [42], celles utilisant des arguments uniquement de faisabilité (FBBT, pour *Feasibility-Based Bound Tightening*) et d'autres utilisant la résolution de problèmes d'optimisation (OBBT, pour *Optimality-Based Bound Tightening*).

FBBT : Feasibility-Based Bound Tightening (Belotti *et al.*) [1]

Les techniques appartenant à la famille FBBT visent à utiliser les contraintes du problème, et à les combiner pour se rendre compte que si une variable se situe dans un certain intervalle, une des contraintes est violée. On en conclut que le domaine de cette variable peut être réduit.

La technique la plus étudiée est celle de l'arithmétique d'intervalle, utilisée sur les domaines des

variables. En appliquant l'arithmétique d'intervalles sur les contraintes linéaires d'un problème, on obtient une technique appartenant à la famille FBBT. Ces techniques sont en général rapides à effectuer avec l'outil numérique.

Un exemple simple pour illustrer cette méthode est le cas d'un problème dont une des contraintes exprime le fait que la somme des variables est plus petit que 1. En outre, chaque variable est positive. On déduit de cela que chaque variable doit être plus petite que 1.

OBBT : Optimality-Based Bound Tightening (Sahinidis *et al.*) [42]

Les techniques OBBT s'appuient sur la résolution de problèmes d'optimisation pour déterminer de nouvelles bornes sur les variables de notre problème. La technique suivante est rudimentaire et coûteuse en ressources de calcul. L'idée est de résoudre deux problèmes d'optimisation par variables v_i. Ces problèmes s'inspirent le plus souvent de la relaxation présente dans l'algorithme de *branch-and-bound* spatial Pour ces deux problèmes, la fonction objectif est la fonction qui à v associe v_i. L'un des problèmes cherche à maximiser v_i, tandis que l'autre cherche à minimiser v_i. On obtient alors deux nouvelles bornes ℓ_i et u_i sur v_i.

Pour illustrer cela, reprenons un algorithme de *branch-and-bound* s'appuyant sur la relaxation $(OPF - LP)$. Les problèmes résolues pour contracter les bornes d'une variable v_i sont alors :

$$(OBBT_i^{min}) \begin{cases} \min\limits_{v,W} \quad v_i \\ \text{s.t.} \\ \quad (1.2), \\ \quad (v, W) \in MCC, \\ \quad v \in \mathbb{R}^n, \\ \quad W \in \mathcal{M}_n(\mathbb{R}). \end{cases}$$

$$(OBBT_i^{max}) \begin{cases} \max\limits_{v,W} \quad v_i \\ \text{s.t.} \\ \quad (1.2), \\ \quad (v, W) \in MCC, \\ \quad v \in \mathbb{R}^n, \\ \quad W \in \mathcal{M}_n(\mathbb{R}). \end{cases}$$

Ces problèmes sont linéaires.

Une autre catégorie de réductions OBBT, elles beaucoup plus rapides à effectuer, sont connues sous le nom de RCBT, pour *Reduced-Cost Bound Tightning* [43]. Elles consistent à utiliser les coûts réduits associés à une variable après avoir résolu un programme d'optimisation convexe pour tenter de modifier une des deux bornes de cette variable. Supposons que l'on vienne de calculer un programme convexe de minimisation et que sa valeur soit LB et que UB soit une borne supérieure valide sur ce programme. Supposons que la solution $\bar{v}$ de ce programme donne $\bar{v}_i = \ell_i$. Alors en utilisant le coût réduit c_i associé à $\bar{v}_i$, on va pouvoir modifier la valeur de u_i. En effet, on a : $u_i \leq \ell_i + \frac{UB-LB}{c_i}$. Notons que symétriquement, si $\bar{v}_i = u_i$, on pourra modifier la valeur de ℓ_i : $\ell_i \geq u_i - \frac{UB-LB}{c_i}$ Notons que ces programmes convexes peuvent être les relaxations calculées aux différents nœuds mais aussi les problèmes OBBT présentées plus haut.

Renforcement des techniques de contraction

Il est possible de renforcer les techniques de contraction vues ci-dessus en cherchant à supprimer des points du domaine des variables qui pourraient être faisables, mais qui sont sous-optimaux au vu du critère à minimiser. Ceci est fait en s'appuyant sur la connaissance d'une borne supérieure UB du problème. Plus cette borne est précise, plus ceci est efficace.

La technique la plus simple consiste à ajouter une contrainte au problème. Cette contrainte stipule que la fonction coût doit être plus basse que notre borne UB (pour un problème de minimisation). L'ajout de cette contrainte peut écarter des points réalisables, mais n'écarte pas les points optimaux. Elle permet d'obtenir un renforcement de la réduction par arithmétique d'intervalle sur les contraintes, grâce, justement, à l'ajout d'une contrainte. Elle permet aussi d'avoir une contrainte supplémentaire

dans les problèmes OBBT ci-dessus. Au lieu d'utiliser la fonction coût du problème d'origine, un sous-estimateur de cette même fonction peut être employé. Par exemple, en reprenant le cas d'un algorithme de *branch-and-bound* s'appuyant sur la relaxation $(QCQP - LP)$, la contrainte $< A, W > + c^t v \leq UB$, est ajoutée aux problèmes $(OBBT_i^{min})$ et $(OBBT_i^{max})$ pour renforcer les contractions de bornes obtenues.

La réduction de bornes dans le cas de l'Optimal Power Flow

Les contractions de bornes sur les variables de l'Optimal Power Flow ont été étudiées en détail dans [11].

1.2.3 La méthode MIQCR

La méthode MIQCR, pour *Mixed-Integer Quadratic Convex Reformulation* [44, 45, 46] est une méthode développée pour résoudre des QCQP en variables mixtes. Ici, on la présente utilisée uniquement sur le problème à variables continues $(QCQP)$, pour faire plus rapidement le lien avec l'OPF. C'est une méthode reposant sur deux étapes :

1. La reformulation de $(QCQP)$ en un autre problème quadratique, mais où la fonction objectif et les contraintes sont convexes, à l'exception d'une seule d'entre elles. Cette reformulation est optimale parmi une famille étendue de reformulations quadratiques. Elle s'obtient après résolution d'une relaxation SDP du problème d'origine.

2. La résolution de ce problème reformulé avec un algorithme de *branch-and-bound* spatial, appuyé sur une relaxation quadratique convexe.

La méthode MIQCR exploite les inégalités de McCormick [24] *(MCC)*. Ces inégalités sont valides dans le cadre de QCQP, quand W est définie par : $W = vv^t$. On les ajoute à $(QCQP)$.

La famille de reformulations quadratiques envisagées par MIQCR est définie comme suit : chaque élément de la famille est identifié par une matrice semi-définie positive notée S. On ajoute également

la matrice de variables W. La reformulation quadratique associée à S est :

$$(QCQP-S)\begin{cases} \min\limits_{v,W} \quad v^tSv+<A-S,W>+c^tv & (1.11)\\ \text{s.t.} \\ \quad (1.4), \\ \quad (v,W)\in MCC, \\ \quad W=vv^t. & (1.12)\end{cases}$$

On remarque que les contraintes de $(QCQP-S)$ sont toutes linéaires, à l'exception de (1.12) qui est quadratique non-convexe. L'objectif est maintenant quadratique convexe car S est SDP.

À chaque reformulation $(QCQP-S)$, on fait correspondre une relaxation quadratique convexe obtenue en supprimant la seule contrainte non-convexe (1.12). Chacune de ces relaxations fournit une borne inférieure sur $(QCQP)$. L'objectif est alors de trouver la reformulation fournissant la meilleure relaxation. Cela revient à trouver une matrice S^* que l'on appelera optimale. Dans [46], les auteurs montrent que la matrice S^* est obtenue comme matrice de variables optimales duales de la relaxation SDP suivante :

$$(QCQP-SDP-MIQCR)\begin{cases} \min\limits_{v,W} \quad <A,W>+c^tv \\ \text{s.t.} \\ \quad (1.4),(1.5), \\ \quad (v,W)\in MCC,\end{cases}$$

De plus, il est démontré que la relaxation associée à la reformulation avec S^* a même valeur que la relaxation SDP $(QCQP-SDP-MIQCR)$, en étant uniquement quadratique et convexe. Une fois que la reformulation optimale est construite, elle est résolue avec un algorithme de *branch-and-bound* spatial similaire à ce qui est présenté en Section 1.2.2. Cet algorithme s'appuie sur la relaxation quadratique convexe $(QCQP-S^*)$.

Chapitre 2

Résolution exacte par reformulation quadratique convexe : la méthode RC-OPF

Contenu

2.1 Passage en variables réelles . . . 75
2.2 Phase 1 : Reformulation quadratique optimale . . . 77
2.2.1 Famille de reformulations quadratiques exactes . . . 77
2.2.2 La relaxation du rang . . . 78
2.2.3 Reformulation optimale . . . 81
2.2.4 Extension aux contraintes de limitation du flux d'intensité . . . 84
2.3 Comparaison avec la méthode MIQCR . . . 85
2.4 Phase 2 : Algorithme de *branch-and-bound* spatial . . . 89
2.4.1 Présentation générale de l'algorithme . . . 89
2.4.2 Choix de la variable de branchement . . . 91
2.4.3 En quelle valeur couper ? . . . 93
2.4.4 Choix du nœud suivant . . . 93
2.4.5 Autres paramètres . . . 94
2.4.6 Notre méthode de résolution : RC-OPF . . . 94
2.5 Illustration de RC-OPF sur l'instance WB2 . . . 96
2.5.1 Présentation de l'instance . . . 96
2.5.2 Substitution de variables . . . 98
2.5.3 Formulation du problème en variables complexes . . . 98
2.5.4 Phase 0 : Réécriture en variables réelles . . . 100
2.5.5 Phase 1.1 : Résolution de la relaxation du rang . . . 101
2.5.6 Phase 1.2 : Construction de la reformulation quadratique optimale . . . 102
2.5.7 Phase 2 : Résolution à l'aide de l'algorithme de *branch-and-bound* . . . 102
2.6 Extension au problème $(OPF - Q)$ . . . 103
2.6.1 Reformulation quadratique . . . 103
2.6.2 Relaxation du rang . . . 105
2.6.3 Reformulation optimale . . . 108

2.6.4 Ajout des inégalités de McCormick 112
2.6.5 Conclusion sur la résolution de $(OPF - Q)$ avec `RC-OPF` 116

Ce troisième chapitre constitue le cœur de ce travail de thèse. Il présente notre méthode de résolution exacte pour une instance d'un problème OPF, appelée `RC-OPF`. Notre méthode fonctionne en deux phases. La première phase consiste en la reformulation exacte du problème OPF. Le problème reformulé à l'issue de la première phase a la propriété d'être optimal parmi une famille de reformulations quadratiques de l'OPF, dans le sens où sa relaxation convexe fournit une borne inférieure maximale. La seconde phase correspond à la résolution optimale du problème reformulé à l'aide d'un algorithme de *branch-and-bound* spatial, utilisant une relaxation quadratique convexe. Dans un premier temps, nous considérons la résolution de la variante, plus épurée, $(OPF - L)$. Nous commençons par formuler $(OPF - L)$ comme un QCQP en variables réelles.

2.1 Passage en variables réelles

On rappelle ici la modélisation en nombres complexes de $(OPF - L)$:

$$(OPF - L) \begin{cases} \min\limits_{V} \quad \sum\limits_{i \in \mathcal{P}} \mathbf{c}_{1i} V^* \left(\dfrac{\mathbf{Y}_i^M + \mathbf{Y}_i^{M*}}{2} \right) V + \mathbf{c}_{1i} Re(\mathbf{S}_i^c) + \mathbf{c}_{0i} \\ \text{s.t.} \\ \quad V^* \mathbf{Y}_i^M V = -\mathbf{S}_i^c, \quad i \in \mathcal{C} & (24) \\ \quad \underline{\mathbf{p}_i} - Re(\mathbf{S}_i^c) \leq V^* \left(\dfrac{\mathbf{Y}_i^M + \mathbf{Y}_i^{M*}}{2} \right) V \leq \overline{\mathbf{p}_i} - Re(\mathbf{S}_i^c), \quad i \in \mathcal{P} & (25) \\ \quad \underline{\mathbf{q}_i} - Im(\mathbf{S}_i^c) \leq V^* \left(\dfrac{\mathbf{Y}_i^M - \mathbf{Y}_i^{M*}}{2\mathbf{i}} \right) V \leq \overline{\mathbf{q}_i} - Im(\mathbf{S}_i^c), \quad i \in \mathcal{P} & (26) \\ \quad \underline{\mathbf{v}_i} \leq |V_i| \leq \overline{\mathbf{v}_i}, \quad \forall i \in \mathcal{N} & (10) \end{cases}$$

Rappel : On note n le nombre de nœuds du réseau électrique étudié, et on suppose que les n nœuds du réseau sont numérotés de 1 à n. On note n_p le nombre de nœuds reliés à des producteurs, et n_c le nombre de nœuds reliés à des consommateurs.

Pour passer en variables réelles, on utilise la modélisation cartésienne des nombres complexes, où $v \in \mathbb{R}^{2n}$ est le vecteur des parties réelles et imaginaires des tensions aux différents nœuds du réseau. Les n premières composantes de v correspondent aux parties réelles, placées dans l'ordre croissant des numéros des nœuds du réseau. Les n dernières composantes correspondent aux parties imaginaires, placées dans le même ordre. Ainsi pour i entre 1 et n, $v_i = Re(V_i)$, tandis que $v_{i+n} = Im(V_i)$. Avec ce

choix de représentation, on observe que le réel V^*MV, pour une matrice hermitienne M quelconque, s'écrit : $v^t \begin{bmatrix} Re(M) & -Im(M) \\ Im(M) & Re(M) \end{bmatrix} v$. On obtient alors la modélisation suivante pour $(OPF-L)$:

$$(OPF-L)\begin{cases} \min\limits_{v} \quad f(v) = v^t\mathbf{C}_1 v + \mathbf{c}_0 \\ \text{s.t.} \\ \quad v^t\mathbf{Y}_i^R v = -\mathbf{p}_i^c, \quad i \in \mathcal{C} & (2.1) \\ \quad v^t\mathbf{Y}_i^I v = -\mathbf{q}_i^c, \quad i \in \mathcal{C} & (2.2) \\ \quad \underline{\mathbf{p}_i} - \mathbf{p}_i^c \leq v^t\mathbf{Y}_i^R v \leq \overline{\mathbf{p}_i} - \mathbf{p}_i^c, \quad i \in \mathcal{P} & (2.3) \\ \quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq v^t\mathbf{Y}_i^I v \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i \in \mathcal{P} & (2.4) \\ \quad \underline{\mathbf{v}_i}^2 \leq v_i^2 + v_{i+n}^2 \leq \overline{\mathbf{v}_i}^2, \quad i \in \mathcal{N} & (2.5) \end{cases}$$

Rappel de la nature de la fonction objectif et des contraintes :

- f est la fonction objectif. $\mathbf{C}_1 \in \mathbb{M}_{2n}(\mathbb{R})$ est la matrice de coût. Elle est égale à : $\sum_{i\in\mathcal{P}} \mathbf{c}_{1i}\mathbf{Y}_i^R$. $\mathbf{c}_0 \in \mathbb{R}$ est la constante de coût, elle vaut : $\sum_{i\in\mathcal{P}} \mathbf{c}_{1i}\mathbf{p}_i^c + \mathbf{c}_{0i}$
- (2.1) est le bilan de puissance active aux différents nœuds de consommation.
- (2.2) est le bilan de puissance réactive aux différents nœuds de consommation.
- (2.3) est le bilan de puissance active aux différents nœuds de production. $\underline{\mathbf{p}} \in \mathbb{R}^{n^g}$ est le vecteur des bornes inférieures sur cette puissance active. $\overline{\mathbf{p}} \in \mathbb{R}^{n^g}$ est celui des bornes supérieures.
- (2.4) est le bilan de puissance réactive aux différents nœuds de production. $\underline{\mathbf{q}} \in \mathbb{R}^{n^g}$ est le vecteur des bornes inférieures sur la puissance réactive produite. $\overline{\mathbf{q}} \in \mathbb{R}^{n^g}$ est celui des bornes supérieures.
- (2.5) représente les bornes sur le carré du module de la tension aux différents nœuds du réseau. $\underline{\mathbf{v}} \in \mathbb{R}^n$ est le vecteur des bornes inférieures sur ce module de tension. $\overline{\mathbf{v}} \in \mathbb{R}^n$ est celui des bornes supérieures.
- $\mathbf{Y}_i^R \in \mathbb{S}_{2n}(\mathbb{R})$ et $\mathbf{Y}_i^I \in \mathbb{S}_{2n}(\mathbb{R})$, pour i entre 1 et n, sont des matrices obtenues à partir des matrices d'admittance $\mathbf{Y}_i^M$, comme expliqué dans la transformation plus haut. Elles permettent d'exprimer les parties réelles et imaginaires de la puissance résultante fournie au réseau par l'extérieur en fonction de la tension en chaque nœud i.
- $\mathbf{p}^c \in \mathbb{R}^n$ est le vecteur des parties réelles des puissances consommées en chaque nœud du réseau.
- $\mathbf{q}^c \in \mathbb{R}^n$ est le vecteur des parties imaginaires des puissances consommées en chaque nœud du réseau.

Ce chapitre se divise en 5 sections. Dans la première, on donne les éléments théoriques permettant d'obtenir la reformulation quadratique optimale dans le cadre de la variante $(OPF-L)$. Dans la seconde on explique les différences entre notre reformulation et celle de la méthode MIQCR. On démontre que notre reformulation dispose des mêmes propriétés mais que son calcul est plus efficace. Dans la troisième, on présente l'algorithme de *branch-and-bound* spatial utilisé pour résoudre la reformulation quadratique optimale de $(OPF-L)$. Dans la quatrième, on donne une illustration de cette méthode pour résoudre une petite instance de test sur un réseau électrique contenant deux nœuds. Dans la dernière, on étend notre méthode `RC-OPF` au cas de la variante $(OPF-Q)$.

2.2 Phase 1 : Reformulation quadratique optimale

Dans cette section, on montre comment obtenir une reformulation quadratique optimale de $(OPF-L)$ en utilisant la solution optimale duale de la relaxation du rang de $(OPF-L)$.

2.2.1 Famille de reformulations quadratiques exactes

Tout d'abord, on introduit une famille de reformulations quadratiques et exactes. Soit $S \in \mathbb{S}_{2n}(\mathbb{R})$ une matrice semi-définie positive et $W \in \mathbb{S}_{2n}(\mathbb{R})$ une matrice de variables représentant le produit vv^t entre les variables de tension. Le problème $(OPF-L_S)$ est alors une reformulation exacte de $(OPF-L)$.

$$(OPF-L_S)\begin{cases} \min\limits_{v,W} \quad f_S(v,W) = v^tSv + <\mathbf{C}_1 - S, W> + \mathbf{c}_0 \\ \text{s.t.} \\ \quad <\mathbf{Y}_i^R, W> = -\mathbf{p}_i^c, \quad i \in \mathcal{C} & (2.6) \\ \quad <\mathbf{Y}_i^I, W> = -\mathbf{q}_i^c, \quad i \in \mathcal{C} & (2.7) \\ \quad \underline{\mathbf{p}_i} - \mathbf{p}_i^c \leq <\mathbf{Y}_i^R, W> \leq \overline{\mathbf{p}_i} - \mathbf{p}_i^c, \quad i \in \mathcal{P} & (2.8) \\ \quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq <\mathbf{Y}_i^I, W> \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i \in \mathcal{P} & (2.9) \\ \quad \underline{\mathbf{v}_i}^2 \leq W_{i,i} + W_{i+n,i+n} \leq \overline{\mathbf{v}_i}^2, \quad i \in \mathcal{N} & (2.10) \\ \quad W = vv^t & (2.11) \end{cases}$$

$(OPF-L_S)$ est bien une reformulation exacte de $(OPF-L)$ car quand (2.11) est vérifiée, les fonctions objectif f et f_S ont même valeur, tandis que les ensembles de contraintes (2.1)-(2.5) et (2.6)-(2.10) définissent les mêmes espaces admissibles. Notons que cette reformulation est obtenue en deux temps,

d'abord on linéarise les termes quadratiques dans la fonction objectif ainsi que dans les contraintes, puis on introduit un terme correctif dans la fonction objectif : $v^t Sv - < S, W >$. Le seul terme non linéaire de la fonction objectif est maintenant : $v^t Sv$. La fonction objectif est quadratique convexe car S est une matrice semi-définie positive. Les contraintes, excepté la contrainte (2.11), sont linéaires. En prenant $S = 0_{2n}$, on obtient la « reformulation linéaire » canonique de $(OPF - L)$, c'est-à-dire le problème $(QCQP - LP)$ pour $(OPF - L)$ (voir Section 1.1.3). En faisant parcourir l'ensemble des matrices semi-définies positives à S, on fait décrire à $(OPF - L_S)$ la famille de reformulations quadratiques et exactes de $(OPF - L)$, considérée par la méthode MIQCR.

2.2.2 La relaxation du rang

La relaxation Semi-Définie Positive (SDP) de l'état de l'art, appelée relaxation du rang, de $(OPF-L)$ est le programme d'optimisation suivant :

$$(SDP - L) \begin{cases} \min\limits_{W} \quad < \mathbf{C}_1, W > + \mathbf{c}_0 & (2.12) \\ \text{s.t.} \\ \quad (2.6) - (2.10) \\ \quad W \succeq 0 & (2.13) \end{cases}$$

$(SDP-L)$ est une relaxation de $(OPF-L)$. Pour s'en convaincre, prenons la reformulation quadratique obtenue avec $S = 0_{2n}$. La contrainte (2.11) peut se relâcher en (2.13). $(SDP - L)$ est obtenue à partir de $(OPF - L_{0_{2n}})$ en effectuant cette relaxation de la contrainte (2.11). On appelle cette relaxation la relaxation du rang [9], car c'est la contrainte $rg(W) = 1$ qui est relaxée dans (2.11) pour se ramener à (2.13).

Le dual de $(SDP-L)$ s'écrit :

$$
(D-SDP-L)\begin{cases}
\max\limits_{\lambda^R,\lambda^I,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma}} g(\lambda^R,\lambda^I,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma}) = (\mathbf{p}^c)^t\lambda^R + \\
\quad (\mathbf{q}^c)^t\lambda^I + (\underline{\mathbf{p}}-\mathbf{p}^c)^t\underline{\mu}^R + (\mathbf{p}^c-\overline{\mathbf{p}})^t\overline{\mu}^R + \\
\quad (\underline{\mathbf{q}}-\mathbf{q}^c)^t\underline{\mu}^I + (\mathbf{q}^c-\overline{\mathbf{q}})^t\overline{\mu}^I + (\underline{\mathbf{v}}^2)^t\underline{\gamma} - (\overline{\mathbf{v}}^2)^t\overline{\gamma} + \mathbf{c}_0 \\
\text{s.t.} \\
\quad \mathbf{C}_1 + \sum\limits_{i\in\mathcal{C}}\left(\lambda_i^R\mathbf{Y}_i^R + \lambda_i^I\mathbf{Y}_i^I\right) + \sum\limits_{i\in\mathcal{P}}\left((\overline{\mu}_i^R-\underline{\mu}_i^R)\mathbf{Y}_i^R + (\overline{\mu}_i^I-\underline{\mu}_i^I)\mathbf{Y}_i^I\right) \\
\quad + \sum\limits_{i\in\mathcal{N}}\left((\overline{\gamma}_i-\underline{\gamma}_i)1_{i,i} + (\overline{\gamma}_i-\underline{\gamma}_i)1_{i+n,i+n}\right) \succeq 0, & (2.14)\\
\quad \lambda_i^R \in \mathbb{R}, \quad i\in\mathcal{C} & (2.15)\\
\quad \lambda_i^I \in \mathbb{R}, \quad i\in\mathcal{C} & (2.16)\\
\quad \underline{\mu}_i^R \geq 0, \quad i\in\mathcal{P} & (2.17)\\
\quad \overline{\mu}_i^R \geq 0, \quad i\in\mathcal{P} & (2.18)\\
\quad \underline{\mu}_i^I \geq 0, \quad i\in\mathcal{P} & (2.19)\\
\quad \overline{\mu}_i^I \geq 0, \quad i\in\mathcal{P} & (2.20)\\
\quad \underline{\gamma}_i \geq 0, \quad i\in\mathcal{N} & (2.21)\\
\quad \overline{\gamma}_i \geq 0, \quad i\in\mathcal{N} & (2.22)
\end{cases}
$$

Où :

- $\lambda^R \in \mathbb{R}^{n_c}$ est le vecteur des variables duales associées aux contraintes (2.6).
- $\lambda^I \in \mathbb{R}^{n_c}$ est le vecteur des variables duales associées aux contraintes (2.7).
- $\underline{\mu}^R \in \mathbb{R}^{n_p}$ et $\overline{\mu}^R \in \mathbb{R}^{n_p}$ sont les vecteurs des variables duales associées respectivement aux contraintes de supériorité et d'infériorité (2.8).
- $\underline{\mu}^I \in \mathbb{R}^{n_p}$ et $\overline{\mu}^I \in \mathbb{R}^{n_p}$ sont les vecteurs des variables duales associées respectivement aux contraintes de supériorité et d'infériorité (2.9).
- $\underline{\gamma} \in \mathbb{R}^n$ et $\overline{\gamma} \in \mathbb{R}^n$ sont les vecteurs des variables duales associées respectivement aux contraintes de supériorité et d'infériorité (2.10).
- $\underline{\mathbf{v}}^2$ et $\overline{\mathbf{v}}^2$ représentent les vecteurs de $\mathbb{R}^n$ dont la i-ème composante vaut respectivement $\underline{\mathbf{v}_i}^2$ et $\overline{\mathbf{v}_i}^2$.
- Pour tout i entre 1 et $2n$, $1_{i,i}$ est la matrice de taille $2n \times 2n$ dont toutes les composantes sont nulles sauf la i-ème composante de la diagonale, qui vaut 1.

On sait que pour un problème SDP, si le primal est réalisable et le dual est strictement réalisable, alors il n'y a pas de saut de dualité. De plus l'ensemble des solutions est non vide et borné. La proposition suivante est déjà connue [39], mais nous la rappelons ici pour en proposer une démonstration qui sera utile par la suite.

Proposition 1. *[39] Pour un problème $(OPF - L)$ réalisable, le saut de dualité entre la relaxation du rang $(SDP - L)$ et son dual $(D - SDP - L)$ est nul. Il y a alors dualité forte pour la relaxation du rang.*

Preuve. Pour démontrer ce résultat, nous procédons en deux temps, nous montrons i) $(SDP - L)$ est réalisable, puis ii) $(D - SDP - L)$ est strictement réalisable.

i) Comme $(OPF - L)$ est réalisable, notons v^* une de ces solutions réalisables.
Dans ce cas, $W = v^*(v^*)^t$ est une solution réalisable de (SDP). En particulier avec ces valeurs, on vérifie bien (2.13). $(SDP - L)$ est donc réalisable.

ii) Dans un premier temps posons :
— $\lambda_i^R = 0, \quad \forall i \in \mathcal{C}$.
— $\lambda_i^I = 0, \quad \forall i \in \mathcal{C}$.
— $\underline{\mu}_i^R = 1, \quad \forall i \in \mathcal{P}$.
— $\overline{\mu}_i^R = 1, \quad \forall i \in \mathcal{P}$.
— $\underline{\mu}_i^I = 1, \quad \forall i \in \mathcal{P}$.
— $\overline{\mu}_i^I = 1, \quad \forall i \in \mathcal{P}$.
— $\underline{\gamma}_i = 1, \quad \forall i \in \mathcal{N}$.
— $\overline{\gamma}_i = \gamma, \quad \forall i \in \mathcal{N}$.
Où γ est une quantité strictement positive encore à définir.
Avec ces valeurs pour les variables duales, les contraintes (2.15)-(2.22) sont strictement vérifiées.
De plus (2.14) s'écrit $\mathbf{C}_1 - \sum_{i\in\mathcal{N}}(1_{i,i} + 1_{i+n,i+n}) + \gamma \sum_{i\in\mathcal{N}}(1_{i,i} + 1_{i+n,i+n}) \succeq 0$.
On remarquera également que $\sum_{i\in\mathcal{N}}(1_{i,i} + 1_{i+n,i+n}) = \mathcal{I}_{2n}$, qui est la matrice identité de taille $2n \times 2n$.
(2.14) s'écrit : $\mathbf{C}_1 + (\gamma - 1)\mathcal{I}_{2n} \succeq 0$. Pour γ assez grand, on obtient $\gamma > 0$ et (2.14) strictement valide.

Ainsi avec les valeurs données pour les variables duales, (2.14) est strictement vérifiée, ce qui achève

de prouver que $(D - SDP - L)$ est strictement réalisable. $\square$

Cette proposition est importante, car nous allons utiliser les variables duales de $(SDP - L)$ pour construire une reformulation quadratique optimale de $(OPF - L)$. Notons que cette proposition n'est pas vraie pour un problème d'optimisation semi-définie positif quelconque. Pour l'OPF c'est la contrainte de boule (2.10) qui est déterminante.

2.2.3 Reformulation optimale

Après avoir introduit la famille des reformulations quadratiques $(OPF - L_S)$, la relaxation SDP appelée relaxation du rang ainsi que son dual, nous allons voir comment obtenir une reformulation quadratique dite optimale de $(OPF - L)$ à partir de cette relaxation du rang.

Dans la Section 2.2.1 nous avons décrit comment reformuler le problème $(OPF - L)$ en introduisant une matrice SDP notée S. Le choix de cette matrice a un impact sur la reformulation obtenue. Nous cherchons alors à déterminer la matrice S fournissant la meilleure reformulation $(OPF - L_S)$ selon un certain critère. Ce critère est la maximisisation de la valeur d'une relaxation quadratique convexe de $(OPF - L_S)$.

Le problème obtenu par reformulation sera résolu avec un algorithme de type *branch-and-bound* spatial. En chaque nœud de l'arbre généré par cet algorithme sera résolue une relaxation quadratique convexe obtenue en relachant la contrainte (2.11) et en la remplaçant par les inégalités de McCormick. Celle-ci fournira la borne inférieure de la racine de l'arbre de *branch-and-bound*. Il semble légitime de chercher à maximiser la valeur de cette borne inférieure initiale. Tout d'abord nous allons travailler comme si les inégalités de McCormick n'étaient pas introduites dans la relaxation. La relaxation à la racine que l'on considère pour l'instant consiste uniquement à supprimer la contrainte (2.11) dans $(OPF - L_S)$, car c'est la seule contrainte non-convexe du problème.

$$(\overline{OPF - L_S}) \begin{cases} \min\limits_{v,W} \quad f_S(v, W) \\ \quad \text{s.t.} \\ \qquad (2.6) - (2.10) \end{cases}$$

La fonction objectif de $(OPF - L_S)$ étant quadratique et convexe, la relaxation considérée est

alors bien quadratique convexe. La construction d'une reformulation optimale se fait en cherchant une solution optimale du problème d'optimisation (P) suivant :

$$(P)\begin{cases} \max\limits_{S} \quad val(\overline{OPF - L_S}) \\ \text{s.t.} \\ \quad S \succeq 0. \end{cases} \tag{2.23}$$

Théorème 1. *Une solution optimale S^* de (P) peut s'obtenir à l'aide d'une solution optimale du problème $(D - SDP - L)$.*

Preuve. Pour démontrer ce résultat, nous procédons par étapes. i) Nous réécrivons le problème $(D - SDP - L)$ en faisant apparaître une matrice de variables S. ii) Nous montrons que $val(P) \geq val(\overline{OPF - L_{S^*}}) \geq val(SDP - L)$, où S^* provient d'une solution optimale de $(D - SDP - L)$ iii) Nous montrons que $val(SDP - L) \geq val(P)$. Ceci nous permet de conclure que $val(P) = val(\overline{OPF - L_{S^*}}) = val(SDP - L)$.

i) Tout d'abord, nous réécrivons le problème $(D - SDP - L)$ en introduisant une matrice de variables d'écarts notée S.

$$(D - SDP - L)\begin{cases} \max\limits_{S,\lambda^R,\lambda^I,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma},} g(\lambda^R, \lambda^I, \underline{\mu}^R, \overline{\mu}^R, \underline{\mu}^I, \overline{\mu}^I, \underline{\gamma}, \overline{\gamma}) \\ \text{s.t.} \\ \quad S = \mathbf{C}_1 + \sum\limits_{i\in\mathcal{C}} \left(\lambda_i^R \mathbf{Y}_i^R + \lambda_i^I \mathbf{Y}_i^I\right) + \sum\limits_{i\in\mathcal{P}} \left((\overline{\mu}_i^R - \underline{\mu}_i^R)\mathbf{Y}_i^R + (\overline{\mu}_i^I - \underline{\mu}_i^I)\mathbf{Y}_i^I\right) \\ \quad + \sum\limits_{i\in\mathcal{N}} \left((\overline{\gamma}_i - \underline{\gamma}_i)1_{i,i} + (\overline{\gamma}_i - \underline{\gamma}_i)1_{i+n,i+n}\right), \\ \quad (2.15) - (2.22), (2.23) \end{cases} \tag{2.24}$$

Pour S une matrice SDP fixée, notons $(D - SDP - L(S))$ le programme d'optimisation résultant de la maximisation de la fonction objectif g sous les contraintes (2.15)-(2.22) et (2.24), de telle sorte que le programme suivant soit équivalent à $(D - SDP - L)$:

$$(D - SDP - L)\begin{cases} \max\limits_{S} \quad val(D - SDP - L(S)) \\ \text{s.t.} \\ \quad (2.23). \end{cases}$$

$(D - SDP - L(S))$ est un programme d'optimisation linéaire, il est donc équivalent à son dual. En introduisant la matrice de variables duales W correspondant à la contrainte (2.24), et en dualisant

$(D - SDP - L(S))$, on construit le problème primal noté $(P - SDP - L(S))$:

$$(P - SDP - L(S)) \begin{cases} \min\limits_{W} & < \mathbf{C}_1 - S, W > + \mathbf{c}_0 \\ \text{s.t.} & \\ & (2.6) - (2.10) \end{cases}$$

$(D - SDP - L)$ est alors équivalent au problème suivant :

$$(D - SDP - L) \begin{cases} \max\limits_{S} & val(P - SDP - L(S)) \\ \text{s.t.} & \\ & (2.23) \end{cases} \tag{2.25}$$

ii) Montrons que $val(P) \geq val(\overline{OPF - L_{S^*}}) \geq val(SDP - L)$.
Soit S^* une solution optimale de $(D - SDP - L)$. Il s'ensuit que $(P - SDP - L(S^*))$ a même valeur optimale que $(D - SDP - L)$. On ajoute un terme positif dans la fonction objectif de $(P - SDP - L(S^*))$. Ce terme est : $v^t S^* v$. Le problème obtenu est alors $(\overline{OPF - L_{S^*}})$. Ainsi, en résolvant la relaxation du rang $(SDP - L)$, on est à même de construire la reformulation $(OPF - L_{S^*})$ dont la relaxation quadratique convexe $(\overline{OPF - L_{S^*}})$ a au moins la valeur de cette relaxation du rang.

iii) Nous montrons maintenant que $val(P) \leq val(SDP - L)$. Nous n'avons qu'à prouver que $val(SDP - L) \geq val(\overline{OPF - L_S})$, pour toute matrice $S \succeq 0$.

Pour cela, nous allons montrer que pour toute solution réalisable de $(SDP - L)$, on peut construire une solution réalisable de $(\overline{OPF - L_S})$ ayant une valeur inférieure. Soit (W^*) une solution de $(SDP - L)$. $(W = W^*, v = 0)$ est alors une solution réalisable de $(\overline{OPF - L_S})$. Comparons leur valeur en soustrayant la fonction objectif de $(SDP - L)$ à celle de $(\overline{OPF - L_S})$:

$$< \mathbf{C}_1, W^* > - < S, W^* > - < \mathbf{C}_1, W^* > + \mathbf{c}_0 - \mathbf{c}_0 \quad = \quad < S, -W^* > .$$

Or, comme W^* vérifie la contrainte (2.13) et S étant SDP, $< S, -W^* > \quad \leq \quad 0$. On a alors démontré que $val(SDP - L) \geq val(\overline{OPF - L_S})$, et ce quelque soit la matrice $S \succeq 0$ considérée.

Ceci permet de conclure quant à la validité du théorème. □

De plus nous pouvons déduire de la proposition précédente le corollaire suivant :

Corrolaire 1. *Les relaxations convexes* $(SDP-L)$ *et* $(\overline{OPF-L_{S^*}})$ *ont même valeur.*

Nous obtenons donc comme reformulation optimale :

$$(OPF-L_{S^*})\begin{cases} \min\limits_{v,W} \quad f_{S^*}(v,W) = v^t S^* v \\ \quad - < \sum\limits_{i\in\mathcal{C}} \left(\lambda_i^{R^*}\mathbf{Y}_i^R + \lambda_i^{I^*}\mathbf{Y}_i^I\right) \\ \qquad + \sum\limits_{i\in\mathcal{P}} \left((\overline{\mu}_i^{R^*} - \underline{\mu}_i^{R^*})\mathbf{Y}_i^R + (\overline{\mu}_i^{I^*} - \underline{\mu}_i^{I^*})\mathbf{Y}_i^I\right) \\ \qquad + \sum\limits_{i\in\mathcal{N}} \left((\overline{\gamma}_i{}^* - \underline{\gamma}_i{}^*)1_{i,i} + (\overline{\gamma}_i{}^* - \underline{\gamma}_i{}^*)1_{i+n,i+n}\right), W > + \mathbf{c}_0 \\ \text{s.t.} \\ \quad < \mathbf{Y}_i^R, W >= -\mathbf{p}_i^c, \quad i\in\mathcal{C} & (2.6) \\ \quad < \mathbf{Y}_i^I, W >= -\mathbf{q}_i^c, \quad i\in\mathcal{C} & (2.7) \\ \quad \underline{\mathbf{p}_i} - \mathbf{p}_i^c \leq < \mathbf{Y}_i^R, W > \leq \overline{\mathbf{p}_i} - \mathbf{p}_i^c, \quad i\in\mathcal{P} & (2.8) \\ \quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq < \mathbf{Y}_i^I, W > \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i\in\mathcal{P} & (2.9) \\ \quad \underline{\mathbf{v}_i}^2 \leq W_{i,i} + W_{i+n,i+n} \leq \overline{\mathbf{v}_i}^2, \quad i\in\mathcal{N} & (2.10) \\ \quad W = vv^t. & (2.11) \end{cases}$$

Où $(\lambda^{R^*}, \lambda^{I^*}, \underline{\mu}^{R^*}, \overline{\mu}^{R^*}, \underline{\mu}^{I^*}, \overline{\mu}^{I^*}, \underline{\gamma}^*, \overline{\gamma}^*)$ est une solution optimale de $(DSPD-L)$. De plus :

$$S^* = \mathbf{C}_1 + \sum_{i\in\mathcal{C}} \left(\lambda_i^{R^*}\mathbf{Y}_i^R + \lambda_i^{I^*}\mathbf{Y}_i^I\right) + \sum_{i\in\mathcal{P}} \left((\overline{\mu}_i^{R^*} - \underline{\mu}_i^{R^*})\mathbf{Y}_i^R + (\overline{\mu}_i^{I^*} - \underline{\mu}_i^{I^*})\mathbf{Y}_i^I\right) + \sum_{i\in\mathcal{N}} \left((\overline{\gamma}_i{}^* - \underline{\gamma}_i{}^*)1_{i,i} + (\overline{\gamma}_i{}^* - \underline{\gamma}_i{}^*)1_{i+n,i+n}\right).$$

2.2.4 Extension aux contraintes de limitation du flux d'intensité

On s'intéresse à prendre en compte la limitation du flux de charges (intensité) sur la ligne partant du nœud i et allant vers le nœud j. La loi d'Ohm permet d'écrire l'intensité $I_{i,j}$ comme combinaison linéaire des tensions V_i et V_j. Ainsi : $I_{i,j} = aV_i + bV_j$ avec a et b complexes dépendant des équipements sur la ligne (i,j), comme vu dans la partie d'introduction. La contrainte de limitation sur le flux de charges $I_{i,j}$ s'écrit alors :

$$|I_{i,j}| \leq \mathbf{I}_{i,j}^{max} \Leftrightarrow I_{i,j}^* I_{i,j} = |I_{i,j}|^2 \leq (\mathbf{I}_{i,j}^{max})^2. \tag{2.26}$$

Ce que l'on peut écrire :

$$|a|^2 V_i^* V_i + |b|^2 V_j^* V_j + ab^* V_j^* V_i + a^* b V_i^* V_j \leq (\mathbf{I}_{i,j}^{max})^2. \tag{2.27}$$

Cette contrainte s'écrit comme une inégalité quadratique (au sens complexe) selon les variables de tension, comme les autres contraintes de l'OPF. On peut donc la traiter similairement à ce que l'on faisait avant. La matrice M de cette contrainte complexe (telle que $V^* M V \leq (\mathbf{I}_{i,j}^{max})^2$ est la suivante (on n'indique que la sous matrice carrée issue de (i, j)) :

$$\begin{pmatrix} |a|^2 & a^* b \\ ab^* & |b|^2 \end{pmatrix}$$

Cette matrice est hermitienne, sa trace est positive et son déterminant vaut 0. Ses deux valeurs propres sont donc réelles positives. Ce sont aussi les valeurs propres de la hessienne de la contrainte écrite en variables réelles. Cette hessienne est donc une matrice semi-définie positive (SDP).

Ainsi les contraintes de borne sur le module de l'intensité fournissent des contraintes quadratiques convexes à l'OPF dans la modélisation que nous en avons faite. Cependant ces contraintes sont en général très larges et donc non actives. Néanmoins elles peuvent être facilement intégrés à la méthode mise en œuvre dans ce chapitre.

2.3 Comparaison avec la reformulation quadratique convexe de la méthode MIQCR

Dans cette section, on remarque que la relaxation du rang ne fait pas appel aux inégalités de McCormick [24], contrairement à la relaxation SDP utilisée dans la méthode MIQCR [44, 45, 46] pour calculer une reformulation quadratique optimale.

Pour construire la relaxation $(\overline{OPF - L_S})$ de la reformulation $(OPF - L_S)$, on a supprimé la contrainte quadratique non convexe (2.11). Cependant, dans le cas de la méthode MIQCR on relâche différemment la contrainte (2.11). En effet, on le fait en ajoutant les inégalités de McCormick au problème, ce qui permet ensuite d'utiliser un algorithme de *branch-and-bound* spatial.

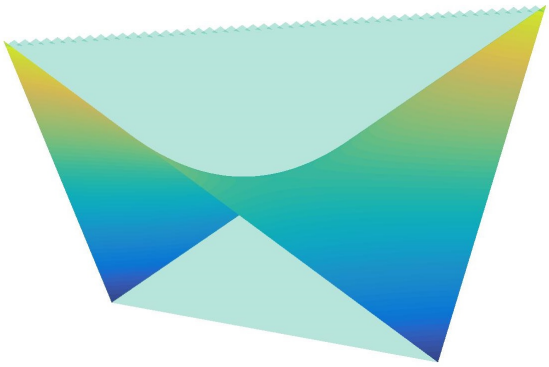

FIGURE 2.1 – Enveloppe convexe définie par les inégalités de McCormick de l'espace non-convexe (2.28) pour des bornes $\ell_i = \ell_j = -1$ et $u_i = u_j = 1$.

Considérons l'espace non convexe suivante pour i et j dans {1,. . . ,2n} :

$$(W_{i,j}, v_i, v_j) \in \mathbb{R}^3 | \begin{cases} W_{i,j} = v_i v_j \\ \ell_i \leq v_i \leq u_i \\ \ell_j \leq v_j \leq u_j \end{cases} \tag{2.28}$$

Où ℓ_i, u_i, ℓ_j et u_j sont les bornes de v_i et v_j. La plus petite enveloppe convexe contenant cet espace est donnée par les quatres inégalités linéaires, dites de McCormick, suivantes :

$$(MCC) \begin{cases} W_{i,j} \leq u_j v_i + \ell_i v_j - \ell_i u_j, \\ W_{i,j} \leq \ell_j v_i + u_i v_j - u_i \ell_j, \\ W_{i,j} \geq u_j v_i + u_i v_j - u_i u_j, \\ W_{i,j} \geq \ell_j v_i + \ell_i v_j - \ell_i \ell_j. \end{cases}$$

En guise d'exemple, considérons que les bornes ℓ_i et ℓ_j valent -1, tandis que les bornes u_i et u_j valent 1. La Figure 2.1 présente la surface correspondant à l'espace non-convexe (2.28). Cette surface est en forme dite de « selle de cheval ». L'enveloppe convexe définie par les inégalités de McCormick est le tétraèdre contenant cette surface.

Dans le cas qui nous intéresse, nos variables sont les variables de tension v. Pour pouvoir appliquer les inégalités de McCormick au problème $(OPF - L_S)$, il faut des bornes sur les variables v. On peut déduire des bornes sur les variables de tension v à l'aide des contraintes (2.5) :

$$\forall i \in \mathcal{N}, \quad -\overline{\mathbf{v}_i} \leq v_i \leq \overline{\mathbf{v}_i} \quad \text{et} \quad -\overline{\mathbf{v}_i} \leq v_{i+n} \leq \overline{\mathbf{v}_i}. \tag{2.29}$$

Nous allons montrer dans la suite qu'ajouter les inégalités de McCormick aux relaxations de

la famille $(\overline{OPF - L_S})$ ne permet pas d'obtenir une meilleure reformulation optimale. En effet, les relaxations de la famille $(\overline{OPF - L_S})$ gardent une valeur optimale inférieure à celle de la relaxation du rang, même après ajout des inégalités de McCormick. La reformulation $(OPF - L_{S^*})$ reste alors une reformulation optimale de $(OPF - L)$, même lorsque les relaxations considérées sont renforcées par les inégalités de McCormick, comme c'est le cas dans la méthode MIQCR.

La relaxation quadratique convexe obtenue en ajoutant les inégalités de McCormick à $(\overline{OPF - L_S})$ est :

$$(\overline{OPF - L_S - MCC}) \begin{cases} \min\limits_{v,W} \quad f_S(v,W) = v^t S v + < \mathbf{C}_1 - S, W > + \mathbf{c}_0 \\ \text{s.t.} \\ \quad < \mathbf{Y}_i^R, W >= -\mathbf{p}_i^c, \quad i \in \mathcal{C} & (2.6) \\ \quad < \mathbf{Y}_i^I, W >= -\mathbf{q}_i^c, \quad i \in \mathcal{C} & (2.7) \\ \quad \underline{\mathbf{p}_i} - \mathbf{p}_i^c \leq < \mathbf{Y}_i^R, W > \leq \overline{\mathbf{p}_i} - \mathbf{p}_i^c, \quad i \in \mathcal{P} & (2.8) \\ \quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq < \mathbf{Y}_i^I, W > \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i \in \mathcal{P} & (2.9) \\ \quad \underline{\mathbf{v}_i}^2 \leq W_{i,i} + W_{i+n,i+n} \leq \overline{\mathbf{v}_i}^2, \quad i \in \mathcal{N} & (2.10) \\ \quad (v_i, v_j, W_{i,j}) \in MCC, \quad (i,j) \in \{1,\ldots,2n\}^2. \end{cases}$$

C'est la relaxation quadratique convexe sur laquelle s'appuie notre résolution dans la suite de ce manuscrit.

La proposition suivante fait parti des contributions originales majeures de ce manuscrit.

Proposition 2. *$val(SDP - L) \geq (\overline{OPF - L_S - MCC})$ pour toute matrice S SDP.*

Preuve. Soit S une matrice SDP, soit $(W^* \succeq 0)$ une solution optimale de $(SDP - L)$. En réutilisant ce qui a été fait précédemment, nous savons que $(W = W^*, v = 0)$ est une solution de $(\overline{OPF - L_S})$ ayant une valeur inférieure à la relaxation du rang. Il ne reste plus qu'à montrer que $(W = W^*, v = 0)$ vérifie les inégalités de McCormick pour pouvoir conclure.

Comme $W^* \succeq 0$, on a : $\forall i \in \{1,\ldots,2n\}, W^*_{i,i} \geq 0$.
Et de plus, d'après (2.10) : $\forall i \in \{1,\ldots,2n\}, W^*_{i,i} \leq \overline{\mathbf{v}_i}^2$ Ici, on a étendu sur $\{n+1,\ldots,2n\}$ la définition de $\overline{\mathbf{v}}$, en considérant que pour $i \in N, \overline{\mathbf{v}}_i = \overline{\mathbf{v}_{i+n}}$. Comme $v = 0$, les quatres inégalités de McCormick se

résument aux deux inégalités suivantes :

$$\begin{cases} W^*_{i,j} \leq \overline{\mathbf{v}_i \mathbf{v}_j} \\ W^*_{i,j} \geq -\overline{\mathbf{v}_i \mathbf{v}_j} \end{cases} \tag{2.30}$$

Or, comme $W^* \succeq 0$, le déterminant du mineur d'odre 2 de W^* relatif aux lignes et colonnes i et j est positif, ce qui implique que ${W^*_{i,j}}^2 \leq W^*_{i,i} W^*_{j,j}$. De plus, comme montré plus haut $W^*_{i,i} \leq \overline{\mathbf{v}_i}^2$ et $W^*_{j,j} \leq \overline{\mathbf{v}_j}^2$, donc : ${W^*_{i,j}}^2 \leq \overline{\mathbf{v}_i}^2 \overline{\mathbf{v}_j}^2$ on en déduit que les inégalités de McCormick sont vérifiées. □

La proposition ci-dessus nous assure que la relaxation du rang $(SDP - L)$ fournit, grâce à la matrice duale optimale S^*, une reformulation optimale, même dans le cas où l'on considère les inégalités de McCormick dans la relaxation résolue au nœud racine du *branch-and-bound* spatial.

La différence majeure avec MIQCR [46] réside dans le fait que la relaxation SDP à résoudre pour construire une reformulation optimale, n'a pas besoin de contenir, elle, les inégalités de McCormick. Ce qui fait qu'elle est plus creuse et plus rapide à résoudre. De plus, elle correspond alors à la relaxation SDP de l'état de l'art pour l'OPF [9]. Nous insisitons sur le fait que l'on retire un nombre de contraintes qui est quadratique en le nombre des variables en passant de la relaxation SDP de MIQCR à la relaxation du rang. Si on ne le faisait pas, il serait trop couteux en temps de calcul de résoudre une relaxation SDP pour une instance représentant un réseau électrique d'une centaine de nœuds.

Reformulation quadratique convexe des contraintes

Une autre idée qui peut être envisagée est de travailler avec une famille plus étendue de formulations. Une extension à notre famille de reformulations serait de considérer que pour chaque contrainte d'inégalité k du problème on ajoute un terme correctif à l'aide d'une matrice semi-définie positive S_k, comme on le fait pour la fonction objectif. Ceci signifie, que pour une contrainte d'inégalité k de $(OPF - L)$ écrite de manière générique sous la forme : $v^t A_k v \leq b_k$, on la reformule de manière équivalente en : $v^t S_k v + < A_k - S_k, W > \leq b_k$.

Les relaxations issues de ces reformulations ont aussi une valeur inférieure à celle de la relaxation du rang, cela pour la même raison que précédemment. En effet, on a montré que pour toute solution réalisable W^* de $(SDP - L)$ on pouvait construire une solution réalisable de $(\overline{OPF - L_S})$, $(0, W^*)$, dont la fonction objectif prend une valeur inférieure. Or $(0, W^*)$ restera réalisable après cette transformation

des contraintes, car la valeur du membre de gauche de la contrainte k reformulée sera inférieure dans la relaxation quadratique convexe par rapport à son équivalent dans la relaxation du rang. En effet :

$$< A_k, W^* > - \quad 0 \quad - < A_k - S_k, W^* > \quad = \quad < S_k, W^* > \quad \geq \quad 0.$$

La contrainte d'infériorité sera alors vérifiée.

Notre méthode fournit donc la reformulation optimale au sein de la famille de reformulations quadratiques qui vient « convexifier » de manière quadratique chaque contrainte du problème de départ. On retrouve alors le résultat de MIQCR mais où seule la résolution d'un problème SDP plus petit, la relaxation dite du rang, est nécessaire.

2.4 Phase 2 : Algorithme de *branch-and-bound* spatial

Nous avons construit une reformulation quadratique de $(OPF - L)$ appelée $(OPF - L_{S^*})$, où S^* s'obtient à l'aide des variables duales optimales de la relaxation du rang de $(OPF - L)$. Nous allons maintenant résoudre cette reformulation à l'optimum global à l'aide d'un algorithme de *branch-and-bound* spatial. Cette section commence par décrire la structure générale de cet algorithme, puis explicite la manière de choisir les variables de branchement ainsi que les valeurs en lesquelles effectuer ce branchement. Elle se termine par le choix effectué quant au nœud de l'arbre à traiter parmi ceux non encore traités au cours de l'algorithme de *branch-and-bound*.

2.4.1 Présentation générale de l'algorithme

Notre algorithme suit d'abord une structure classique pour un algorithme de *branch-and-bound*. Les variables de branchement sont celles de tension v. Il construit et parcourt un arbre comme décrit dans la partie relative à l'état de l'art. Les nœuds de l'arbre sont caractérisés par les bornes ℓ et u considérées en chaque nœud sur les variables v. Les inégalités de McCormick varient donc à chaque nœud avec les bornes sur les variables. La Figure 2.2 décrit les actions entreprises en chaque nœud de notre algorithme.

La relaxation quadratique et convexe résolue en chaque nœud de l'arbre est $(\overline{OPF - L_{S^*} - MCC})$ Il faut noter que dans notre implémentation, on ne crée une variable $W_{i,j}$ que

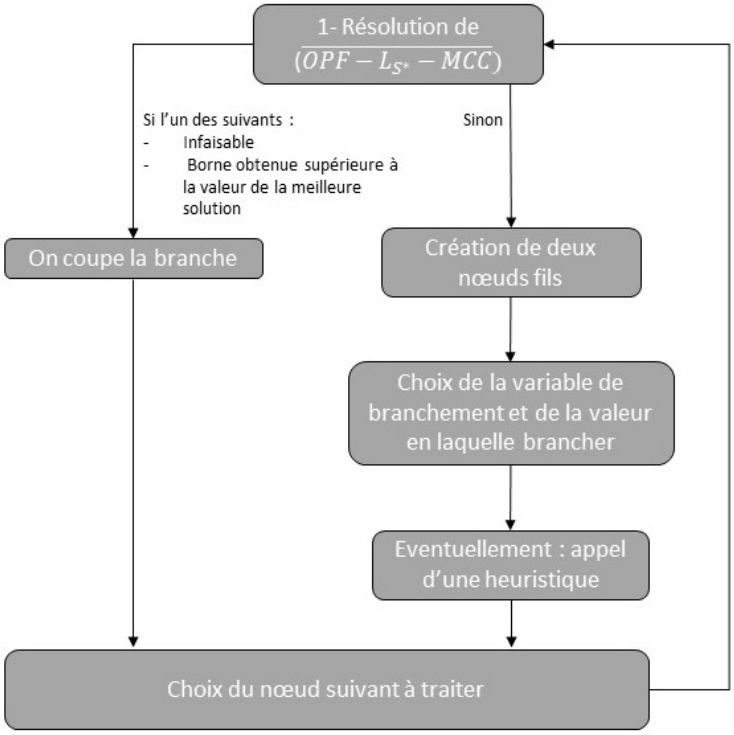

FIGURE 2.2 – Schéma des actions effectuées en chaque nœud de l'arbre de l'algorithme

si cette dernière se trouve associée à un coefficient non nul dans la fonction objectif ou dans une des contraites de $(OPF - L_{S^*})$. De même, on ne génère que les inégalités de McCormick relatives aux variables $W_{i,j}$ créées.

L'algorithme débute en construisant un nœud racine. À la racine, comme décrit dans la Section 2.3, les bornes ℓ et u sont données par : $\forall i, \quad \ell_i = -\overline{\mathbf{v}_i}, \quad u_i = \overline{\mathbf{v}_i}$. Même si ces bornes sont correctes, on peut cependant quelque peu les réduire immédiatement, et ce point est très important en ce qui concerne les algorithmes de type *branch-and-bound* spatial. En effet, pour l'OPF, on cherche la valeur complexe des tensions en chaque nœud du réseau, cependant le problème est doté d'une symétrie particulière. Ce dernier est invariant par rotation de l'ensemble des tensions dans le plan complexe. L'interprétation physique de cela est que les phases des tensions sont arbitraires, seules les différences de phase ont une importance. Ainsi, on peut choisir une phase de référence, c'est-à-dire fixer à zéro la phase de la tension en un nœud, que l'on note r. En coordonnées cartésiennes, cela nous permet de dire que $Im(V_r) = 0$ et $Re(V_r) \geq 0$. Combiné avec les informations déjà à disposition, on en déduit les bornes suivantes pour v_r et v_{r+n} :

$$\ell_r = \underline{\mathbf{v}_r}; u_r = \overline{\mathbf{v}_r} \quad ; \quad \ell_{r+n} = u_{r+n} = 0. \tag{2.31}$$

En faisait cette brisure de symétrie, on réduit le nombre de solutions optimales équivalentes pour $(OPF - L)$, ce qui simplifie la résolution du problème avec un algorithme de *branch-and-bound*.

En ce qui concerne l'heuristique utilisée pour trouver des bornes supérieures sur la valeur optimale du problème, on résout tout simplement le problème de départ $(OPF - L)$ avec un algorithme de points intérieurs. Ceci nous fournit une solution réalisable mais ne nous garantit pas son optimalité.

2.4.2 Choix de la variable de branchement

Ce qui rend la relaxation résolue en chaque nœud non exacte, c'est la relaxation de la contrainte $y_{i,j} = x_i x_j$ en les inégalités de McCormick. On cherche alors à orienter le choix du branchement de manière à venir valider cette contrainte.

Au cours de nos travaux, plusieurs méthodes ont été envisagées, implémentées et évaluées pour atteindre ce résultat, on en présente deux ci-dessous. Toutes ces méthodes partent de la solution

optimale renvoyée par la relaxation quadratique convexe calculée au nœud courant, on note $(\hat{v}, \hat{W})$ cette solution.

Branchement « erreur bilinéaire »

L'idée derrière ce branchement est de chercher le couple de variables $(i, j), i \leq j$ pour lequel l'égalité $\hat{W}_{i,j} = \hat{v}_i \hat{v}_j$ est la plus violée. C'est-à-dire :

$$(i, j) = \text{argmax}_{k,l} |\hat{W}_{k,l} - \hat{v}_k \hat{v}_l|.$$

On sait que lorsque la relaxation est exacte, cette égalité n'est pas violée, et de plus la réciproque est vraie. Le choix de cette méthode de branchement nous assure alors la convergence de notre algorithme de résolution à l'optimum global. Pour sélectionner v_i ou v_j comme variable de branchement, on sélectionne la variable dont le domaine $[\ell, u]$ est le plus grand au nœud courant.

Une variante peut être de donner plus d'importance aux termes diagonaux. En reprenant l'idée précédente, on cherche la variable i pour laquelle l'égalité $\hat{W}_{i,i} = \hat{v}_i^2$ est la plus violée. c'est-à-dire :

$$i = \text{argmax}_k |\hat{W}_{k,k} - \hat{v}_k^2|.$$

Là encore, quand la relaxation est exacte cette égalité n'est pas violée. Si par contre l'égalité est vérifiée pour tout i, on reprend l'idée de départ. Cette variante de branchement revient à donner la priorité aux termes diagonaux dans la violation de $\hat{W}_{i,j} = \hat{v}_i \hat{v}_j$. L'intérêt de cette méthode réside aussi dans le fait que le calcul de la variable sur laquelle brancher est plus rapide si l'on trouve un terme diagonal pour lequel l'égalité est violée.

Branchement « erreur bilinéaire pondérée »

Cette méthode est une modification de la précédente, où l'on cherche à pondérer la violation de l'égalité $\hat{W}_{i,j} = \hat{v}_i \hat{v}_j$ par l'impact que cette violation va avoir sur la différence entre la fonction objectif de la relaxation et celle du problème d'origine. On cherche alors le couple (i, j) maximisant la quantité :

$$|S^*_{i,j} - \mathbf{C}_{1i,j}| \cdot |\hat{W}_{i,j} - \hat{v}_i \hat{v}_j|.$$

C'est cette méthode qui est utilisée dans l'algorithme de MIQCR. On peut aussi chercher à donner la priorité aux termes diagonaux.

2.4.3 En quelle valeur couper ?

Une fois que l'on a choisi une variable v_c dont on va couper en deux le domaine, il faut choisir en quelle valeur cela sera fait. Dans ce travail, on choisit de suivre les recommandations données dans [1] et utilisées par MIQCR [46]. Ces dernières indiquent de couper quelque part entre le centre du domaine actuel de la variable et la valeur prise par cette variable dans la solution de la relaxation $\hat{v}$. Ainsi, on définit un paramètre $\delta \in [0, 1]$, et on choisit de couper en :

$$v_c^{cut} = \delta \frac{\ell_c + u_c}{2} + (1 - \delta)\hat{v}_c.$$

Couper au centre du domaine ($\delta = 1$) a l'avantage de diminuer le nombre de nœuds à générer pour faire diminuer la taille globale de tous les domaines. Couper en la valeur de la relaxation ($\delta = 0$) a l'avantage de faire disparaitre la solution de la relaxation dans les espaces réalisables des deux nœuds fils. En effet, si une variable i atteint une de ses bornes (c'est-à-dire si $v_i = l_i$ ou $v_i = u_i$), les inégalités de McCormick impliquent que pour tout j : $W_{i,j} = v_i v_j$, pour une solution réalisable. C'est d'ailleurs cette propriété fondamentale des enveloppes de McCormick qui assurent la convergence de l'algorithme.

Ne pas couper en la valeur de la relaxation peut sembler risqué car cela n'assure pas que les nœuds fils donnent une solution optimale différente du père. Mais cela semble rentable expérimentalement, surtout quand la solution de la relaxation donne une valeur proche des bornes de la variables.

2.4.4 Choix du nœud suivant

Le choix du nœud suivant à explorer dans l'algorithme de *branch-and-bound* se fait en général en suivant l'une ou l'autre des 2 méthodes suivantes : soit on cherche à aller prioritairement en profondeur, c'est-à-dire à suivre les branches les plus longues de l'arbre, soit on cherche à aller en largeur, c'est-à-dire à ouvrir le plus de branches possibles, en traitant en priorité le nœud qui a la meilleure borne inférieure. La première méthode est souvent privilégiée quand il est difficile d'obtenir de bonnes bornes supérieures sur le problème.

Dans le cadre de l'algorithme implémenté pour résoudre $(OPF - L)$, nous avons utilisé un dispositif « meilleur d'abord » qui puise plutôt sa philosophie dans un parcours en largeur. Ce choix est fait car il est facile d'obtenir de très bonnes bornes supérieures pour l'OPF [22]. L'enjeu réside plutôt dans la

remontée de la borne inférieure. Ainsi, on attache à chaque nœud fils créé un potentiel qui correspond à la borne inférieure obtenue à son nœud parent. Cette borne est valide pour le nœud fils. Quand vient l'heure de choisir un nouveau nœud, on prend celui ayant le potentiel le plus bas. Cette méthode vise donc à faire remonter le plus vite possible la borne inférieure globale de l'arbre, qui est égale au minimum des potentiels des nœuds inexplorés.

2.4.5 Autres paramètres

Il nous reste à éclairer deux choix en plus de ceux relatifs aux branchements. Tout d'abord le choix du nœud du réseau électrique ayant le potentiel de référence. Pour ce nœud, la partie imaginaire de la tension est nulle, comme expliqué dans (2.4.1). Mettre à zéro une variable permet de supprimer plusieurs termes dans les contraintes du problème. Il est donc tentant de choisir la variable apparaissant le plus souvent. Cela revient à choisir de fixer la tension de référence dans le nœud ayant le plus de voisins dans le réseau électrique. De plus, en ce qui concerne l'heuristique utilisée pour trouver des bornes supérieures, on la renforce légèrement en lui donnant comme point de départ la solution courante de la relaxation. Cela nous permet de trouver des solutions différentes à chaque appel, et de guider le logiciel utilisé dans les zones qui ont le plus de potentiel à contenir de meilleures solutions réalisables.

2.4.6 Notre méthode de résolution : `RC-OPF`

Cette section donne les éléments nous permettant de résoudre une instance du problème $(OPF - L)$ a l'optimum global. Tout d'abord nous contruisons la reformulation quadratique optimale décrite dans la Section 2.2 en résolvant la relaxation du rang. Cette reformulation est ensuite résolue à l'optimum global à l'aide de l'algorithme de *branch-and-bound* de la Section 2.4. Notre reformulation quadratique optimale nous assure que la borne inférieure obtenue au nœud racine de notre arbre de *branch-and-bound* est maximale. Cette borne vaut celle de la relaxation du rang. L'algorithme **??** permet de résoudre de manière exacte une instance $(OPF - L)$, c'est l'algorithme que nous avons implémenté.

Algorithme 1 : Méthode `RC-OPF`

1 `Phase 0` : Transformer le problème quadratique à variables complexes en un problème quadratique à variables réelles;
2 `Phase 1` :
3 Résoudre la relaxation du rang $(SDP - L)$: $LB \leftarrow val(SDP - L)$;
4 À l'aide de la solution duale optimale de $(SDP - L)$, construire la reformulation $(OPF - L_{S^*})$;
5 `Phase 2` :
6 Déterminer le nœud électrique de référence qui est un nœud électrique ayant un maximum de voisins dans le réseau ainsi que les bornes supérieures et inférieures sur les variables v comme dans (2.29) et (2.31);
7 Lancer une heuristique de résolution de $(OPF - L)$ ayant $V_i = 1, \forall i \in \mathcal{N}$ comme point de départ : $UB \leftarrow val(Heuristique)$;
8 **tant que** $\frac{UB-LB}{UB} \geq \epsilon$ **faire**
9 Choisir la feuille non coupée de l'arbre avec le potentiel le plus élevé;
10 Appeler une méthode de contraction de bornes (voir chapitre 3);
11 Résoudre la relaxation $(\overline{OPF - L_{S^*} - MCC})$ avec les bornes sur v du nœud courant;
12 **si** *la relaxation est infaisable* **alors**
13 Couper le nœud;
14 **sinon**
15 **si** $val(\overline{OPF - L_{S^*} - MCC}) \geq UB$ **alors**
16 Couper le nœud;
17 **sinon**
18 Déterminer la variable de branchement v_c avec la méthode « erreur bilinéaire pondérée », sans donner la priorité aux termes diagonaux;
19 Déterminer la valeur de branchement v_c^{cut}. Ici on choisit $\delta = 0.5$. $v_c^{cut} \leftarrow \delta \frac{\ell_c + u_c}{2} + (1 - \delta)\hat{v}_c$;
20 Construire deux nouveaux nœuds comme fils du nœud courant avec comme potentiel la valeur de la relaxation du nœud courant. Les bornes sur v de chaque fils sont celles du nœud courant, sauf pour v_c où l'un des fils remplace la borne inférieure par v_c^{cut}, et l'autre sa borne supérieure par v_c^{cut};
21 Lancer l'heuristique de résolution de $(OPF - L)$ avec comme point de départ la solution en v de la relaxation;
22 **si** $val(Heuristique) < UB$ **alors**
23 $UB \leftarrow val(Heuristique)$;
24 Remplacer la valeur de LB par celle du plus petit potentiel parmi les feuilles de l'arbre non coupées.

2.5 Illustration de RC-OPF sur l'instance WB2

2.5.1 Présentation de l'instance

WB2 [2] est une instance du réseau électrique à deux noeuds, le noeud 1 est muni d'un générateur et n'exprime aucune demande électrique. L'objectif est alors d'assouvir la demande du noeud 2 (purement consommateur) à l'aide de la production du noeud 1 en passant par l'unique arête du réseau reliant ces deux noeuds.

Les variables sur lesquelles on souhaite optimiser sont les suivantes :

— La tension V_i complexe en chaque noeud i du réseau.

— La puissance générée au noeud 1 : S_1.

— I_i le courant complexe injecté en chaque noeud i du réseau.

— $I_{1,2}$ le courant complexe circulant du noeud 1 vers le noeud 2 dans l'arête du réseau.

On dispose des données suivantes :

— La puissance demandée au noeud 2, de partie réelle $\mathbf{p}_2^c$ et de partie imaginaire $\mathbf{q}_2^c$.

— Les bornes sur le module de la tension en chaque noeud : $\underline{\mathbf{v}_1}$, $\overline{\mathbf{v}_i}$, $\underline{\mathbf{v}_2}$ et $\overline{\mathbf{v}_2}$.

— Les bornes sur la partie réelle de la puissance générée au générateur $\underline{\mathbf{p}_1}$ et $\overline{\mathbf{p}_1}$.

— Les bornes sur la partie imaginaire de la puissance générée au générateur $\underline{\mathbf{q}_1}$ et $\overline{\mathbf{q}_1}$.

— L'admittance $\mathbf{Y}_{1,2}$ de l'arête reliant les deux noeuds du réseau.

— Le coefficient $\mathbf{c}_{1,1}$ de coût linéaire en la puissance active au générateur.

Le tableau suivant recense toutes ces valeurs, qui sont exprimées sans unités car ses grandeurs ont été normalisées.

$\mathbf{p}_2^c$	350
$\mathbf{q}_2^c$	-350
$\underline{\mathbf{v}_1}$	0.95
$\overline{\mathbf{v}_1}$	1.05
$\underline{\mathbf{v}_2}$	0.95
$\overline{\mathbf{v}_2}$	1.028
$\underline{\mathbf{p}_1}$	0
$\overline{\mathbf{p}_1}$	600
$\underline{\mathbf{q}_1}$	-400
$\overline{\mathbf{q}_1}$	400
$\mathbf{Y}_{1,2}$	$96.15 - 480.77\mathbf{i}$
$\mathbf{c}_{1,1}$	2

Le problème physique vise à minimiser la quantité de puissance réelle produite suivante :

$$\mathbf{c}_{1,1} Re(S_1). \tag{2.32}$$

Sachant que cette même puissance est bornée ainsi :

$$\underline{\mathbf{p}_1} \leq Re(S_1) \leq \overline{\mathbf{p}_1}. \tag{2.33}$$

La principale contrainte du problème est celle relative à la puissance injectée. La puissance injectée au noeud i vaut $V_i\overline{I_i}$ et elle est égale à la puissance produite en ce noeud à laquelle on soustrait la puissance demandée en ce même noeud.

$$V_1\overline{I_1} = S_1. \tag{2.34}$$

$$V_2\overline{I_2} = -\mathbf{p}_2^c - j\mathbf{q}_2^c. \tag{2.35}$$

La puissance imaginaire est elle aussi bornée :

$$\underline{\mathbf{q}_1} \leq Re(S_1) \leq \overline{\mathbf{q}_1}. \tag{2.36}$$

Les tensions en chaque noeud sont aussi bornées en module :

$$\underline{\mathbf{v}_1} \leq |V_1| \leq \overline{\mathbf{v}_1}. \tag{2.37}$$

$$\underline{\mathbf{v}_2} \leq |V_2| \leq \overline{\mathbf{v}_2}. \tag{2.38}$$

La loi des noeuds relie les intensités injectées en chaque noeud et les courants régnant dans chaque ligne incidantes à ce noeud, ici cela se traduit par les deux égalités simples suivantes :

$$I_1 = I_{1,2}. \tag{2.39}$$

$$I_2 = -I_{1,2}. \tag{2.40}$$

Enfin la loi d'Ohm relie les tensions aux courants de la façon suivante :

$$I_{1,2} = \mathbf{Y}_{1,2}(V_1 - V_2). \tag{2.41}$$

2.5.2 Substitution de variables

Dans cette section on illustre l'écriture du problème à l'aide des seules variables de tension. On voit dans un premier temps avec (2.41) que le courant dans la ligne s'exprime en fonction des tensions aux extrémités de la ligne. De plus les courants injectés en chaque noeud s'expriment en fonction du courant traversant la ligne reliant ces deux nœuds (2.39) et (2.40). On en conclut que tous les courants se déduisent des tensions et que l'on peut construire une matrice complexe $\mathbf{Y}$ telle que $I = \mathbf{Y}V$. $\mathbf{Y}$ s'appelle la matrice d'admittance du réseau. On obtient la matrice suivante [1] :

$$\mathbf{Y} = \begin{bmatrix} 96.15 - 480.8\mathbf{i} & -96.15 + 480.8\mathbf{i} \\ -96.15 + 480.8\mathbf{i} & 96.15 - 480.8\mathbf{i} \end{bmatrix}$$

Maintenant il faut substituer la variable de puissance S_1 pour se ramener aux seules variables V de tension. Grâce à la matrice d'admittance on peut exprimer les courants en fonction des tensions, ainsi (2.34) permet d'exprimer S_1 uniquement à l'aide des tensions. On peut donc écrire le problème en fonction des seules variables de tension en procédant ainsi : A partir de $I = \mathbf{Y}V$, on en déduit que la puissance injectée au noeud i, vérifie : $V_i I_i^* = V^* \mathbf{Y}_i^M V$ où $\mathbf{Y}_i^M$ est la matrice définie en Section (0.2.5). Puis :

$$Re(S_1) = V^* \frac{\mathbf{Y}_1^M + \mathbf{Y}_1^{M^*}}{2} V$$

$$Im(S_1) = V^* \frac{\mathbf{Y}_1^M - \mathbf{Y}_1^{M^*}}{2\mathbf{i}} V$$

2.5.3 Formulation du problème en variables complexes

Le problème en variables complexes prend donc la forme standard suivante :

$$\begin{array}{llllll} \min\limits_{V} & V^*CV & & & & \\ \text{s.t.} & b_k^l & \leq & V^*B_kV & \leq & b_k^u, \quad k \in \{1,2,3,4\} \\ & & & V^*A_kV & = & a_k, \quad k \in \{1,2\} \end{array}$$

L'objectif provient de (2.32), ainsi :

$$C = \mathbf{c}_{1,1} \frac{\mathbf{Y}_1^M + \mathbf{Y}_1^{M^*}}{2} = \begin{bmatrix} 192.3 & -96.15 + 480.8\mathbf{i} \\ -96.15 - 480.8\mathbf{i} & 0 \end{bmatrix}.$$

1. La matrice a été renormalisée, tous ces termes ont été multipliés par 100 qui représente le coefficient de renormalisation de ce réseau.

Les contraintes d'inégalité proviennent de (2.33), (2.36), (2.37) et (2.38). (2.33) fournit :

$$B_1 = \frac{\mathbf{Y}_1^M + \mathbf{Y}_1^{M^*}}{2} = \begin{bmatrix} 96.15 & -48.075 + 240.4\mathbf{i} \\ -48.075 - 240.4\mathbf{i} & 0 \end{bmatrix}.$$

$$b_1^l = 0.$$

$$b_1^u = 600.$$

(2.36) fournit :

$$B_2 = \frac{\mathbf{Y}_1^M - \mathbf{Y}_1^{M^*}}{2\mathbf{i}} = \begin{bmatrix} 480.8 & -240.4 - 48.075\mathbf{i} \\ -240.4 + 48.075\mathbf{i} & 0 \end{bmatrix}.$$

$$b_2^l = -400.$$

$$b_2^u = 400.$$

(2.37) fournit :

$$B_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

$$b_3^l = 0.9025.$$

$$b_3^u = 1.1025.$$

(2.38) fournit :

$$B_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$b_4^l = 0.9025.$$

$$b_4^u = 1.0568.$$

La contrainte d'égalité provient de (2.35), elle fournit :

$$A = \mathbf{Y}_2^M = \begin{bmatrix} 0 & -96.15 - 480.8\mathbf{i} \\ 0 & 96.15 + 480.8\mathbf{i} \end{bmatrix}.$$

$$a = -350 + 350\mathbf{i}.$$

On remarque que l'objectif ainsi que les contraintes sont des fonctions quadratiques des variables. Les matrices mises en jeu sont hermitiennes dans le cas de l'objectif et des contraintes d'inégalités, on

est tenté de faire de même pour la contrainte d'égalité, cela revient à la scinder en deux contraintes portant l'une sur l'égalité des parties réelles et l'autre sur l'égalité des parties imaginaires. On obtient :

$$A_1 = \frac{\mathbf{Y}_2^M + \mathbf{Y}_2^{M^*}}{2} = \begin{bmatrix} 0 & -48.075 - 240.4\mathbf{i} \\ -48.075 + 240.4\mathbf{i} & 96.15 \end{bmatrix}$$

$$a_1 = -350$$

$$A_2 = \frac{\mathbf{Y}_2^M - \mathbf{Y}_2^{M^*}}{2\mathbf{i}} = \begin{bmatrix} 0 & -240.4 + 48.075\mathbf{i} \\ -240.4 - 48.075\mathbf{i} & 480.8 \end{bmatrix}$$

$$a_2 = 350$$

2.5.4 Phase 0 : Réécriture en variables réelles

Notre problème est maintenant exprimé dans un formalisme standard d'optimisation, cependant la méthode que nous allons utiliser pour le résoudre nécessite l'emploi de nombres réels. Pour réaliser ce passage nous doublons le nombre de variables, au lieu d'avoir comme variables la tension en chaque noeud, on a maintenant la partie réelle et la partie imaginaire de la tension en chaque noeud. Le nouveau vecteur de variables v, est constitué ainsi : v_1 est la partie réelle de la tension au noeud, v_2 celle au noeud 2, v_3 la partie imaginaire de la tension au noeud 1, v_4 celle au noeud 2. En ce qui concerne les matrices des contraintes, il faut doubler leur dimension, ainsi par abus de notation une matrice M va devenir :

$$M = \begin{bmatrix} Re(M) & -Im(M) \\ Im(M) & Re(M) \end{bmatrix}$$

Le problème garde la forme standard précédente, il n'y a que l'expression des matrices et la taille du vecteur de variables qui sont modifiées. On donne à la suite les expressions complètes des matrices du problème.

$$C = \begin{bmatrix} 192.3 & -96.15 & 0 & -480.8 \\ -96.15 & 0 & 480.8 & 0 \\ 0 & 480.8 & 192.3 & -96.15 \\ -480.8 & 0 & -96.15 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 96.15 & -48.075 & 0 & -240.4 \\ -48.075 & 0 & 240.4 & 0 \\ 0 & 240.4 & 96.15 & -48.075 \\ -240.4 & 0 & -48.075 & 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 480.8 & -240.4 & 0 & 48.075 \\ -240.4 & 0 & -48.075 & 0 \\ 0 & -48.075 & 480.8 & -240.4 \\ 48.075 & 0 & -240.4 & 0 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0 & -48.075 & 0 & 240.4 \\ -48.075 & 96.15 & -240.4 & 0 \\ 0 & -240.4 & 0 & -48.075 \\ 240.4 & 0 & -48.075 & 96.15 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & -240.4 & 0 & -48.075 \\ -240.4 & 480.8 & 48.075 & 0 \\ 0 & 48.075 & 0 & -240.4 \\ -48.075 & 0 & -240.4 & 480.8 \end{bmatrix}$$

2.5.5 Phase 1.1 : Résolution de la relaxation du rang

On considère la relaxation du rang :

$$\begin{array}{lllllll} \min\limits_{W} & < C, W > & & & & & \\ \text{s.t.} & b_k^l & \leq & < B_k, W > & \leq & b_k^u & k \in \{1,2,3,4\} \\ & & & < A_k, W > & = & a_k & k \in \{1,2\} \\ & & & W & \succeq & 0 & \end{array}$$

On rappelle qu'il y a dualité forte pour ce problème. Le problème dual associé est :

$$\begin{array}{lll} \max\limits_{\alpha,\underline{\beta}\overline{\beta}} & - < \alpha, a > + < \underline{\beta}, b^l > - < \overline{\beta}, b^u > & \\ \text{s.t.} & C + \sum\limits_{k\in\{1,2\}} \alpha_k A_k + \sum\limits_{k\in\{1,2,3,4\}} (\overline{\beta_k} - \underline{\beta_k}) B_k & \succeq 0 \\ & \overline{\beta}, \underline{\beta} & \geq 0 \end{array}$$

On résout le problème dual à l'aide du solveur de problèmes SDP Sedumi. La valeur de l'objectif donne une borne inférieure au problème de l'OPF tandis que la matrice SDP des contraintes, notée S^*, va nous servir à concevoir une relaxation quadratique convexe de l'OPF.

$$obj = 885.715$$

$$S^* = 1e^{-3} \times \begin{bmatrix} 0.2711 & -0.1280 & 0 & 0.2160 \\ -0.1280 & 0.2325 & -0.2160 & 0 \\ 0 & -0.2160 & 0.2711 & -0.1280 \\ 0.2160 & 0 & -0.1280 & 0.2325 \end{bmatrix}$$

2.5.6 Phase 1.2 : Construction de la reformulation quadratique optimale

À partir des élèments précédents on peut construire la reformulation quadratique optimale :

$$\begin{array}{lllllll} \max\limits_{v,W} & v^t S^* v + < C - S^*, W > & & & & & \\ \text{s.t.} & b_k^l & \leq & < B_k, W > & \leq & b_k^u, & k \in \{1,2,3,4\} \\ & & & < A_k, W > & = & a_k, & k \in \{1,2\} \end{array}$$

2.5.7 Phase 2 : Résolution à l'aide de l'algorithme de *branch-and-bound*

Notre réseau électrique ne contient que deux nœuds reliés entre eux. Chacun dispose alors d'un seul voisin. On choisit arbitrairement le nœud 1 comme nœud de référence.

L'heuristique de résolution obtient une solution réalisable dont l'objectif vaut 905.728. La borne à la racine de l'arbre de *branch-and-bound* spatial vaut 885.715.

Notre algorithme traite 31 nœuds de l'arbre avant de converger. Les bornes finales sont :

$$LB = 905.723,$$

$$UB = 905.723.$$

La valeur de la solution optimale obtenue est :

$$v_1 = 0.95,$$

$$v_2 = 0.413411,$$

$$v_3 = 0,$$

$$v_4 = -0.884211\mathbf{i}.$$

2.6 Extension de la méthode de reformulation quadratique optimale au problème $(OPF - Q)$ et algorithme RC-OPF-Q

Cette section s'attache à étendre ce qui a été fait dans la Section 2.2 à la variante $(OPF - Q)$ cette fois-ci. Nous mettons en place les éléments théoriques permettant de généraliser l'approche et l'algorithme RC-OPF au problème $(OPF - Q)$. En appliquant la même méthode pour passer des variables complexes aux variables réelles, on obtient la formulation suivante en coordonnées cartésiennes de $(OPF - Q)$:

$$(OPF - Q)\begin{cases} \min\limits_{v,p} \quad f^q(v,p) = \mathbf{c}_1^t p + p^t \mathbf{C}_2 p + \mathbf{c}_0^q \\ \text{s.t.} \\ \quad \underline{\mathbf{p}_i} \leq p_i \leq \overline{\mathbf{p}_i}, \quad \forall i \in \mathcal{P}, & (8) \\ \quad v^t \mathbf{Y}_i^R v = -\mathbf{p}_i^c, \quad i \in \mathcal{C} & (2.1) \\ \quad v^t \mathbf{Y}_i^I v = -\mathbf{q}_i^c, \quad i \in \mathcal{C} & (2.2) \\ \quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq v^t \mathbf{Y}_i^I v \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i \in \mathcal{P} & (2.4) \\ \quad \underline{\mathbf{v}_i}^2 \leq v_i^2 + v_{i+n}^2 \leq \overline{\mathbf{v}_i}^2, \quad i \in \mathcal{N} & (2.5) \\ \quad v^t \mathbf{Y}_i^R v = p_i - \mathbf{p}_i^c, \quad i \in \mathcal{P} & (2.42) \end{cases}$$

Où $\mathbf{c}_0^q = \sum_{i \in \mathcal{P}} \mathbf{c}_{0i}$. $\mathbf{c}_1$ est le vecteur des $(\mathbf{c}_{1,i})_{i \in \mathcal{P}}$. $\mathbf{C}_2$ est la matrice diagonale de $\mathcal{M}_{n_p}(\mathbb{C})$ contenant en position i sur la diagonale le coefficient $\mathbf{C}_{2,i}$. On peut observer que la fonction objectif est quadratique convexe.

2.6.1 Reformulation quadratique

Soit $T \in \mathbb{S}_{2n+n_p}(\mathbb{R})$ une matrice semi-définie positive. On pose la matrice de variables $Z \in \mathbb{S}_{2n+n_p}(\mathbb{R})$. Pour simplifier les notations dans les problèmes qui suivent, on fera référence à certaines sous-matrices de Z par des noms spécifiques. Voici comment on décompose la matrice Z :

$$Z = \begin{bmatrix} W & \star \\ \star & N \end{bmatrix}. \tag{2.43}$$

Où W est de taille $2n \times 2n$ et N de taille $n_p \times n_p$.

Le problème $(OPF - Q_T)$ est alors une reformulation exacte de $(OPF - Q)$.

$$
(OPF - Q_T) \begin{cases}
\min\limits_{Z,v,p} \quad f_T^q(Z,v,p) = \mathbf{c}_1^t p + < \mathbf{C}_2, N > + \begin{bmatrix} v \\ p \end{bmatrix}^t T \begin{bmatrix} v \\ p \end{bmatrix} \\
\quad - < T, Z > + \mathbf{c}_0^q \\
\text{s.t.} \\
\quad \underline{\mathbf{p}_i} \leq p_i \leq \overline{\mathbf{p}_i}, \quad \forall i \in \mathcal{P} & (8) \\
\quad < \mathbf{Y}_i^R, W >= -\mathbf{p}_i^c, \quad i \in \mathcal{C} & (2.6) \\
\quad < \mathbf{Y}_i^I, W >= -\mathbf{q}_i^c, \quad i \in \mathcal{C} & (2.7) \\
\quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq < \mathbf{Y}_i^I, W > \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i \in \mathcal{P} & (2.9) \\
\quad \underline{\mathbf{v}_i}^2 \leq W_{i,i} + W_{i+n,i+n} \leq \overline{\mathbf{v}_i}^2, \quad i \in \mathcal{N} & (2.10) \\
\quad < \mathbf{Y}_i^R, W >= p_i - \mathbf{p}_i^c, \quad i \in \mathcal{P} & (2.44) \\
\quad Z = \begin{bmatrix} W & \star \\ \star & N \end{bmatrix} = \begin{bmatrix} v \\ p \end{bmatrix} \begin{bmatrix} v \\ p \end{bmatrix}^t = \begin{bmatrix} vv^t & vp^t \\ pv^t & pp^t \end{bmatrix}. & (2.45)
\end{cases}
$$

$(OPF - Q_T)$ est bien une reformulation exacte de $(OPF - Q)$ car quand (2.45) est vérifiée, les fonctions objectifs ont même valeur, tandis que les ensembles de contraintes définissent les mêmes espaces admissibles.

Notons que cette reformulation est obtenue en deux temps, exactement comme pour $(OPF - L)$, voir Section 2.2. La fonction objectif est quadratique convexe car T est une matrice semi-définie positive. Les contraintes, sans compter la contrainte (2.45), sont linéaires. En prenant $T = 0_{2n+n_p}$, on obtient la « reformulation linéaire » canonique de $(OPF - Q)$.

En faisant parcourir l'ensemble des matrices semi-définies positives à T, on fait décrire à $(OPF - Q_T)$ une famille de reformulations quadratiques et exactes de (OPF).

2.6.2 Relaxation du rang

La relaxation semi-définie positive (SDP-Q) canonique, appelée relaxation du rang, de $(OPF - Q)$ est le programme d'optimisation suivant :

$$(SDP-Q)\begin{cases} \min\limits_{Z,u,v,p} \quad f^q_{0_{2n+n_p}}(Z,v,p) = \mathbf{c}_1^t p + < \mathbf{C}_2, N > + \mathbf{c}_0^q \\ \text{s.t.} \\ \quad (8),(2.6),(2.7),(2.9),(2.10),(2.44), \\ \quad u = 1, & (2.46) \\ \quad \begin{bmatrix} u & v^t & p^t \\ v & W & \star \\ p & \star & N \end{bmatrix} \succeq 0. & (2.47) \end{cases}$$

Où u est un réel.

$(SDP-Q)$ est une relaxation de $(OPF-Q)$. Pour le prouver, prenons la reformulation quadratique obtenue avec $T = 0_{2n+n_p}$. La contrainte (2.45) peut se relaxer en :

$$Z - \begin{bmatrix} v \\ p \end{bmatrix} \begin{bmatrix} v \\ p \end{bmatrix}^t \succeq 0, \tag{2.48}$$

ce qui est équivalent à :

$$\begin{bmatrix} 1 & v^t & p^t \\ v & W & \star \\ p & \star & N \end{bmatrix} \succeq 0. \tag{2.49}$$

$(SDP-Q)$ est obtenue à partir de $(OPF-Q_{0_{2n+n_p}})$ en effectuant cette relaxation de la contrainte (2.45). On appelle cette relaxation la relaxation du rang, car c'est la contrainte $\text{rang}(Z) = 1$ qui est relaxée dans (2.45) pour se ramener à (2.49).

Le dual de $(SDP - Q)$ s'écrit :

$$(D-SDP-Q)\begin{cases} \max\limits_{\lambda^R,\lambda^I,\alpha,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma},\eta} g^q(\lambda^R,\lambda^I,\alpha,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma},\eta) = \\ (\mathbf{p}^c)^t\lambda^R + (\mathbf{q}^c)^t\lambda^I + (\mathbf{p}^c)^t\alpha + \underline{\mathbf{p}}^t\underline{\mu}^R - \overline{\mathbf{p}}^t\overline{\mu}^R + \\ \quad (\underline{\mathbf{q}} - \mathbf{q}^c)^t\underline{\mu}^I + (\mathbf{q}^c - \overline{\mathbf{q}})^t\overline{\mu}^I + (\underline{\mathbf{v}}^2)^t\underline{\gamma} - (\overline{\mathbf{v}}^2)^t\overline{\gamma} - \eta + \mathbf{c}_0^q \\ \text{s.t.} \\ \quad \begin{bmatrix} \eta & 0 & \frac{(\mathbf{c}_1-\alpha-\underline{\mu}^R+\overline{\mu}^R)^t}{2} \\ 0 & S & 0 \\ \frac{\mathbf{c}_1-\alpha-\underline{\mu}^R+\overline{\mu}^R}{2} & 0 & \mathbf{C}_2 \end{bmatrix} \succeq 0 & (2.50) \\ \quad (2.15)-(2.22). \end{cases}$$

Où :

- $\lambda^R \in \mathbb{R}^{n_c}$ est le vecteur des variables duales associées aux contraintes (2.6).
- $\lambda^I \in \mathbb{R}^{n_c}$ est le vecteur des variables duales associées aux contraintes (2.7).
- $\alpha \in \mathbb{R}^{n_p}$ est le vecteur des variables duales associées aux contraintes (2.44).
- $\underline{\mu}^R \in \mathbb{R}^{n_p}$ et $\overline{\mu}^R \in \mathbb{R}^{n_p}$ sont les vecteurs des variables duales associées respectivement aux contraintes de supériorité et d'infériorité (8).
- $\underline{\mu}^I \in \mathbb{R}^{n_p}$ et $\overline{\mu}^I \in \mathbb{R}^{n_p}$ sont les vecteurs des variables duales associées respectivement aux contraintes de supériorité et d'infériorité (2.9).
- $\underline{\gamma} \in \mathbb{R}^n$ et $\overline{\gamma} \in \mathbb{R}^n$ sont les vecteurs des variables duales associées respectivement aux contraintes de supériorité et d'infériorité (2.10).
- η est la variable duale associée à la contrainte (2.46).
- $\underline{\mathbf{v}}^2$ et $\overline{\mathbf{v}}^2$ représentent les vecteurs de $\mathbb{R}^n$ dont la i-ème composante vaut respectivement $\underline{\mathbf{v}_i}^2$ et $\overline{\mathbf{v}_i}^2$.
- $S = \sum_{i\in\mathcal{C}} \lambda_i^R \mathbf{Y}_i^R + \lambda_i^I \mathbf{Y}_i^I + \sum_{i\in\mathcal{P}} \alpha_i \mathbf{Y}_i^R + (\overline{\mu}_i^I - \underline{\mu}_i^I)\mathbf{Y}_i^I + \sum_{i\in\mathcal{N}} (\overline{\gamma}_i - \underline{\gamma}_i) 1_{i,i} + (\overline{\gamma}_i - \underline{\gamma}_i) 1_{i+n,i+n}$.
- Pour tout i entre 1 et $2n$, $1_{i,i}$ est la matrice de taille $2n \times 2n$ dont toutes les composantes sont nulles sauf la i-ème composante de la diagonale, qui vaut 1.

On sait que pour un problème SDP, si le primal est réalisable et le dual est strictement réalisable, alors il n'y a pas de saut de dualité. De plus l'ensemble des solutions est non vide et borné. De manière similaire à ce qui est fait dans la Section 2.2.2, nous montrons qu'il y a dualité forte pour $(SDP - Q)$ quand $(OPF - Q)$ est réalisable. Ce résultat n'est pas nouveau et peut être trouvé dans [39]. Nous en proposons une autre démonstration éclairant les futurs résultats de cette section.

Proposition 3. *Pour un problème $(OPF - Q)$ réalisable, le saut de dualité entre la relaxation du rang $(SDP - Q)$ et son dual $(D - SDP - Q)$ est nul. Il y a alors dualité forte pour la relaxation du rang.*

Preuve. Pour démontrer ce résultat, nous allons procéder en deux temps, premièrement nous allons montrer que $(SDP - Q)$ est réalisable, puis que $(DSDP - Q)$ est strictement réalisable.

Première partie :

Comme $(OPF - Q)$ est réalisable, notons (v^*, p^*) une de ces solutions réalisables.

Dans ce cas, (Z^*, u^*, v^*, p^*) avec $Z^* = \begin{bmatrix} v^*(v^*)^t & v^*(p^*)^t \\ p^*(v^*)^t & p^*(p^*)^t \end{bmatrix}$ et $u^* = 1$ est une solution réalisable de $(SDP - Q)$.

En particulier avec ces valeurs, on vérifie bien la contrainte (2.48), ce qui assure la validité de la contrainte (2.47). $(SDP - Q)$ est donc réalisable.

Seconde partie :

Dans un premier temps posons :

— $\lambda_i^R = 0, \quad \forall i \in \mathcal{C}$.

— $\lambda_i^I = 0, \quad \forall i \in \mathcal{C}$.

— $\alpha = \mathbf{c}_1$.

— $\underline{\mu}_i^R = 1, \quad \forall i \in \mathcal{P}$.

— $\overline{\mu}_i^R = 1, \quad \forall i \in \mathcal{P}$.

— $\underline{\mu}_i^I = 1, \quad \forall i \in \mathcal{P}$.

— $\overline{\mu}_i^I = 1, \quad \forall i \in \mathcal{P}$.

— $\underline{\gamma}_i = 1, \quad \forall i \in \mathcal{N}$.

— $\overline{\gamma}_i = \gamma, \quad \forall i \in \mathcal{N}$.

— $\eta = 0$.

Où γ est une quantité strictement positive encore à définir.

Avec ces valeurs pour les variables duales, les contraintes (2.15)-(2.22) sont strictement vérifiées.

De plus, la contrainte (2.50) devient avec ces valeurs : $\begin{bmatrix} 0 & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & \mathbf{C}_2 \end{bmatrix} \succeq 0$,

avec $S = \sum_{i \in \mathcal{P}} (\mathbf{c}_1)_i \mathbf{Y}_i^R - \sum_{i \in \mathcal{N}} (1_{i,i} + 1_{i+n,i+n}) + \gamma \sum_{i \in \mathcal{N}} (1_{i,i} + 1_{i+n,i+n})$.

On remarquera également que $\sum_{i \in \mathcal{N}} (1_{i,i} + 1_{i+n,i+n}) = \mathcal{I}_{2n}$, qui est la matrice identité de taille $2n \times 2n$.

En notant $S' = \sum_{i \in \mathcal{P}} (\mathbf{c}_1)_i \mathbf{Y}_i^R - \mathcal{I}_{2n}$, on obtient $S = S' + \gamma \mathcal{I}_{2n}$.

En décomposant par bloc la contrainte (2.50), on obtient que celle-ci est équivalente au fait que S et $\mathbf{C}_2$ soient SDP. Comme $\mathbf{C}_2$ est SDP par définition, il ne reste qu'à montrer que S peut être rendue SDP, voire strictement SDP pour que la contrainte (2.50) soit rendue strictement réalisable, en choisissant une bonne valeur pour γ. C'est chose faite en posant $\gamma = 1 + \max(0, -\text{vp}_{\min}(S'))$. Où $\text{vp}_{\min}(S')$ est

la plus petite valeur propre de S'. Ainsi avec les valeurs données pour les variables duales, (2.50) est strictement vérifiée , ce qui achève de prouver que $(D - SDP - Q)$ est strictement réalisable. $\square$

Cette proposition est importante, car nous allons utiliser les variables duales de $(SDP - Q)$ pour construire une reformulation quadratique optimale de $(OPF - Q)$.

2.6.3 Reformulation optimale

Après avoir introduit la famille des reformulations quadratiques $(OPF - Q_T)$, la relaxation SDP appelée relaxation du rang ainsi que son dual, nous allons voir comment obtenir une reformulation quadratique dite optimale de $(OPF - Q)$ à partir de cette relaxation du rang.

2.6.3.1 Définition de la notion de reformulation optimale

En Section 2.6.1 nous avons décrit comment reformuler le problème $(OPF - Q)$ en introduisant une matrice SDP notée T. Le choix de cette matrice a un impact sur la reformulation obtenue. Nous cherchons alors à déterminer la matrice T fournissant la meilleure reformulation $(OPF - Q_S)$ selon un certain critère, exactement comme dans le cas de $(OPF - L)$. Nous allons procéder comme dans la Section 2.2, en passant par les mêmes étapes, mais en les adaptant au cas de $(OPF - Q)$. La reformulation obtenue sera résolue avec un algorithme de type *branch-and-bound*. Il semble alors légitime de chercher à maximiser la valeur de cette borne inférieure initiale.

La relaxation à la racine que l'on considère pour l'instant consiste à supprimer la contrainte (2.45) dans $(OPF - Q_T)$, cette contrainte étant la seule non-convexe. La fonction objectif f_T^q étant quadratique et convexe, la relaxation considérée est alors bien quadratique convexe. On la note $(\overline{OPF - Q_T})$:

$$(\overline{OPF - Q_T}) \begin{cases} \min\limits_{Z,v,p} \quad f_T^q(Z,v,p) \\ \text{s.t.} \\ \quad (8), (2.6), (2.7), (2.9), (2.10), (2.44), \\ \end{cases} \tag{2.51}$$

La construction d'une reformulation optimale se fait en cherchant une solution optimale du problème

d'optimisation (P) suivant :

$$(P)\begin{cases} \max\limits_{T} \quad val(\overline{OPF - Q_T}) \\ \text{s.t.} \\ \quad T \succeq 0 \end{cases} \qquad (2.23)\ .$$

2.6.3.2 Construction d'une reformulation optimale

Le cœur de cette section se situe dans le théorème suivant :

Théorème 2. *La matrice SDP T^* construite à l'aide d'une solution optimale du problème $(D-SDP-Q)$ est une solution optimale de (P).*

Preuve. Pour démontrer ce résultat, nous procédons par étapes. i) Nous réécrivons de manière équivalente le problème $(D - SDP - Q)$. ii) Nous construisons une relaxation quadratique convexe $(\overline{OPF - Q_{T^*}})$ telle que $val(P) \geq val(\overline{OPF - Q_{T^*}}) \geq val(SDP{-}Q)$. iii) Nous montrons que $val(SDP{-}Q) \geq val(P)$. Ceci nous permet de conclure que $val(P) = val(\overline{OPF - Q_{T^*}}) = val(SDP - Q)$.

i) Problème équivalent à $(D - SDP - Q)$:

Dans cette section, nous allons mettre en avant un problème d'optimisation équivalent à $(D - SDP - Q)$, ce qui nous servira dans la suite pour donner une solution optimale de (P).

Dans un premier temps, réécrivons le problème $(D - SDP - Q)$ en introduisant une matrice de variables d'écarts T_D.

$$(D-SDP-Q)\begin{cases} \max\limits_{T_D,\lambda^R,\lambda^I,\alpha,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma},\eta} g^q(\lambda^R,\lambda^I,\alpha,\underline{\mu}^R,\overline{\mu}^R,\underline{\mu}^I,\overline{\mu}^I,\underline{\gamma},\overline{\gamma},\eta) \\ \text{s.t.} \\ \quad T_D = \begin{bmatrix} \eta & 0 & \frac{(\mathbf{c}_1-\alpha-\underline{\mu}^R+\overline{\mu}^R)^t}{2} \\ 0 & S & 0 \\ \frac{\mathbf{c}_1-\alpha-\underline{\mu}^R+\overline{\mu}^R}{2} & 0 & \mathbf{C}_2 \end{bmatrix}, & (2.52) \\ \quad (2.15)-(2.22), \\ \quad T_D \succeq 0. & (2.53) \end{cases}$$

Pour T_D une matrice SDP fixée, notons $(D-SDP-Q(T_D))$ le programme d'optimisation résultant en la maximisation de la fonction objectif g^q sous les contraintes (2.15)-(2.22) et (2.52), de telle sorte à

ce que le programme suivant soit équivalent à $(D - SDP - Q)$:

$$(D - SDP - Q) \begin{cases} \max\limits_{T_D} & val(D - SDP - Q(T_D)) \\ \text{s.t.} & \\ & (2.53). \end{cases}$$

$(D - SDP - Q(T_D))$ est un programme d'optimisation linéaire, il est donc équivalent à son dual. En introduisant la matrice de variables duales Z_P correspondant à la contrainte (2.52), et en dualisant $(D - SDP - Q(T_D))$, on construit le problème primal noté $(P - SDP - Q(T_D))$:

$$(P - SDP - Q(T_D)) \begin{cases} \min\limits_{Z_P} & \mathbf{c}_1^t p + < \mathbf{C}_2, N > - < T_D, Z_P > + \mathbf{c}_0^q \\ \text{s.t.} & \\ & (8), (2.6), (2.7), (2.9), (2.10), (2.44), \\ & (Z_P)_{1,1} = 1. \end{cases} \tag{2.54}$$

Où Z_P a été décomposée de la manière suivante :
$Z_P = \begin{bmatrix} 1 & v^t & p^t \\ v & W & \star \\ p & \star & N \end{bmatrix}$.
$(D - SDP - Q)$ est alors équivalent au problème suivant :

$$(D - SDP - Q) \begin{cases} \max\limits_{T_D} & val(P - SDP - Q(T_D)) \quad (2.55) \\ \text{s.t.} & \\ & T_D \succeq 0. \quad (2.56) \end{cases}$$

ii) On montre que $val(\overline{OPF - Q_{T^*}}) \geq val(D - SDP - Q) = val(SDP - Q)$:

Soit T_D^* une solution optimale de $(D - SDP - Q)$. $(P - SDP - Q(T_D^*))$ a donc même valeur que $(D - SDP - Q)$. On ajoute un terme positif dans la fonction objectif de $(P - SDP - Q(T_D^*))$. Ce terme est :
$\begin{bmatrix} 1 \\ v \\ p \end{bmatrix}^t T_D^* \begin{bmatrix} 1 \\ v \\ p \end{bmatrix}$.

Le problème obtenu est alors $(\overline{OPF - Q_{T^*}})$, où T^* est la sous matrice de T_D^* ne contenant pas sa première ligne et sa première colonne. Ainsi en résolvant la relaxation du rang de $(OPF - Q)$, on est à même de construire la reformulation $(OPF - Q_{T^*})$ dont la relaxation initiale $(\overline{OPF - Q_{T^*}})$ aura au

moins la valeur de cette même relaxation du rang. On va maintenant montrer que cette reformulation est solution optimale du problème (P).

iii) On montre que $\forall T, val(SDP - Q) \geq val(\overline{OPF - Q_T})$:

Soit T une matrice SDP. Nous allons ici montrer que la relaxation du rang $(SDP - Q)$ a une valeur optimale supérieure à $(\overline{OPF - Q_T})$. Comme la valeur optimale de la relaxation $(\overline{OPF - Q_{T^*}})$ est supérieure à la valeur optimale de la relaxation du rang (voir section précédente), on aura alors prouvé que T^* est une solution optimale du problème (P), et que la résolution de la relaxation du rang fournit une reformulation quadratique optimale.

Pour prouver que $val(SDP - Q) \geq val(\overline{OPF - Q_T})$, nous allons montrer que pour toute solution réalisable de $(SDP - Q)$, on peut construire une solution réalisable de $(\overline{OPF - Q_T})$ ayant une valeur inférieure.

Soit (Z^*, u^*, v^*, p^*) une solution de $(SDP - Q)$, avec $Z^* = \begin{bmatrix} W^* & \star \\ \star & N^* \end{bmatrix}$.
(Z^*, v^*, p^*) est alors une solution réalisable de $(\overline{OPF - Q_T})$.
Comparons leur valeur en soustrayant la fonction objectif de $(SDP - Q)$ à celle de $(\overline{OPF - Q_T})$:

$$\begin{aligned} \mathbf{c}_1^t p^* + < \mathbf{C}_2, N^* > + \begin{bmatrix} v^* \\ p^* \end{bmatrix}^t T \begin{bmatrix} v^* \\ p^* \end{bmatrix} - < T, Z^* > + \mathbf{c}_0^q - \mathbf{c}_1^t p^* - < \mathbf{C}_2, N^* > - \mathbf{c}_0^q \\ = \quad < T, \begin{bmatrix} v^* \\ p^* \end{bmatrix} \begin{bmatrix} v^* \\ p^* \end{bmatrix}^t - Z^* > . \end{aligned}$$

Or, comme (Z^*, u^*, v^*, p^*) vérifie la contrainte (2.47) et que $u^* = 1$,
on vérifie : $Z^* - \begin{bmatrix} v^* \\ p^* \end{bmatrix} \begin{bmatrix} v^* \\ p^* \end{bmatrix}^t \succeq 0$.
Ceci implique que, T étant SDP, $< T, \begin{bmatrix} v^* \\ p^* \end{bmatrix} \begin{bmatrix} v^* \\ p^* \end{bmatrix}^t - Z^* > \quad \leq \quad 0$.

On a alors démontré que $(SDP - Q)$ a une valeur optimale supérieure à $(\overline{OPF - Q_T})$, et ce quelque soit la matrice T SDP considérée. Ceci permet de conclure quant à la validité du théorème. □

De plus le corollaire suivant s'applique :

Proposition 4. *Les relaxations convexes $(SDP - Q)$ et $(\overline{OPF - Q_{T^*}})$ ont même valeur.*

Soit $(T_D^*, \lambda^{R^*}, \lambda^{I^*}, \alpha^*, \underline{\mu}^{R^*}, \overline{\mu}^{R^*}, \underline{\mu}^{I^*}, \overline{\mu}^{I^*}, \underline{\gamma}^*, \overline{\gamma}^*, \eta^*)$ une solution optimale de $(D - SDP - Q)$. On extrait

$$T^* = \begin{bmatrix} S^* & 0 \\ 0 & \mathbf{C}_2 \end{bmatrix},$$

avec :

$$S^* = \sum_{i \in \mathcal{C}} \left(\lambda_i^{R^*} \mathbf{Y}_i^R + \lambda_i^{I^*} \mathbf{Y}_i^I \right) + \sum_{i \in \mathcal{P}} \left(\alpha_i^* \mathbf{Y}_i^R + (\overline{\mu}_i^{I^*} - \underline{\mu}_i^{I^*}) \mathbf{Y}_i^I \right) + \sum_{i \in \mathcal{N}} \left((\overline{\gamma}_i{}^* - \underline{\gamma}_i{}^*) 1_{i,i} + (\overline{\gamma}_i{}^* - \underline{\gamma}_i{}^*) 1_{i+n,i+n} \right).$$

La reformulation quadratique optimale prend alors la forme :

$$(OPF - Q_{T^*}) \begin{cases} \min\limits_{Z,v,p} \quad f_{T^*}^q(Z, v, p) = v^t S^* v + p^t \mathbf{C}_2 p + \mathbf{c}_1^t p - < S^*, W > + \mathbf{c}_0^q \\ \text{s.t.} \\ \quad \underline{\mathbf{p}_i} \leq p_i \leq \overline{\mathbf{p}_i}, \quad \forall i \in \mathcal{P} & (8) \\ \quad < \mathbf{Y}_i^R, W >= -\mathbf{p}_i^c, \quad i \in \mathcal{C} & (2.6) \\ \quad < \mathbf{Y}_i^I, W >= -\mathbf{q}_i^c, \quad i \in \mathcal{C} & (2.7) \\ \quad \underline{\mathbf{q}_i} - \mathbf{q}_i^c \leq < \mathbf{Y}_i^I, W > \leq \overline{\mathbf{q}_i} - \mathbf{q}_i^c, \quad i \in \mathcal{P} & (2.9) \\ \quad \underline{\mathbf{v}_i}^2 \leq W_{i,i} + W_{i+n,i+n} \leq \overline{\mathbf{v}_i}^2, \quad i \in \mathcal{N} & (2.10) \\ \quad < \mathbf{Y}_i^R, W >= p_i - \mathbf{p}_i^c, \quad i \in \mathcal{P} & (2.44) \\ \quad Z = \begin{bmatrix} v \\ p \end{bmatrix} \begin{bmatrix} v \\ p \end{bmatrix}^t = \begin{bmatrix} vv^t & vp^t \\ pv^t & pp^t \end{bmatrix}. & (2.45) \end{cases}$$

Il faut noter que seule la partie de T^* notée S^* n'est utile pour construire la reformulation quadratique optimale $(OPF - Q_{T^*})$.

2.6.4 Ajout des inégalités de McCormick

Pour construire la relaxation $(\overline{OPF - Q_T})$ de la reformulation $(OPF - Q_T)$, on a supprimé la contrainte quadratique non convexe (2.45). On peut se demander si une meilleure relaxation quadratique

convexe peut être obtenue en relâchant différemment la contrainte (2.45), par exemple en ajoutant les inégalités de McCormick à la relaxation.

Dans le cas qui nous intéresse, nos variables sont les variables de tension v et de puissance p. On peut distinguer trois familles de produits entre ces variables :

1. Les produits entre deux tensions v_i et v_j. Dans $(OPF - Q_T)$ c'est la variable $W_{i,j}$ qui représente le produit $v_i v_j$.
2. Les produits entre deux puissances p_i et p_j. Dans $(OPF - Q_T)$ c'est la variable $N_{i,j}$ qui représente le produit $p_i p_j$.
3. Les produits entre une tension v_i et une puissance p_j. La variable représentant le produit $v_i p_j$ n'a pas de nom particulier dans la matrice de variables Z. Nous l'appelons $R_{i,j}$ à partir de maintenant.

Pour pouvoir appliquer les inégalités de McCormick au problème $(OPF - Q_T)$, il faut des bornes sur les variables v et p. Les variables de puissance p sont déjà bornées par les contraintes (8). On peut déduire des bornes sur les variables de tension v à l'aide des contraintes (2.5) comme cela est fait pour $(OPF - L)$:

$$\forall i \in N, \quad -\underline{\mathbf{v}_i} \leq v_i \leq \overline{\mathbf{v}_i} \quad \text{et} \quad -\underline{\mathbf{v}_i} \leq v_{i+n} \leq \overline{\mathbf{v}_i}.$$

Nous allons montrer dans la suite, qu'avec ces bornes, ajouter les inégalités de McCormick aux relaxations $(\overline{OPF - Q_T})$ ne permet pas d'obtenir une meilleure reformulation. En effet, les relaxations de la famille $(\overline{OPF - Q_T})$ ont toujours une valeur inférieure à celle de la relaxation du rang, même après ajout des inégalités de McCormick. La reformulation $(OPF - Q_{T^*})$ reste alors une reformulation optimale de $(OPF - Q)$, même lorsque les relaxations considérées sont renforcées par les inégalités de McCormick.

Décomposition parcimonieuse de (SDP) :

Pour démontrer le résultat ci-dessus, il faut d'abord introduire la résolution par décomposition parcimonieuse de la relaxation du rang.

Si l'on regarde la contrainte de semi-définie positivité (2.47) dans $(SDP - Q)$, on peut noter que les variables v n'interviennent que dans cette contrainte du problème, de même que les variables des blocs notés $\star$. Ceci fait que le bloc W est complétement séparé du reste des variables prenant part à la fonction objectif et aux contraintes. D'après le théorème de décomposition parcimonieuse, la contrainte (2.47) peut alors se décomposer comme ceci dans le cadre de la résolution de $(SDP - Q)$:

$$\begin{cases} W \succeq 0, \\ \begin{bmatrix} u & p^t \\ p & N \end{bmatrix} \succeq 0. \end{cases}$$

De plus, comme il n'y a que les éléments diagonaux de N qui interviennent dans l'écriture de la fonction objectif et des contraintes de $(SDP - Q)$ (car $\mathbf{C}_2$ est diagonale), on peut encore décomposer la contrainte :
$\begin{bmatrix} u & p^t \\ p & N \end{bmatrix} \succeq 0.$
On rappelle que $u = 1$, il s'ensuit que cette contrainte se décompose, à l'aide du théorème de décomposition parcimonieuse, et dans le cadre de la résolution de $(SDP - Q)$ en :
$\forall i \in \mathcal{P}, N_{i,i} \geq p_i^2$. Cette décomposition parcimonieuse va nous permettre de déduire plusieurs résultats quant à la forme que l'on peut donner à une solution optimale de $(SDP - Q)$.

Proposition 5. *À l'optimum de $(SDP - Q)$, le résultat suivant s'applique : $\forall i \in \mathcal{P}, N_{i,i} = p_i^2$.*

Preuve. Pour $i \in \mathcal{P}$, $N_{i,i}$ n'apparait que dans la fonction objectif avec le terme $(\mathbf{C}_2)_{i,i}$ qui est positif, et avec la contrainte $N_{i,i} \geq p_i^2$. Il s'ensuit qu'à l'optimum, $N_{i,i} = p_i^2$. □

Proposition 6. *Il existe une solution optimale de $(SDP - Q)$ telle que : $N = pp^t$.*

Preuve. On a déjà montré plus haut qu'à l'optimum, $\forall i \in \mathcal{P}, N_{i,i} = p_i^2$. De plus, pour i et j deux éléments différents de $\mathcal{P}$, le terme $N_{i,j}$ n'intervient dans le problème que pour assurer la contrainte suivante :
$\begin{bmatrix} 1 & p^t \\ p & N \end{bmatrix} \succeq 0.$
Or, si l'on choisit $N_{i,j} = p_i p_j$, cette contrainte est respectée. La proposition est alors démontrée. □

Proposition 7. *Il existe une solution optimale de $(SDP - Q)$ telle que : $N = pp^t$ et $v = 0$.*

Preuve. v n'apparait dans le problème $(SDP - Q)$ que pour assurer la contrainte (2.47). On sait qu'il existe une solution optimale pour laquelle $W \succeq 0$ et $N = pp^t$. En prenant $v = 0$, on assure que la

contrainte (2.47) est vérifiée. On a ainsi une solution optimale pour laquelle $N = pp^t$ et $v = 0$. $\square$

Avec ces propositions, il sera aisé de montrer que la relaxation du rang a une valeur optimale supérieure aux relaxations $(\overline{OPF - Q_T})$ renforcées avec les inégalités de McCormick.

Proposition 8. *La relaxation du rang $(SDP - Q)$ a une valeur optimale supérieure aux relaxations quadratiques convexes $(\overline{OPF - Q_T})$ même lorsqu'on leur ajoute les inégalités de McCormick.*

Preuve. Pour montrer ce résultat, nous allons procéder exactement comme précédemment lorsque nous avons montré que la relaxation du rang a une valeur optimale supérieure aux relaxations $(\overline{OPF - Q_T})$ dans la Section 2.6.3.
Soit T une matrice SDP, nous savons à l'aide des propositions précédentes qu'il existe une solution optimale (Z^*, u^*, v^*, p^*) de $(SDP - Q)$ avec : $Z^* = \begin{bmatrix} W^* & \star \\ \star & N^* \end{bmatrix}$ et vérifiant $W^* \succeq 0$, $N^* = p^*(p^*)^t$, $u^* = 1$ et $v^* = 0$.
En réutilisant ce qui a été fait précédemment, nous savons que (Z^*, v^*, p^*) est une solution de $(\overline{OPF - Q_T})$ ayant une valeur inférieure à la relaxation du rang. Il ne reste plus qu'à montrer que (Z^*, v^*, p^*) vérifie les inégalités de McCormick pour pouvoir conclure.

Commençons par les produits de la première famille, c'est-à-dire pour i et j dans $\{1, \ldots, 2n\}$, le produit $v_i v_j$, représenté par la variable $W_{i,j}$. Comme $W^* \succeq 0$, on a : $\forall i \in \{1, \ldots, 2n\}, W^*_{i,i} \geq 0$.
Et de plus, d'après (2.10) : $\forall i \in \{1, \ldots, 2n\}, W^*_{i,i} \leq \overline{\mathbf{v}_i}^2$ [2]. Comme $v^* = 0$, les quatres inégalités de McCormick deviennent :

$$\begin{cases} W^*_{i,j} \leq \overline{\mathbf{v}_i}^2 \overline{\mathbf{v}_j}^2 \\ W^*_{i,j} \geq -\overline{\mathbf{v}_i}^2 \overline{\mathbf{v}_j}^2 \end{cases}$$

Ce qui est le cas car comme $W^* \succeq 0$, le déterminant mineur d'odre 2 relatif aux lignes et colonnes i et j est positif, ce qui implique que ${W^*_{i,j}}^2 \leq W^*_{i,i} W^*_{j,j}$. En utilisant les bornes données plus hauts pour les $W^*_{i,i}$, on en déduit que les inégalités de McCormick sont valides pour les produits de la première famille.

Pour les produits de la deuxième famille, les inégalités de McCormick sont automatiquement vérifiées, car $N^* = p^*(p^*)^t$ Pour les produits de la troisième famille, elles sont aussi vérifiées, en posant $R^*_{i,j} = v^*_i p^*_j = 0$.

2. Ici, on a étendu sur $\{n+1, \ldots, 2n\}$ la définition de $\overline{\mathbf{v}}$, en considérant que pour $i \in N, \overline{\mathbf{v}}_i = \overline{\mathbf{v}_{i+n}}$

La proposition est donc bien démontrée. □

La proposition ci-dessus nous assure que la relaxation du rang $(SDP - Q)$ fournit, grâce à la matrice duale optimale T^*, une reformulation optimale, même dans le cas où l'on considère les inégalités de McCormick dans la relaxation $(\overline{OPF - Q_T})$.

2.6.5 Conclusion sur la résolution de $(OPF - Q)$ avec `RC-OPF`

Notre méthode fournit une reformulation optimale au sein de la famille de reformulations quadratiques qui vient « convexifier » de manière quadratique l'objectif et les contraintes du problème de départ. On retrouve alors le résultat de MIQCR mais où seule la résolution d'un problème SDP plus petit, la relaxation dite du rang, est nécessaire. Nous venons dans cette section de généraliser la méthode de reformulation quadratique optimale à la variante $(OPF - Q)$. Nous pouvons ainsi utiliser l'algorithme **??** `RC-OPF-Q` pour résoudre de manière exacte une instance $(OPF - Q)$.

La relaxation $(\overline{OPF - Q_{T^*} - MCC})$ résolue en chaque nœud de notre algorithme `RC-OPF-Q` prend la forme suivante :

$$(\overline{OPF - Q_{T^*} - MCC}) \begin{cases} \min\limits_{W,v,p} \quad \mathbf{c}_1^t p + p^t \mathbf{C}_2 p + v^t S^* v - < S^*, W > + \mathbf{c}_0^q \\ \text{s.t.} \\ \quad (8), (2.6), (2.7), (2.9), (2.10), (2.44) \\ \quad (v_i, v_j, W_{i,j}) \in MCC, \quad (i,j) \in \{1, \ldots, 2n\}^2. \end{cases}$$

Algorithme 2 : Algorithme `RC-OPF-Q`

1 `Phase 0` : Transformer le problème quadratique à variables complexes en un problème quadratique à variables réelles;
2 `Phase 1` :
3 Résoudre la relaxation du rang $(SDP - Q)$: $LB \leftarrow val(SDP - Q)$;
4 À l'aide de la solution duale optimale de $(SDP - Q)$, construire la reformulation $(OPF - Q_{T^*})$;
5 `Phase 2` :
6 Déterminer le nœud électrique de référence qui est un nœud électrique ayant un maximum de voisins dans le réseau ainsi que les bornes supérieures et inférieures sur les variables v comme dans (2.29) et (2.31);
7 Lancer une heuristique de résolution de $(OPF - Q)$ ayant $V_i = 1, \forall i \in \mathcal{N}$ comme point de départ : $UB \leftarrow val(Heuristique)$;
8 **tant que** $\frac{UB-LB}{UB} \geq \epsilon = 10^{-3}$ **faire**
9 Choisir la feuille non coupée de l'arbre avec le potentiel le plus élevé;
10 Appeler une méthode de contraction de bornes (voir chapitre 3);
11 Résoudre la relaxation $(\overline{OPF - Q_{T^*} - MCC})$ avec les bornes sur v du nœud courant;
12 **si** *la relaxation est infaisable* **alors**
13 Couper le nœud;
14 **sinon**
15 **si** $val(\overline{OPF - Q_{T^*} - MCC}) \geq UB$ **alors**
16 Couper le nœud;
17 **sinon**
18 Déterminer la variable de branchement v_c avec la méthode « erreur bilinéaire pondérée », sans donner la priorité aux termes diagonaux;
19 Déterminer la valeur de branchement v_c^{cut}. Ici on choisit $\delta = 0.5$. $v_c^{cut} \leftarrow \delta \frac{\ell_c + u_c}{2} + (1 - \delta)\hat{v}_c$;
20 Construire deux nouveaux nœuds comme fils du nœud courant avec comme potentiel la valeur de la relaxation du nœud courant. Les bornes sur v de chaque fils sont celles du nœud courant, sauf pour v_c où l'un des fils remplace la borne inférieure par v_c^{cut}, et l'autre sa borne supérieure par v_c^{cut};
21 Lancer l'heuristique de résolution de $(OPF - Q)$ avec comme point de départ la solution en v de la relaxation;
22 **si** $val(Heuristique) < UB$ **alors**
23 $UB \leftarrow val(Heuristique)$;
24 Remplacer la valeur de LB par celle du plus petit potentiel parmi les feuilles de l'arbre non coupées.

Chapitre 3

Contractions de bornes dans la méthode RC-OPF

Contenu

3.1 Contractions fondées sur l'optimalité en phase 2 120
3.1.1 Contraction de bornes par résolution de problèmes convexes 121
3.1.2 Synergie avec la reformulation quadratique optimale 122
3.1.3 Contractions fondées sur les coûts réduits 122
3.1.4 Algorithme 122
3.2 Contractions fondées sur la faisabilité en phase 2 123
3.2.1 Contractions linéaires [1] 123
3.2.2 Contractions non-linéaires 123
3.2.3 Algorithme 125
3.3 Extension des contractions de bornes à $(OPF - Q)$ 128

Ce chapitre est consacré à la présentation des contractions de bornes effectuées pendant l'éxécution de `RC-OPF` pour résoudre à l'optimum global $(OPF-L)$. Ces considérations sont étendues à la résolution du problème $(OPF-Q)$. Comme montré dans les expérimentations numériques du chapitre 5, lorsque la relaxation du rang n'a pas la même valeur optimale que le problème d'origine, notre algorithme a besoin de parcourir un grand nombre de nœuds dans l'arbre avant de converger. Ce comportement peut être atténué en appliquant les méthodes de contractions, vues dans l'état de l'art, sur les bornes des variables v de tension. Ces différentes contractions peuvent être utilisées en chaque nœud de l'arbre de notre algorithme de *branch-and-bound*, comme pre-processing avant que la relaxation quadratique convexe $(\overline{OPF-L_{S^*}})$ n'ait été résolue. On cherche prioritairement à contracter le domaine des variables v, mais on peut aussi utiliser certaines de ces techniques sur les domaines des variables W.

La première section rappelle les techniques appartenant à la classe OBBT (Optimality-Based Bound Tightening [43]) présente dans notre algorithme de résolution. Nous détaillons également l'apport de l'utilisation de notre reformulation quadratique optimale en synergie avec les techniques de l'état de l'art. La deuxième section s'attarde sur les techniques appartenant à la classe FBBT(Feasiblity-Based Bound Tightening [1]) et présente notamment des réductions non-linéaires originales pour l'OPF. La troisième section explique comment réutiliser ces techniques dans la résolution de $(OPF-Q)$.

3.1 Contractions fondées sur l'optimalité en phase 2

Cette section présente une amélioration des techniques OBBT (pour *Optimality-Based Bound Tightening*, ou contraction de bornes fondée sur l'optimalité) mises en œuvre pour accélérer la résolution de $(OPF-L)$ avec notre algorithme de *branch-and-bound* spatial. Dans un premier temps, on présente l'ajout d'une contrainte quadratique convexe liant l'objectif à une borne supérieure. On met en lumière l'efficacité de l'ajout de cette contrainte en montrant la synergie avec la reformulation quadratique optimale $(OPF-L_{S^*})$. Le fait d'utiliser la matrice de reformulation S^* dans cette contrainte constitue une nouveauté et renforce fortement l'efficacité de cette méthode quand elle est utilisée avec une relaxation quadratique convexe. Dans un second temps, on rappelle les contractions de type RCBT qui sont présentes dans notre algorithme. On conclue avec l'algorithme de contractions OBBT mis en œuvre dans notre méthode `RC-OPF`.

3.1.1 Contraction de bornes par résolution de problèmes convexes

On présente ici la technique visant à résoudre un problème de minimisation et un problème de maximisation convexe pour chacune des variables du problème afin de contracter les bornes sur cette variable. Les problèmes considérés pour une variable v d'indice k sont présentés ci-dessous. En rouge, la contrainte ajoutée aux problèmes, nécessitant de connaitre une borne supérieure sur ceux-ci. UB est la meilleure valeur connue du problème d'origine. On remarquera que cette contrainte est quadratique convexe et fait intervenir la matrice de reformulation optimale S^*.

Problème de minimisation :

$$(MIN_k)\begin{cases} \min\limits_{v,W} \quad v_k \\ \text{s.t.} \\ \quad (2.6) - (2.10), \\ \quad (v, W) \in (MCC) \\ \quad v^t S^* v + < \mathbf{C}_1 - S^*, W > + \mathbf{c}_0 \leq UB. \end{cases} \tag{3.1}$$

Problème de maximisation :

$$(MAX_k)\begin{cases} \max\limits_{v,W} \quad v_k \\ \text{s.t.} \\ \quad (2.6) - (2.10), \\ \quad (v, W) \in (MCC) \\ \quad (3.1). \end{cases}$$

Ces deux problèmes ont une fonction objectif linéaire et des contraintes linéaire à l'exception de la contrainte quadratique convexe (3.1).

Notons que l'on peut utiliser la même technique pour contracter les bornes sur les variables W et sur les contraintes du problèmes. La seule différence se situe dans la fonction objectif à minimiser ou maximiser. On peut ajouter des bornes explicites sur les variables W pour les deux problèmes (MIN_k) et (MAX_k). On le fait dès lors que l'on choisit d'utiliser nos techniques de contraction de bornes sur les domaines des variables W.

3.1.2 Synergie avec la reformulation quadratique optimale

Ici nous analysons plus en détail la contrainte (3.1). On voit que cette contrainte fait à la fois intervenir la matrice de reformulation optimale S^* et la meilleure borne supérieure connue sur le problème UB.

Par définition de S^*, quels que soient v et W admissibles pour (MIN_k) ou (MAX_k), la relation suivante est vérifiée :

$$v^t S^* v + < \mathbf{C}_1 - S^*, W > + \mathbf{c}_0 \geq LB.$$

Où LB est la borne inférieure donnée par la relaxation du rang. On voit alors que cette contrainte fait intervenir les bornes inférieures et supérieures sur le problème, et ces bornes sont souvent proches dans le cas de l'OPF. Ainsi, la contrainte (3.1) est susceptible d'être saturée dès que les valeurs prises par les variables diffèrent trop de leur valeur à un optimum global, et donc coupe d'importantes sections du domaine des variables v. On vérifiera cette intuition dans les résultats numériques du chapitre 5.

Cette synergie entre relaxation du rang, meilleure reformulation et contrainte d'objectif inférieur à la meilleure borne supérieure connue conduit à des contractions très efficaces.

3.1.3 Contractions fondées sur les coûts réduits

Les contractions de type RCBT (pour *Reduced-Costs Bound Tightening*, ou contraction de bornes fondée sur les coûts réduits) sont aussi implémentées dans notre algorithme de *branch-and-bound* spatial. Ces contractions sont celles de l'état de l'art [1]. On les utilise lorsque l'on traite la relaxation $(\overline{OPF - L_{S^*} - MCC})$ d'un nœud, comme expliqué dans 1.2.2. On peut aussi utiliser ces contractions à chaque fois qu'une relaxation (MIN_k) ou (MAX_k) est résolue. Dans ce cas, la borne supérieure sur la relaxation UB, utilisée pour effectuer la contraction, est :

— u_k dans le cas de (MIN_k)

— $-\ell_k$ dans le cas de (MAX_k), où l'on cherche plutôt à minimiser $-v_k$.

3.1.4 Algorithme

L'algorithme **??** décrit comment les contractions OBBT sont implémentées dans `RC-OPF`. Dans cet algorithme, *bo_mod* est un booléen servant à représenter le fait que l'on a obtenu au moins une

modification dans la boucle de contraction courante. Enfin, *Nb_ite* sert à compter le nombre de boucles parcourrues, ce nombre devant rester inférieur à la limite représentée par *Nb_ite_lim*.

3.2 Contractions fondées sur la faisabilité en phase 2

Dans cette section, on présente les contractions de type FBBT (pour *Feasibility-Based Bound Tightening*, ou contraction de bornes fondée sur la faisabilité) mises en œuvre dans notre algorithme de *branch-and-bound* spatial utilisé pour résoudre $(OPF - L)$ à l'optimum global. Dans un premier temps, on explique ce qui est mis en place en terme de contractions exploitant les contraintes linéaires du problème, puis nous présentons des contractions non-linéaires que nous avons imaginées et implémentées, exploitant les contraintes non-linéaires et non convexes de $(OPF - L)$. Enfin nous donnons l'algorithme utilisé pour mettre en œuvre ces contractions dans `RC-OPF`.

3.2.1 Contractions linéaires [1]

Les contractions linéaires mises en œuvre utilisent l'arithmétique d'intervalles sur les contraintes linéaires du problème pour réduire les bornes sur les variables v. Nous n'avons pas implémenté nous-même ces techniques, nous utilisons les algorithmes disponibles dans le logiciel d'optimisation AMPL [47]. Comme les contraintes linéaires des relaxations quadratiques convexes font appel aux variables W, pour que l'arithmétique d'intervalles donne des résultats, il faut fournir des bornes sur ces variables. On utilise pour ce faire les inégalités suivantes :

$$\min(\ell_i\ell_j, \ell_i u_j, u_i\ell_j, u_i u_j) \leq W_{i,j} \leq \max(\ell_i\ell_j, \ell_i u_j, u_i\ell_j, u_i u_j).$$

Ces contractions linéaires sont moins efficaces que celles obtenues avec la résolution de problèmes d'optimisation, mais elles sont beaucoup plus rapides à calculer.

3.2.2 Contractions non-linéaires

Ces contractions originales utilisent les contraintes non-linéaires sur les modules de tension (2.5) pour réduire les bornes sur les variables de tension en chaque nœud. Elles sont bien plus rapides à calculer en chaque nœud que des relaxations quadratiques convexes. On utilise les inégalités suivantes pour réduire les bornes en tension :

$$\underline{\mathbf{v}_i}^2 \leq v_i^2 + v_{i+n}^2 \leq \overline{\mathbf{v}_i}^2$$

Algorithme 3 : Algorithme OBBT-OPF

1 Initialiser Nb_ite_lim;
2 $bo_mod \leftarrow$ **Vrai** ;
3 $Nb_ite \leftarrow 0$;
4 **tant que** bo_mod *et* $Nb_ite < Nb_ite_lim$ **faire**
5 $\quad bo_mod \leftarrow$ **Faux**;
6 $\quad Nb_ite \leftarrow Nb_ite + 1$;
7 $\quad$ **pour** $k \in \{1, \ldots, 2n\}$ **faire**
8 $\quad\quad$ Résolution de (MIN_k);
9 $\quad\quad LB_k \leftarrow val(MIN_k)$;
10 $\quad\quad UB_k \leftarrow u_k$;
11 $\quad\quad \bar{v} \leftarrow$ solution optimale de (MIN_k);
12 $\quad\quad$ <u>RCBT</u> : **pour** $i \in \{1, \ldots, 2n\}$ **faire**
13 $\quad\quad\quad rc_i \leftarrow$ coût réduit de v_i dans (MIN_k);
14 $\quad\quad\quad$ **si** $\bar{v}_i = \ell_i$ *et* $\ell_i + \frac{UB_k - LB_k}{rc_i} < u_i$ **alors**
15 $\quad\quad\quad\quad u_i \leftarrow \ell_i + \frac{UB_k - LB_k}{rc_i}$;
16 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
17 $\quad\quad\quad$ **si** $\bar{v}_i = u_i$ *et* $u_i - \frac{UB_k - LB_k}{rc_i} > \ell_i$ **alors**
18 $\quad\quad\quad\quad \ell_i \leftarrow u_i - \frac{UB_k - LB_k}{rc_i}$;
19 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
20 $\quad\quad$ **si** $LB_k > \ell_k$ **alors**
21 $\quad\quad\quad \ell_k \leftarrow LB_k$;
22 $\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
23 $\quad\quad$ Résolution de (MAX_k);
24 $\quad\quad UB_k \leftarrow val(MAX_k)$;
25 $\quad\quad LB_k \leftarrow \ell_k$;
26 $\quad\quad \bar{v} \leftarrow$ solution optimale de (MAX_k);
27 $\quad\quad$ <u>RCBT</u> : **pour** $i \in \{1, \ldots, 2n\}$ **faire**
28 $\quad\quad\quad rc_i \leftarrow$ coût réduit de v_i dans (MAX_k);
29 $\quad\quad\quad$ **si** $\bar{v}_i = \ell_i$ *et* $\ell_i + \frac{UB_k - LB_k}{rc_i} < u_i$ **alors**
30 $\quad\quad\quad\quad u_i \leftarrow \ell_i + \frac{UB_k - LB_k}{rc_i}$;
31 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
32 $\quad\quad\quad$ **si** $\bar{v}_i = u_i$ *et* $u_i - \frac{UB_k - LB_k}{rc_i} > \ell_i$ **alors**
33 $\quad\quad\quad\quad \ell_i \leftarrow u_i - \frac{UB_k - LB_k}{rc_i}$;
34 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
35 $\quad\quad$ **si** $UB_k < u_k$ **alors**
36 $\quad\quad\quad u_k \leftarrow UB_k$;
37 $\quad\quad\quad bo_mod \leftarrow$ **Vrai**;

$$\ell_i \leq v_i \leq u_i \tag{3.2}$$

$$\ell_{i+n} \leq v_{i+n} \leq u_{i+n} \tag{3.3}$$

À partir de ces trois contraintes, on peut, en comparant leurs bornes respectives, en déduire des bornes plus serrées.

Considérons que l'on cherche à resserrer la contrainte (3.3). On voit que :

$$\underline{\mathbf{v}_i}^2 - v_i^2 \leq v_{i+n}^2 \leq \overline{\mathbf{v}_i}^2 - v_i^2.$$

Donc :

$$\underline{\mathbf{v}_i}^2 - (\max(|\ell_i|, |u_i|))^2 \leq v_{i+n}^2 \quad \text{et} \quad v_{i+n}^2 \leq \overline{\mathbf{v}_i}^2 - (\min([\ell_i, u_i]))^2.$$

On en déduit que si $\underline{\mathbf{v}_i}^2 \geq (\max(|\ell_i|, |u_i|))^2$ alors :

$$v_{i+n} \leq -\sqrt{\underline{\mathbf{v}_i}^2 - (\max(|\ell_i|, |u_i|))^2} \quad \text{ou} \quad \sqrt{\underline{\mathbf{v}_i}^2 - (\max(|\ell_i|, |u_i|))^2} \leq v_{i+n}.$$

Et si $\overline{\mathbf{v}_i}^2 \geq (\min([\ell_i, u_i]))^2$ alors :

$$-\sqrt{\overline{\mathbf{v}_i}^2 - (\min([\ell_i, u_i]))^2} \leq v_{i+n} \leq \sqrt{\overline{\mathbf{v}_i}^2 - (\min([\ell_i, u_i]))^2}.$$

Ces inégalités fournissent éventuellement de meilleures bornes sur v_{i+n}. On peut procéder de même pour la contrainte (3.2).

Tous ces résultats s'obtiennent également en faisant un dessin de l'anneau contenant les variables v_i et v_{i+n}. La Figure 3.1 représente cet anneau. La Figure 3.2 montre un exemple de situation dans laquelle on peut resserer la contrainte 3.3 en augmentant la valeur de ℓ_{i+n}. Dans cet exemple, l'intérieur du rectangle orange correspond à l'espace définit par les bornes sur les variables v_i et v_{i+n}.

3.2.3 Algorithme

L'algorithme **??** décrit comment les contractions FBBT sont implémentées dans `RC-OPF`. Les variables informatiques *bo_mod*, *Nb_ite* et *Nb_ite_lim* jouent le même rôle dans cet algorithme que dans l'algorithme **??**.

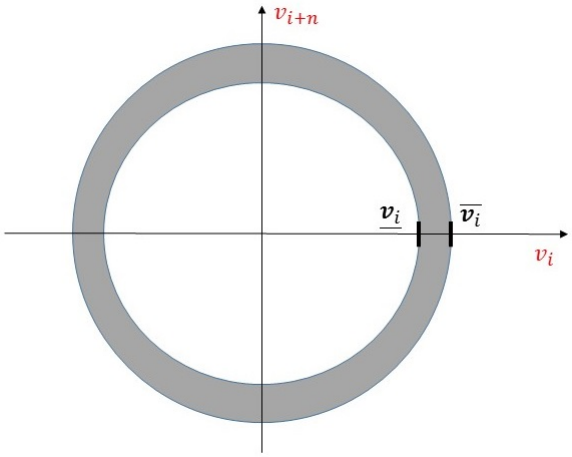

FIGURE 3.1 – Anneau contenant les variables v_i et v_{i+n}

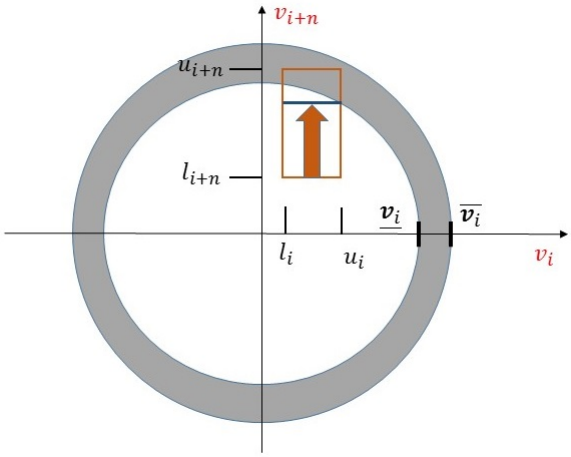

FIGURE 3.2 – Exemple de réduction de bornes avec les réductions non linéaires

Algorithme 4 : Algorithme FBBT-OPF

1 Initialiser Nb_ite_lim ;
2 $bo_mod \leftarrow$ **Vrai** ;
3 $Nb_ite \leftarrow 0$;
4 **tant que** bo_mod *et* $Nb_ite < Nb_ite_lim$ **faire**
5 $\quad bo_mod \leftarrow$ **Faux**;
6 $\quad Nb_ite \leftarrow Nb_ite + 1$;
7 $\quad$ **pour** $k \in \{1, \ldots, 2n\}$ **faire**
8 $\quad\quad$ **si** *borne inf AMPL* `v[k].lb2` *supérieure à* ℓ_k **alors**
9 $\quad\quad\quad \ell_k \leftarrow$ `v[k].lb2`;
10 $\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
11 $\quad\quad$ **si** *borne sup AMPL* `v[k].ub2` *est inférieure à* u_k **alors**
12 $\quad\quad\quad u_k \leftarrow$ `v[k].ub2`;
13 $\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
14 $\quad$ **pour** $k \in \{1, \ldots, 2n\}$ **faire**
15 $\quad\quad$ (dans la suite $k + n$ doit s'interpréter comme $k + n$ modulo $2n$);
16 $\quad\quad$ **si** $\ell_k > 0$ **alors**
17 $\quad\quad\quad$ **si** $-\sqrt{\overline{\boldsymbol{v}_k}^2 - \ell_k^2} > \ell_{k+n}$ **alors**
18 $\quad\quad\quad\quad \ell_{k+n} \leftarrow -\sqrt{\overline{\mathbf{v}_k}^2 - \ell_k^2}$;
19 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
20 $\quad\quad\quad$ **si** $\sqrt{\overline{\boldsymbol{v}_k}^2 - \ell_k^2} < u_{k+n}$ **alors**
21 $\quad\quad\quad\quad u_{k+n} \leftarrow \sqrt{\overline{\mathbf{v}_k}^2 - \ell_k^2}$;
22 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
23 $\quad\quad$ **si** $u_k < 0$ **alors**
24 $\quad\quad\quad$ **si** $-\sqrt{\overline{\boldsymbol{v}_k}^2 - u_k^2} > \ell_{k+n}$ **alors**
25 $\quad\quad\quad\quad \ell_{k+n} \leftarrow -\sqrt{\overline{\mathbf{v}_k}^2 - u_k^2}$;
26 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
27 $\quad\quad\quad$ **si** $\sqrt{\overline{\boldsymbol{v}_k}^2 - u_k^2} < u_{k+n}$ **alors**
28 $\quad\quad\quad\quad u_{k+n} \leftarrow \sqrt{\overline{\mathbf{v}_k}^2 - u_k^2}$;
29 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
30 $\quad\quad$ **si** $max(-\ell_k, u_k) < \underline{\boldsymbol{v}_k}$ **alors**
31 $\quad\quad\quad x_k \leftarrow \sqrt{\underline{\mathbf{v}_k}^2 - (\max(|\ell_k|, |u_k|))^2}$;
32 $\quad\quad\quad$ **si** $-x_k \leq \ell_{k+n}$ *et* $\ell_{k+n} < x_k$ *et* $x_k \leq u_{k+n}$ **alors**
33 $\quad\quad\quad\quad \ell_{k+n} \leftarrow x_k$;
34 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;
35 $\quad\quad\quad$ **si** $\ell_{k+n} \leq -x_k$ *et* $-x_k < u_{k+n}$ *et* $u_{k+n} \leq x_k$ **alors**
36 $\quad\quad\quad\quad u_{k+n} \leftarrow -x_k$;
37 $\quad\quad\quad\quad bo_mod \leftarrow$ **Vrai**;

3.3 Extension des contractions de bornes à la résolution de $(OPF - Q)$

Dans cette section nous allons voir comment intégrer les contractions vu ci-dessus à la résolution de la variante $(OPF - Q)$. Il faut noter que comme dans le cas du problème $(OPF - L)$, on ne cherche qu'à contracter les bornes sur les variables de tension v. On ne cherche pas à contracter les variables de puissance p. Les contraintes de type FBBT et RCBT s'intègrent de la même manière et on utilise le même algorithme de contractions dans les variantes $(OPF - L)$ et $(OPF - Q)$. La seule différence se fait dans le traitement des contractions par résolution de problèmes convexes.

Contractions par résolution de problèmes convexes

Ici on reprend également la même idée que dans le cas de $(OPF - L)$ mais les problèmes (MIN_k) et (MAX_k) prennent la forme suivante :

$$(MIN_k - Q)\begin{cases} \min\limits_{v,p,Z} \quad v_k \\ \text{s.t.} \\ \quad (8), (2.6), (2.7), (2.9), (2.10), (2.44) \\ \quad (v, W) \in (MCC) \\ \quad\quad \mathbf{c}_1^t p + < \mathbf{C}_2, N > + \begin{bmatrix} v \\ p \end{bmatrix}^t S^* \begin{bmatrix} v \\ p \end{bmatrix} - < S^*, Z > + \mathbf{c}_0^q \leq UB. \end{cases} \tag{3.4}$$

$$(MAX_k - Q)\begin{cases} \max\limits_{v,p,Z} \quad v_k \\ \text{s.t.} \\ \quad (8), (2.6), (2.7), (2.9), (2.10), (2.44) \\ \quad (v, W) \in (MCC) \\ \quad (3.4). \end{cases}$$

Chapitre 4

Liens avec la méthode de la hiérarchie de Lasserre

Contenu

4.1 Introduction **131**
4.2 Bref rappel de la Hiérarchie de Lasserre **132**
4.3 Reformulation et relaxation lagrangienne **134**
4.3.1 Reformulation d'ordre d 134
4.3.2 Relaxation lagrangienne 136
4.3.3 Conditions pour une absence de saut de dualité 137
4.3.4 Équivalence primale-duale 138
4.4 Principal résultat **138**
4.4.1 Reformulation de (P_d^2) 138
4.4.2 Le résultat 139
4.5 Conséquences **139**
4.5.1 Reformulation « optimale » 139
4.5.2 Reformulation *Sum of Squares* 140

Dans ce chapitre, nous explorons le lien entre la hiérarchie de Lasserre et notre méthode de reformulation optimale `RC-OPF`. `RC-OPF` utilise l'ordre 1 de la hiérarchie de Lasserre, donc la relaxation du rang, pour reformuler de manière optimale des QCQP. Nous nous demandons dans ce chapitre si l'on peut reformuler à partir des ordres suivants de la hiérarchie de Lasserre. Nous cherchons également à généraliser à une classe de problèmes plus larges que les QCQP cette notion de reformulation optimale. Nous donnons des éléments de réponses à ces interrogations dans le cadre de l'optimisation polynomiale : à partir de chaque ordre de la hiérarchie de Lasserre, il est possible de construire une reformulation optimale.

Notre méthode consiste, une fois que l'on s'est donné un ordre d dans la hiérarchie de Lasserre, à reformuler de manière linéaire et équivalente un problème polynomial tout en conservant une famille de contraintes quadratiques assurant l'équivalence de la reformulation. Puis on transforme en contraintes SDP les contraintes d'inégalités du problème d'origine, de manière similaire à ce qui est fait dans l'ordre d de la hiérarchie de Lasserre. Enfin on ajoute un terme quadratique convexe dans la fonction objectif comme on l'a fait dans le cadre de notre méthode `RC-OPF`. En fonction du terme que l'on ajoute dans la fonction objectif, on obtient différentes reformulations, toute relative à l'ordre d de la hiérarchie de Lasserre. Notre méthode permet de construire la reformulation dite « optimale » gràce la résolution de l'ordre d de cette hiérarchie.

La première section de ce chapitre introduit notre problématique plus précisèment, puis dans la deuxième section nous rappelons la forme prise par les problèmes primal et dual d'un ordre donné dans la hiérarchie de Lasserre. La troisième section présente une famille de reformulations associée à chaque ordre de la hiérarchie de Lasserre. Cette famille est composée de reformulations ayant une fonction objectif quadratique, des contraintes linéaires, des contraintes SDP et une famille de contraintes quadratiques non-convexes. La quatrième section contient le cœur de ce chapitre, à savoir le résultat caractérisant la reformulation optimale au sein de notre famille de reformulations, pour chaque ordre de la hiérarchie de Lasserre. Cette reformulation est optimale dans le sens où elle maximise la borne inférieure obtenue en relaxant la famille de contraintes quadratiques non-convexes. On montre aussi que cette borne obtenue par relaxation d'une reformulation optimale a même valeur que celle de l'ordre de la hiérarchie de Lasserre dont elle provient. La cinquième et dernière section résume ces résultats.

4.1 Introduction

Notre objectif dans ce chapitre est de résoudre exactement le problème d'optimisation polynomial (P) :

$$(P)\begin{cases} \min\limits_{x} \quad f_0(x) \\ \text{s.t.} \\ \quad f_k(x) + b_k \geq 0, \quad k = 1, .., m \end{cases} \tag{4.1}$$

Où m est le nombre de contraintes du problème. f_k pour $k = 0, .., m$ sont des polynômes sur $\mathbb{R}^n, n \in \mathbb{N}^*$. Ces polynômes n'ont pas de termes constants, cette hypothèse n'est pas restrictive. Les b_k pour $k = 1, .., m$ sont des réels. L'objectif et les contraintes sont *a priori* non convexes.

La méthode `RC-OPF`, s'appuie sur une relaxation semi-définie positive (SDP) pour reformuler puis résoudre des problèmes quadratiques. Cette relaxation correspond à l'ordre 1 de la hiérarchie de Lasserre [39]. Notre idée est d'appliquer une philosophie analogue à celle de `RC-OPF` pour reformuler puis résoudre des problèmes polynômiaux à l'aide de n'importe quel ordre d de la hiérarchie de Lasserre.

Dans la suite on dira que deux problèmes d'optimisation (A) et (B) sont équivalents et on notera $(A) \Leftrightarrow (B)$ si (A) et (B) ont même valeur optimale et si leurs solutions optimales sont liées. C'est-à-dire que l'on sait exprimer les solutions optimales de l'un à partir des solutions optimales de l'autre.

Soit d un entier. L'objectif est de construire un problème (P^*) reformulation de (P) tel que :

— $(P^*) \Leftrightarrow (P)$,

— Le calcul d'une borne inférieure de (P^*) soit la résolution d'un problème convexe.

— La borne inférieure obtenue ait au moins la valeur de la relaxation de Lasserre d'ordre d de (P).

Pour ce faire nous allons à la fois raisonner par dualité comme dans la Section (2.2), et introduire les monômes du problème comme dans la hiérarchie de Lasserre.

4.2 Bref rappel de la Hiérarchie de Lasserre

Dans cette section on réintroduit briévement la hiérarchie de Lasserre. Voir la Section (1.2.1) pour plus de détails.

Soit f une fonction polynômiale, on note $d(f)$ son degré. On note $d(P)$ le degré du problème (P). C'est le plus haut degré parmi ceux des polynômes $f_k, k = 0, .., m$ caractérisant (P).

Pour un polynôme f, on note f_α son coefficient relatif au monôme α de f. Il en suit que $f(x) = \sum_{\alpha \in \Gamma_n^{d(P)}} f_\alpha x^\alpha$. En utilisant ces notations, on écrit (P) :

$$(P) \begin{cases} \min\limits_{x} \quad \sum\limits_{\alpha \in \Gamma_n^{d(P)}} f_{0,\alpha} x^\alpha \\ \text{s.t.} \\ \quad \sum\limits_{\alpha \in \Gamma_n^{d(P)}} f_{k,\alpha} x^\alpha + b_k \geq 0, \quad k = 1, .., m \end{cases} \tag{4.2}$$

On rappelle la relaxation de Lasserre d'ordre d de (P) qui s'écrit :

$$(L_d - P) \begin{cases} \min\limits_{y} \quad f_{L_d - P}(y) = \sum\limits_{\alpha \in \Gamma_n^{2d}} f_{0,\alpha} y_\alpha \\ \text{s.t.} \\ \quad \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha B_{k,\alpha}^d + b_k B_{k,0}^d \succeq 0, \quad k = 1, .., m & (4.3) \\ \quad \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha B_{0,\alpha}^d + B_{0,0}^d \succeq 0. & (4.4) \end{cases}$$

Exemple 1. *Au cours de ce chapitre on prend le problème suivant comme exemple :*

$$(EX) \begin{cases} \min\limits_{x_1, x_2} \quad x_1 + x_2 \\ s.t. \\ \quad 2x_1 + 3x_2 - 1 \geq 0, & (4.5) \\ \quad x_1^2 - x_1 = 0, & (4.6) \\ \quad x_2^2 - x_2 = 0, & (4.7) \end{cases}$$

C'est un programme linéaire en variables 0-1. En effet les deux dernières contraintes modélisent le fait que les deux variables du problème valent 0 ou 1.

Illustrons la hiérarchie de Lasserre avec notre problème (EX). Commençons avec le plus simple, l'ordre 1. Celui-ci coïncide avec la fameuse relaxation du rang.

Ordre 1 : Ici on introduit les variables y_α relatives aux monômes d'ordre 1 et deux, c'est-à-dire : $y_{10}, y_{01}, y_{20}, y_{02}, y_{11}$. La nouvelle fonction objectif linéarisée devient $y_{10} + y_{01}$. Les d'_k relatifs aux contraintes sont tous nuls, on linéarise uniquement les contraintes. Notons qu'elles sont déjà écrites comme combinaison linéaire des monômes. Il ne reste qu'à ajouter la semi-défine positivité de $\mathcal{M}_1^2(y)$. L'ordre 1 est alors :

$$(L_1 - EX)\begin{cases} \min\limits_{y} \quad y_{10} + y_{01} \\ s.t. \\ \quad 2y_{10} + 3y_{01} - 1 \geq 0, & (4.8) \\ \quad y_{20} - y_{10} = 0, & (4.9) \\ \quad y_{02} - y_{01} = 0, & (4.10) \\ \quad \begin{pmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix} \succeq 0. & (4.11) \end{cases}$$

La décomposition en combinaison linéaire de $\mathcal{M}_1^2(y)$ est alors :

$$\begin{pmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + y_{10}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + y_{01}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} +$$

$$y_{20}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + y_{02}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + y_{11}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$= B^1_{0,00} + y_{10}B^1_{0,10} + y_{01}B^1_{0,01} + y_{20}B^1_{0,20} + y_{02}B^1_{0,02} + y_{11}B^1_{0,11}.$$

La relaxation d'ordre 1 $(L_1 - EX)$ a pour valeur $\frac{1}{3}$. La valeur optimale du problème (EX) est 1. C'est donc une relaxation non exacte de (EX).

Ordre 2 : Ici on introduit les variables y_α relatives aux monômes d'ordre 1, 2, 3 et 4, c'est-à-dire : $y_{10}, y_{01}, y_{20}, y_{02}, y_{11}, y_{30}, y_{03}, y_{21}, y_{12}, y_{40}, y_{04}, y_{31}, y_{13}, y_{22}$. La nouvelle fonction objectif linéarisée reste $y_{10} + y_{01}$. Les d'_k relatifs aux contraintes valent tous 1, on linéarise les contraintes puis on les multiplie

par $\mathcal{M}_1^2(y)$. Il ne reste qu'à ajouter la semi-défine positivité de $\mathcal{M}_2^2(y)$. On obtient le problème suivant :

$$(L_2 - EX)\begin{cases} \min\limits_{y} \quad y_{10} + y_{01} \\ s.t. \\ \begin{pmatrix} 2y_{10}+3y_{01}-1 & 2y_{20}+3y_{11}-y_{10} & 2y_{11}+3y_{02}-y_{01} \\ 2y_{20}+3y_{11}-y_{10} & 2y_{30}+3y_{21}-y_{20} & 2y_{21}+3y_{12}-y_{11} \\ 2y_{11}+3y_{02}-y_{01} & 2y_{21}+3y_{12}-y_{11} & 2y_{12}+3y_{03}-y_{02} \end{pmatrix} \succeq 0, & (4.12) \\ \begin{pmatrix} y_{20}-y_{10} & y_{30}-y_{20} & y_{21}-y_{11} \\ y_{30}-y_{20} & y_{40}-y_{30} & y_{31}-y_{21} \\ y_{21}-y_{11} & y_{31}-y_{21} & y_{22}-y_{12} \end{pmatrix} = 0, & (4.13) \\ \begin{pmatrix} y_{02}-y_{01} & y_{12}-y_{11} & y_{03}-y_{02} \\ y_{12}-y_{11} & y_{22}-y_{21} & y_{13}-y_{12} \\ y_{03}-y_{02} & y_{13}-y_{12} & y_{04}-y_{03} \end{pmatrix} = 0, & (4.14) \\ \begin{pmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{02} & y_{11} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{12} & y_{21} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{03} & y_{12} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{22} & y_{31} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{04} & y_{13} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{13} & y_{22} \end{pmatrix} \succeq 0 & (4.15) \end{cases}$$

Pour l'ordre 2, la valeur est de 0.9311. C'est donc aussi une relaxation non exacte.

Le dual de l'ordre d s'écrit :

$$(L_d - D)\begin{cases} \max\limits_{Z,S} \quad g_{L_d-D}(Z,S) = -<S, B_{0,0}^d> - \sum\limits_{k=1}^{m} b_k <Z_k, B_{k,0}^d> \\ \text{s.t.} \\ f_{0,\alpha} = \sum\limits_{k=1}^{m} <Z_k, B_{k,\alpha}^d> + <S, B_{0,\alpha}^d>, \quad \forall \alpha \in \Gamma_n^{2d} & (4.16) \\ Z_k \succeq 0, & (4.17) \\ S \succeq 0. & (4.18) \end{cases}$$

Z_k est la matrice de variables duales associée à la contrainte (4.3). S est la matrice de variables duales associée à la contrainte de moment d'ordre d (4.4).

4.3 Reformulation et relaxation lagrangienne

4.3.1 Reformulation d'ordre d

Soit d un entier naturel tel que $2d \geq d(P)$, nous construisons une reformulation d'ordre d du problème. Elle est liée à l'ordre d de la hiérarchie de Lasserre. Pour construire cette reformulation d'un problème d'optimisation polynomial, on introduit pour chaque monôme la variable y_α correspondante.

On linéarise l'objectif et les contraintes en remplaçant les monômes x^α par y_α. Cependant, pour reformuler de manière équivalente il faut conserver le fait que les y_α sont bien des produits de variables x. Pour cela on ajoute une variable z_α pour chaque monôme de degré d ou moins, ainsi que pour le monôme nul correspondant au polynôme constant de valeur 1. On ajoute alors la contrainte suivante au problème :

$$zz^t = \sum_{\alpha \in \Gamma_n^{2d}} y_\alpha B_{0,\alpha}^d + B_{0,0}^d \tag{4.19}$$

On reformule (P) :

$$(P_d) \begin{cases} \min\limits_{y,z} \quad f_{L_d - P}(y) \\ \text{s.t.} \\ \quad \sum\limits_{\alpha \in \Gamma_n^{2d}} f_{k,\alpha} y_\alpha + b_k \geq 0, \quad k = 1, .., m \\ \quad zz^t = \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha B_{0,\alpha}^d + B_{0,0}^d \qquad (4.19). \end{cases} \tag{4.20}$$

Proposition 9. $(P) \Leftrightarrow (P_d)$.

Preuve. Toute solution x^* de (P) fournit une solution de (P_d) de même valeur en posant $y_\alpha = (x^*)^\alpha$ et $z_\alpha = (x^*)^\alpha$.

Soit y^* et z^* une solution de (P_d), on prend x égal au vecteur constitué des coordonnées des y^* censés représenter les monômes d'ordre 1 en x. La contrainte (4.19) entraine par récurrence sur le degré des monômes α que $x^\alpha = y_\alpha^*$. Ainsi x est une solution de (P) de même valeur. □

Maintenant on linéarise les contraintes polynomiales grâce aux variables y qui représentent les produits entre variables z. On les modifier comme cela est fait dans le cas de la relaxation SDP de l'ordre d de la hiérarchie. Cette modification est équivalente ici grâce à la présence de la contraite (4.19).

De ces nouvelles contraintes, on tire la reformulation suivante :

$$(P_d^2)\begin{cases} \min\limits_{y,z} \quad f_{L_d-P}(y) \\ \text{s.t.} \\ \quad \sum\limits_{\alpha\in\Gamma_n^{2d}} y_\alpha B_{k,\alpha}^d + b_k B_{k,0}^d \succeq 0, \quad k=1,..,m & (4.3) \\ \quad zz^t = \sum\limits_{\alpha\in\Gamma_n^{2d}} y_\alpha B_{0,\alpha}^d + B_{0,0}^d & (4.19) \end{cases}$$

Proposition 10. *La reformulation (P_d^2) est équivalente à (P_d).*

Preuve. Soit (z^*, y^*) une solution de (P_d), alors comme vu ci-dessus :
$\forall k = 1,..,m, \sum_{\alpha\in\Gamma_n^{2d}} y_\alpha^* B_{k,\alpha}^d + b_k B_{k,0}^d \succeq 0$. Ainsi (z^*, y^*) est une solution de (P_d^2).

Soit (z^*, y^*) une solution de (P_d^2), alors $\forall k = 1,..,m, \sum_\alpha y_\alpha^* B_{k,\alpha}^d + b_k B_{k,0}^d \succeq 0$. Ce qui implique que les coefficients diagonaux de ces matrices sont tous positifs. Or leur premier coefficient diagonal est : $\sum_\alpha f_{k,\alpha} y_\alpha^* + b_k$. Ainsi (z^*, y^*) est une solution de (P_d) de même valeur. □

La première étape de reformulation est maintenant terminée, elle nous a fait linéariser les contraintes et l'objectif du problème selon les monômes d'un ordre d de reformulation, de plus les contraintes d'origine ont été transformées en des inégalités linéaires matricielles qui sont plus fortes. La seconde étape consiste à étudier la relaxation lagrangienne de cette reformulation.

4.3.2 Relaxation lagrangienne

Soit :

$$\begin{aligned} L(z,y,Z,S) = \textstyle\sum_{\alpha\in\Gamma_n^{2d}} f_{0,\alpha} y_\alpha - \sum_{k=1}^m < Z_k, \sum_{\alpha\in\Gamma_n^{2d}} y_\alpha B_{k,\alpha}^d + b_k B_{k,0}^d > \\ + < S, zz^t - \textstyle\sum_{\alpha\in\Gamma_n^{2d}} y_\alpha B_{0,\alpha}^d - B_{0,0}^d > . \end{aligned}$$

L est le lagrangien de (P_d^2), et :

$$(P_d^2) \Leftrightarrow \min_{z,y} \max_{Z\succeq 0, S} L(z,y,Z,S).$$

Le dual lagrangien de (P_d^2) est alors :

$$\max_{Z\succeq 0, S} \min_{z,y} L(z,y,Z,S)$$

On le note $(P_d - DL)$. C'est la relaxation lagrangienne de (P_d^2).

$$(P_d - DL) \begin{cases} \max\limits_{Z,S} & g_{L_d - D}(Z, S) \\ \text{s.t.} & \\ & (4.16) - (4.18). \end{cases}$$

On retrouve ici le dual de l'ordre d dans la hiérarchie de Lasserre. C'est un programme SDP convexe.

Le dual de $(P_d - DL)$, donc le bidual de (P_d^2) est alors le primal de l'ordre d dans la hiérarchie de Lasserre. C'est le programme SDP convexe suivant :

$$(P_d - PL) \begin{cases} \min\limits_{y} & f_{L_d - P}(y) \\ \text{s.t.} & \\ & (4.3), (4.4). \end{cases}$$

On va maintenant s'intéresser aux conditions assurant une absence de saut de dualité entre le dual et le primal de l'ordre d de la hiérarchie de Lasserre.

4.3.3 Conditions pour une absence de saut de dualité

Comme le dual et le primal de l'ordre d de la hiérarchie de Lasserre sont des programmes SDP, il y a absence de saut de dualité dès lors que les conditions de Slater sont vérifiées.

Si le problème initial (P) est réalisable, alors $(P_d - PL)$ est réalisable. De plus si l'on dispose d'une contrainte de boule dans (P) [1] et que cette contrainte a été portée à l'ordre maximal (i.e. le d_k de cette contrainte vaut : $\frac{2d - d(P_k)}{2}$), alors le problème dual $(P_d - DL)$ est strictement réalisable [2]. Ces deux conditions suffisent pour assurer l'absence de saut de dualité.

Dans la suite, nous avons besoin de la validité de la proposition ci-dessus. Nous nous limitons donc aux problèmes polynomiaux disposant d'une contrainte de boule. Notons que c'est le dans le problème

1. Une contrainte de boule est une contrainte de la forme $\sum_i x_i^2 \leq B$.
2. Voir [39] et la preuve de la Section (2.2.2)

OPF. On notera aussi qu'il est toujours possible d'ajouter une contrainte de boule dans un problème où l'on connait des bornes sur chacune des variables. Cette hypothèse n'est donc pas restrictive pour un grand nombre de problèmes appliqués. On s'intéresse maintenant aux différents problèmes équivalents en l'absence de saut de dualité.

4.3.4 Équivalence primale-duale

Si les conditions de Slater sont vérifiées (vrai dans le cas de l'OPF), il y a équivalence entre $(P_d - PL)$ et $(P_d - DL)$. De plus le problème suivant est aussi équivalent au dual lagrangien de (P_d^2) :

$$(P_d - Eq) \begin{cases} \max\limits_{S \succeq 0} \quad \min\limits_{z,y} \quad f_S^{Eq}(z,y) = z^t S z + \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha (f_{0,\alpha} - < S, B_{0,\alpha}^d >) - < S, B_{0,0}^d > \\ \qquad \text{s.t.} \\ \qquad\qquad (4.3) \end{cases}$$

4.4 Principal résultat

4.4.1 Reformulation de (P_d^2)

Soit S une matrice semi-définie positive de taille $(|d,n|+1)^2$. On peut introduire cette matrice dans l'objectif et reformuler ainsi (P_d^2) :

$$(P_d - S) \begin{cases} \min\limits_{z,y} \quad f_S^{Eq}(z,y) \\ \quad \text{s.t.} \\ \qquad (4.3), (4.19). \end{cases}$$

Proposition 11. $\forall S \succeq 0, (P_d - S) \Leftrightarrow (P_d^2)$.

Preuve. Comme la contrainte (4.19) est vérifiée, l'objectif a même valeur en tout point (y,z) dans les deux problèmes. □

L'objectif de $(P_d - S)$ est quadratique convexe en les variables. Les contraintes sont convexes hormis la contrainte (4.19). Si on la relaxe, on obtient une relaxation convexe de (P). $(P_d - S)$ est analogue à la reformulation quadratique dans `RC-OPF`. Pour $d = 1$ sur un problème quadratique sans composantes

linéaires (comme $(OPF - L)$), on retrouve exactement la reformulation présentée en Section (2.2.3). Le choix de $S = 0$ revient à une linéarisation totale (l'objectif est lui aussi linéaire).

Pour résoudre (P) on se propose de résoudre $(P_d - S)$ dans un branch-and-bound où la borne inférieure calculée en chaque noeud est obtenue par la relaxation de la contrainte non convexe (4.19).

La borne inférieure, dépendante de S à la racine du branch-and-bound est donc la valeur du problème :

$$(QR_d - S) \begin{cases} \min\limits_{z,y} & f_S^{Eq}(z,y) \\ \text{s.t.} & \\ & (4.3). \end{cases}$$

4.4.2 Le résultat

Voici le principal résultat de cette étude :

Proposition 12.

$$\max_{S \succeq 0} \quad val(QR_d - S) \Leftrightarrow (P_d - DL)$$

De plus le S optimal peut être pris comme valeur optimale de la variable S dans $(P_d - DL)$.

Preuve. Les calculs, et notamment le dernier résultat, de la Section 4.3.4 prouvent ce résultat. □

4.5 Conséquences

4.5.1 Reformulation « optimale »

Les problèmes $(P_d - S)$ pour S une matrice SDP, constitue une famille de reformulations de (P). En prenant S égale à la solution duale de la relaxation de Lasserre d'ordre d, S^*, on obtient un problème $(P_d - S^*)$ dont la relaxation convexe $(QR_d - S^*)$ a la plus grande valeur parmi les relaxations convexes des problèmes de notre famille de reformulation.

On rappelle l'ambition de départ : construire un problème (P^*) reformulation de (P) tel que :

— $(P^*) \Leftrightarrow (P)$,

— Le calcul d'une borne inférieure de (P^*) soit la résolution d'un problème convexe.

— La borne inférieure obtenue ait au moins la valeur de la relaxation de lasserre d'ordre d de (P).

En choisissant $(P^*) = (P_d - S^*)$ on s'assure de répondre à cette ambition.

On notera que l'on n'est pas obligé de monter à l'ordre maximal toutes les contraintes du problèmes dans la reformulation d'ordre d. Pour assurer l'absence de saut de dualité, il est suffisant de passer la contrainte de boule à cet ordre maximal. Mais les autres contraintes peuvent être traitées différemment. On peut alors obtenir le même résultat d'équivalence entre une relaxation de Lasserre d'ordre d partiel et une reformulation optimale. Il faut cependant s'attendre à avoir une borne inférieure de moins bonne qualité au noeud racine.

Exemple 2. *Pour l'ordre 1, la matrice S^* obtenue par résolution du dual de $(L_1 - EX)$ est :* $\begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{pmatrix}$. *La reformulation obtenue est alors :*

$$(EX_1 - R) \begin{cases} \min\limits_{z,y} & \frac{1}{3}z_{10}^2 + y_{10} + y_{01} - \frac{1}{3}y_{20} \\ \quad s.t. & \\ & (4.8) - (4.10), \\ & \begin{pmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix} = \begin{pmatrix} z_{00}^2 & z_{00}z_{10} & z_{00}z_{01} \\ z_{00}z_{10} & z_{10}^2 & z_{10}z_{01} \\ z_{00}z_{01} & z_{10}z_{01} & z_{01}^2 \end{pmatrix}. \end{cases} \tag{4.21}$$

La borne inférieure obtenue à la racine est la valeur de :

$$(EX_1 - RR) \begin{cases} \min\limits_{z,y} & \frac{1}{3}z_{10}^2 + y_{10} + y_{01} - \frac{1}{3}y_{20} \\ \quad s.t. & \\ & (4.8) - (4.10). \end{cases}$$

On retouve alors $\frac{1}{3}$.

4.5.2 Reformulation *Sum of Squares*

Une autre catégorie de reformulations peut être envisagée. Au lieu de linéariser les monômes de degré d ou moins avec l'introduction des variables z, on peut conserver le vecteur $x_d \in \mathbb{R}^{|d,n|}$ dont les

coordonnées sont les monômes sur x. De même on construit la reformulation suivante de (P) :

$$(PSS^d - S)\begin{cases} \min\limits_{x,y} \quad x_d^t S x_d + \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha (f_{0,\alpha} - < S, B_{0,\alpha}^d >) - < S, B_{0,0}^d > \\ \quad \text{s.t.} \\ \qquad (4.2), \\ \qquad \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha B_{0,\alpha}^d + B_{0,0}^d = x_d x_d^t. \end{cases} \tag{4.22}$$

Le lien qui lie x et x_d est non linéaire et aussi non convexe dans le cas général [3]. Cette reformulation a donc un objectif qui dans le cas général est non convexe en x. cependant le terme $x_d^T S x_d$ peut se réécrire comme un polynôme en x. Un polynôme particulier puisque c'est une somme de carré de polynômes $(Q_i)_{i=1..p}$ (d'où le nom de reformulation *Sum of Squares*).

Une reformulation équivalente utilisant le polynôme *Sum of Squares* serait :

$$(PQ^d - S)\begin{cases} \min\limits_{x,y} \quad \sum\limits_{i=1}^{p} Q_i^2(x) + \sum\limits_{\alpha \in \Gamma_n^{2d}} y_\alpha (f_{0,\alpha} - < S, B_{0,\alpha}^d >) - < S, B_{0,0}^d > \\ \quad \text{s.t.} \\ \qquad (4.2), (4.22). \end{cases}$$

3. La fonction qui associe à x_1, x_2 le produit $x_1 x_2$ est non convexe sur $\mathbb{R}^2$.

Chapitre 5

Expérimentations numériques avec RC-OPF

Contenu

5.1 Caractéristiques des instances étudiées 144

5.2 Évaluation de RC-OPF sur des instances $(OPF - L)$ 146

5.2.1 Comparaison numérique de RC-OPF sans contractions de bornes avec le logiciel BARON 146

5.2.2 Impact des contractions de bornes 148

5.2.3 Impact du choix du paramètre δ 149

5.2.4 Résultats de RC-OPF avec contractions 151

5.3 Résultats obtenus avec RC-OPF-Q sur $(OPF - Q)$ 153

5.4 Génération d'instances difficiles 157

Dans ce chapitre, on présente plusieurs expérimentations numériques. Dans un premier temps, on donne les caractéristiques des différentes instances d'OPF étudiées, en particulier, les instances que l'on considère proviennent de 3 bibliothèques : MATPOWER [3], Bukhsh [2] et PG-LIB [4]. Ensuite, nous évaluons notre méthode `RC-OPF` sur des instances de la classe $(OPF - L)$. Pour cela, nous comparons ses performances avec celles du logiciel BARON 17.3.1 [40] qui est un logiciel géénrique capacble de résoudre des instances de problèmes non-linéaires en variables mixtes. Nous effectuons également une étude détaillée de l'impact du choix du paramètre δ (qui détermine la valeur d'un branchement pour la variable sélectionnée) sur `RC-OPF`. Nous terminons l'évaluation expérimentale de `RC-OPF` pour la classe $(OPF - L)$ par l'analyse de l'effet produit par notre algorithme de contractions de bornes. Nous présentons ensuite les résultats numériques pour notre algorithme `RC-OPF-Q` sur des instances de la classe $(OPF - Q)$ Pour évaluer nos résultats, nous les mettons en perspective avec ceux de l'état de l'art sur la résolution exacte de l'OPF. Enfin, dans une dernière partie, nous présentons une méthodologie générique pour constuire de nouveaux problèmes d'OPF. Nous l'utiliserons pour contruire des instances de la classe $(OPF - L)$ à partir d'instances de MATPOWER. Nous présenterons ensuite les résultats obtenus par `RC-OPF` sur ces instances ainsi qu'une analyse des choix de branchement pour notre algorithme.

5.1 Caractéristiques des instances étudiées

Dans ce chapitre, nous étudions plusieurs instances d'OPF pour nos expérimentations numériques. Ces instances proviennent de trois bibliothèques : MATPOWER [3], celle de Bukhsh [2] et enfin PG-LIB [4]. Chacune de ces instances est disponible sous forme d'un fichier écrit au format MATPOWER décrivant le réseau électrique sous-jacent. Les coûts de production en chaque nœud producteur sont exprimés comme un polynôme de degré 2. En ne gardant que le coût linéaire on construit une instance $(OPF - L)$, sinon on construit une instance $(OPF - Q)$.

La Table **??** donne les caractéristiques de ces instances. Chaque ligne de ce tableau correspond à une instance d'OPF. Dans la première colonne on trouve le nom de l'instance. La seconde colonne indique dans quelle bibliothèque on peut trouver cette instance. La troisième colonne donne le nombre de nœuds n dans le réseau électrique pour cette instance. La quatrième donne le nombre de nœuds n_p reliés à des producteurs. Enfin, dans la dernière colonne se trouve le nombre d'arêtes dans le graphe $\mathcal{G}$ représentant le réseau électrique de l'instance.

Nom	Bibliothèque	n	n_p	*Nombre arêtes*
WB2	Bukhsh	2	1	1
LMBM3	Bukhsh	3	3	3
WB3	Bukhsh	3	1	2
WB5	Bukhsh	5	2	6
9mod	Bukhsh	9	3	9
22loop	Bukhsh	22	11	22
39mod1	Bukhsh	39	10	46
39mod2	Bukhsh	39	10	46
118mod	Bukhsh	118	54	186
300mod	Bukhsh	300	69	411
6ww	MATPOWER	6	3	11
9	MATPOWER	9	3	9
14	MATPOWER	14	5	20
30	MATPOWER	30	6	41
39	MATPOWER	39	10	46
57	MATPOWER	57	7	80
89pegase	MATPOWER	89	12	210
118	MATPOWER	118	54	186
300	MATPOWER	300	69	411
1354pegase	MATPOWER	1354	260	1991
3_lmbd	PG-LIB	3	3	3
3_lmbd_api	PG-LIB	3	3	3
14_ieee	PG-LIB	14	5	20
14_ieee_api	PG-LIB	14	5	20
14_ieee_sad	PG-LIB	14	5	20
30_fsr	PG-LIB	30	6	41
30_fsr_api	PG-LIB	30	6	41
30_ieee	PG-LIB	30	6	41
30_ieee_api	PG-LIB	30	6	41
39_epri	PG-LIB	39	10	46
39_epri_api	PG-LIB	39	10	46
57_ieee	PG-LIB	57	7	80
57_ieee_api	PG-LIB	57	7	80
89_pegase	PG-LIB	89	12	210
118_ieee	PG-LIB	118	54	186
118_ieee_api	PG-LIB	118	54	186
118_ieee_sad	PG-LIB	118	54	186
162_ieee_dtc	PG-LIB	162	12	284
162_ieee_dtc_api	PG-LIB	162	12	284
162_ieee_dtc_sad	PG-LIB	162	12	284
300_ieee	PG-LIB	300	69	411
300_ieee_api	PG-LIB	300	69	411

TABLE 5.1 – Caractéristiques des 42 instances étudiées. n est le nombre de nœuds du réseau électrique. n_p est le nombre de nœuds producteurs.

On remarque que pour des instances ayant le même de nombre de nœuds électriques, on retrouve souvent le même nombre de producteurs et le même nombre d'arêtes. Dans ce cas, c'est parce que toutes ces instances ne sont que des variantes d'une instance originelle qui est souvent celle de la bibliothèque MATPOWER. Les différences entre ces instances se situent alors au niveau des valeurs prises par les différentes données : admitance, bornes de production, tension maximale aux nœuds ou puissance demandée.

5.2 Évaluation de RC-OPF sur des instances $(OPF - L)$

5.2.1 Comparaison numérique de RC-OPF sans contractions de bornes avec le logiciel BARON

Dans cette section, on présente des résultats obtenus en résolvant les 20 instances $(OPF - L)$ provenant des bibliothèques MATPOWER et Bukhsh. Nous ne présentons pas ici les résultats obtenus sur les instances de la bibliothèque PG-LIB. Néanmoins, les résultats obtenus par notre méthode RC-OPF avec contractions de bornes sur les instances $(OPF - L)$ issues de PG-LIB sont présentés en section 5.2.4. Pour chaque instance, on lance notre algorithme ainsi que le logiciel BARON 17.3.1 [40], avec les options par défaut. Pour notre algorithme, la relaxation du rang $(SDP - L)$ est résolue avec le logiciel SEDUMI. Les relaxations quadratiques convexes $(\overline{OPF - L_{S^*} - MCC})$ résolues en chaque nœud de l'arbre de *branch-and-bound* le sont avec le logiciel Xpress [48]. L'heuristique de recherche de solutions réalisables est lancée tous les trois nœuds de notre arbre. Elle est résolue avec le logiciel KNITRO [20, 49]. Nous rappelons qu'aucune technique de contraction de bornes n'est utilisée dans les expérimentations dont les résultats sont présentées en Table **??**.

Pour effectuer ces expérimentations, nous utilisons les paramètres suivants :

Tolérance de faisabilité absolue pour l'heuristique	10^{-5}
Gap d'optimalité final ϵ	10^{-3}
Paramètre de détermination de la valeur de coupe δ pour le branchement	0.5

Avec ces paramètres, on considère qu'un gap d'optimalité nul correspond à un gap d'optimalité inférieur à ϵ. Le choix de $\delta = 0.5$ est fait après des tests numériques que nous avons effectués ainsi qu'à partir des considérations de [1].

La Table **??** présente les résultats numériques. Chaque ligne correspond à une instance $(OPF - L)$.

	RC-OPF		BARON	
Instances	*Gap à la racine*	*Gap-Temps*	*Gap à la racine*	*Gap-Temps*
WB2	2.2%	1s	2.2%	1s
LMBM3	0%	1s	0%	1s
WB3	0%	1s	0%	1s
WB5	16.7%	23s	16.8%	1.92s
6ww	0%	1s	0.2%	1s
9	0%	1s	0%	1.34s
9mod	0.1%	(0.1%)	12.4%	5.83s
14	0%	1s	100%	(21.1%)
22loop	0%	1s	31.6%	(31.6%)
30	0%	1s	100%	(100%)
39	0%	2s	100%	(100%)
39mod1	0.1%	(0.1%)	100%	(100%)
39mod2	0.1%	(0.1%)	100%	(100%)
57	0%	1s	100%	(100%)
89pegase	0%	2s	72%	(72%)
118	0%	3s	100%	(100%)
118mod	0%	7s	100%	(100%)
300	0%	10s	100%	(100%)
300mod	0%	12s	100%	(100%)
1354pegase	0%	204s	69%	(69%)

TABLE 5.2 – Comparaison entre RC-OPF sans contractions de bornes et BARON. Le gap d'optimalité relatif ϵ vaut 10^{-3}. Le temps limite est de 5 minutes.

La première colonne regroupe les noms des instances. Le numéro dans le nom d'une instance correspond au nombre de nœuds électriques dans le réseau étudié, le nombre de variables réelles de ($OPF - L$) est donc égal au double de ce nombre. La double colonne « RC-OPF » correspond aux résultats numériques obtenus avec notre algorithme RC-OPF décrit en Section 2.4. La double colonne « BARON » correspond aux résultats numériques obtenus avec le logiciel BARON. Pour chaque méthode, la colonne « Gap à la racine » donne le gap relatif entre la borne inférieure obtenue au nœud racine de l'arbre et la valeur de la meilleure solution réalisable connue. Si on note UB la valeur de la meilleure solution réalisable trouvée et LB une borne inférieure, le gap relatif entre ces deux quantités vaut alors : $\frac{UB-LB}{UB}$. La colonne « Gap-Temps » renvoie le temps de résolution si l'optimalité a été atteinte en moins de cinq minutes de calcul. Si ce n'est pas le cas, cette colonne renvoie le gap relatif entre la borne inférieure finale et la valeur de la meilleure solution réalisable trouvée.

Pour les instances de moins de 10 nœuds électriques, le gap d'optimalité est réduit en deça de ϵ par les deux méthodes, sauf pour l'instance *9mod* où notre algorithme échoue à passer en dessous de 10^{-3}. On observe que le gap à la racine pour notre algorithme est systématiquement très faible. Cela provient de la qualité de la relaxation ($SDP - L$) comme cela a déjà été observé par d'autres auteurs [22, 9]. Au contraire, le gap à la racine obtenu avec BARON est souvent très important pour les instances de plus de 10 nœuds électriques. De plus ce gap n'est pas réduit au cours des 5 minutes de calculs.

On notera aussi qu'une solution réalisable est obtenu pour chaque instance par notre algorithme, et qu'elle peut être sensiblement améliorée au cours du parcours de l'arbre de *branch-and-bound*. Par exemple pour l'instance « 300 », la première solution réalisable obtenue vaut 475783, tandis que la meilleure obtenue vaut 475462.2.

5.2.2 Impact des contractions de bornes

Dans cette section on discute l'impact de l'ajout de techniques de contractions de bornes sur les domaines des variables au nœud racine de notre arbre.

La Table **??** présente les résultats obtenus sur deux instances ($OPF - L$), l'une de la bibliothèque de Bukhsh (« WB5 »), l'autre de la bibliothèque MATPOWER (« 57 »). Pour chaque instance, on a calculé la réduction relative moyenne des domaines des variables v obtenue avec différentes techniques

Instances	*FB+RC*	*QP*	*QP+FB+RC*	*QP2*	*QP2+FB+RC*
WB5	0.0%	66.8%	66.9%	81.7%	81.7%
57	0.4%	7.3%	7.3%	93.2%	93.2%

TABLE 5.3 – Contractions des domaines des variables pour deux instances ($OPF - L$)

de contractions de bornes. La colonne *FB+RC* correspond à l'utilisation de l'algorithme FBBT-OPF décrit dans la Section 3.2 ainsi que de la partie RCBT de l'algorithme OBBT-OPF décrit dans la Section 3.1. La colonne *QP* correspond à l'utilisation de l'algorithme OBBT-OPF, mais sans effectuer les opérations RCBT. Cela consiste donc à résoudre une succession de problèmes quadratiques convexes. Ici on considère que l'on n'ajoute pas la contrainte quadratique convexe utilisant S^* (3.1) aux problèmes (MIN_k) et (MAX_k). Cette contrainte sert à indiquer que la fonction objectif f_{S^*} doit rester en dessous de la meilleure borne supérieure connue à ce stade du parcours de RC-OPF. La colonne *QP+FB+RC* correspond à l'utilisation des deux algorithmes OBBT-OPF et FBBT-OPF toujours sans inclure la contrainte (3.1). La colonne *QP2* correspond aux limitations de la colonne *QP* où l'on ajoute la contrainte (3.1) aux problèmes quadratiques convexes (MIN_k) et (MAX_k). Enfin la dernière colonne *QP2+FB+RC* correspond à l'utilisation conjointe des deux algorithmes OBBT-OPF et FBBT-OPF avec l'ajout de la contrainte (3.1).

On observe que les techniques FBBT et RCBT n'ont qu'un faible impact sur la réduction de la taille des domaines, au contraire de la résolution de problèmes quadratiques convexes. Dans le cas de l'instance 57, on observe le rôle significatif de la contrainte (3.1). Les importantes réductions moyennes obtenues prouvent que lors de la résolution à l'optimum global de ($OPF - L$) avec un algorithme de *branch-and-bound*, l'utilisation de ces techniques pourra potentiellement grandement réduire le nombre de nœuds à parcourir dans l'arbre.

5.2.3 Impact du choix du paramètre δ

Dans cette section nous étudions l'impact du paramètre de branchement δ sur RC-OPF. Nous rappelons que le rôle du paramètre δ est de déterminer en quelle valeur couper l'intervalle de la variable sélectionnée par la méthode de branchement (voir Section 2.4.3). Pour mener à bien cette étude, nous nous proposons de comparer le nombre de nœuds dans l'algorithme RC-OPF nécessaires pour résoudre à l'exacte une instance d'OPF en fonction de la valeur choisie pour δ. L'instance *WB5* est adaptée à cette

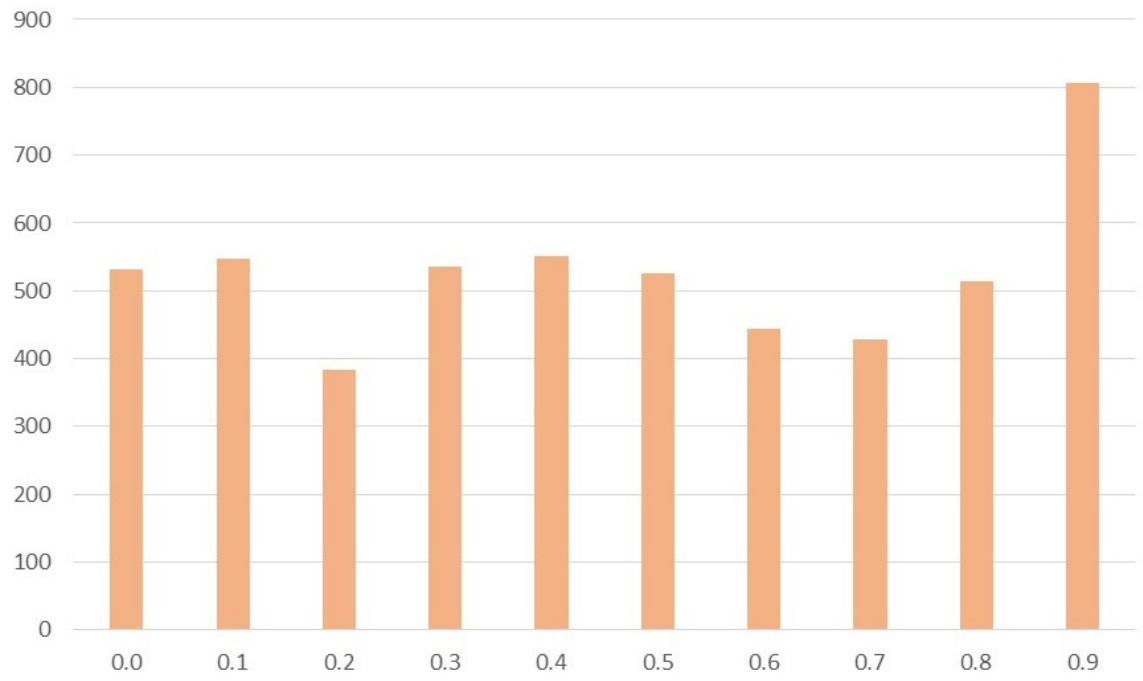

FIGURE 5.1 – Nombre de nœuds nécessaires pour atteindre l'optimalité dans le cadre de la résolution de WB5 avec RC-OPF, en fonction de δ. On utilise FBBT-OPF comme contractions.

étude, car d'une part nous savons la résoudre de manière exacte, et d'autre part nous avons besoin de plusieurs centaines de nœuds pour la résoudre de manière exacte en utilisant RC-OPF uniquement avec les contractions FBBT-OPF.

Dans la Figure 5.1, on voit le nombre nœuds nécessaires pour atteindre l'optimalité dans le cadre de la résolution de *WB5* avec RC-OPF. Les valeurs prises par le paramètre δ sont indiquées en abscisse.

Nous remarquons sur ce graphique que le nombre de nœuds ne varie pas très fortement avec la valeur de δ. Nous remarquons également deux minima locaux pour $\delta = 0.2$ et $\delta = 0.7$. Il n'y a que la valeur de 0.9 qui semble se détacher. Nous concluons ici qu'il vaut mieux éviter les valeurs de δ proches de 1, c'est-à-dire qu'il vaut mieux plutôt privilégier le branchement proche de la valeur optimale prise par la variable dans la résolution ($\overline{OPF_{S^*} - MCC}$).

5.2.4 Résultats de RC-OPF avec contractions

Dans cette section, on décrit les résultats obtenus en résolvant des instances $(OPF - L)$ provenant des bibliothèques de MATPOWER, de Bukhsh et PG-LIB à l'aide de notre méthode RC-OPF et en appliquant les techniques de contractions de bornes décrites au chapitre 3, c'est-à-dire les algorithmes OBBT-OPF et FBBT-OPF. Pour notre algorithme, la relaxation du rang $(SDP - L)$ est résolue avec le logiciel SEDUMI. En utilisant les variables duales obtenues à l'issue de cette résolution, on construit la matrice de reformulation optimale S^*. Les relaxations quadratiques convexes $(\overline{OPF - L_{S^*} - MCC})$ résolues en chaque nœud de l'arbre de *branch-and-bound* le sont avec le logiciel Xpress [48]. Il en est de même pour les relaxations (MIN_k) et (MAX_k) résolues pour toute variable k en chaque nœud de l'arbre de *branch-and-bound*. L'heuristique de recherche de solutions réalisables est lancée tous les trois nœuds de notre arbre. Elle est résolue avec le logiciel KNITRO [20, 49].

Pour effectuer ces expérimentations, nous utilisons les paramètres suivants :

Tolérance de faisabilité absolue pour l'heuristique	10^{-5}
Gap d'optimalité final ϵ	10^{-3}
Paramètre de détermination de la valeur de coupe δ pour le branchement	0.5

La Table 5.4 présente les résultats obtenus pour nos 42 instances $(OPF - L)$. Pour les instances relatives à des réseaux de plus de 60 nœuds, nous n'utilisons pas les techniques de contractions de bornes s'appuyant sur la résolution de programmes quadratiques convexes, c'est-à-dire l'algorithme OBBT-OPF. En effet ces techniques sont trop longues à mettre en œuvre pour ces instances.

Dans ce tableau, chaque ligne correspond à une instance. La première colonne regroupe les noms des instances. La seconde colonne donne le gap d'optimalité pour $(SDP - L)$, qui vaut : $\frac{\mathbf{UB}-\mathbf{LB}_{SDP}}{\mathbf{UB}}$, où **UB** est la meilleure solution réalisable trouvée pour l'instance et $\mathbf{LB}_{SDP}$ est la borne inférieure fournie par $(SDP - L)$. Cela correspond au gap initial de RC-OPF. La troisième colonne est le gap d'optimalité à la fin de l'algorithme de *branch-and-bound* RC-OPF. Posons $\mathbf{LB}_{B\&B}$ la borne inférieure obtenue à la fin de l'algorithme. Le gap d'optimalité vaut : $\frac{\mathbf{UB}-\mathbf{LB}_{B\&B}}{\mathbf{UB}}$. La quatrième colonne est le nombre de nœuds traités au cours du parcours de l'arbre dans l'algorithme. La dernière colonne donne le temps de calcul en secondes.

Pour beaucoup d'instances, la valeur optimale de la relaxation du rang $(SDP - L)$ est égale à la

Instances	*gap SDP*	*gap B&B*	*Nœuds*	*Temps*
WB2	2.2%	0%	3	1s
WB3	0%	0%	1	1s
LMBM3	0%	0%	1	1s
3_lmbd	0%	0%	1	1s
3_lmbd_api	0%	0%	1	3s
WB5	17%	0%	27	9s
6ww	0%	0%	1	1s
9	0%	0%	1	3s
9mod	0.01%	0%	55	48s
14	0%	0%	1	8s
14_ieee	0%	0%	1	24s
14_ieee_api	0%	0%	1	27s
14_ieee_sad	0%	0%	1	23s
22loop	0%	0%	1	9s
30	0%	0%	1	31s
30_fsr	0%	0%	1	1s
30_fsr_api	0%	0%	1	1s
30_ieee	0%	0%	1	2s
30_ieee_api	0%	0%	1	7s
39	0%	0%	1	2s
39mod1	0.15%	0.15%	203	3600s
39mod2	0.15%	0.15%	203	3600s
39_epri	0%	0%	1	13s
39_epri_api	0%	0%	1	2s
57	0%	0%	1	79s
57_ieee	0%	0%	1	164s
57_ieee_api	0%	0%	1	342s
89pegase	0%	0%	3	2s
89_pegase	0%	0%	52	19s
118	0%	0%	1	3s
118mod	0%	0%	1940	3600s
118_ieee	0%	0%	1	3s
118_ieee_api	0%	0%	103	52s
118_ieee_sad	0%	0%	1	3s
162_ieee_dtc	2.7%	2.7%	1587	3600s
162_ieee_dtc_api	0%	0%	1419	3600s
162_ieee_dtc_sad	2.7%	2.7%	1699	3600s
300	0%	0%	1391	3600s
300mod	0%	0%	1	12s
300_ieee	0.33%	0.33%	1122	3600s
300_ieee_api	0%	0%	1241	3600s
1354pegase	0%	0%	1	204s

TABLE 5.4 – Résultats pour 42 instances ($OPF - L$) de RC-OPF avec OBBT-OPF et FBBT-OPF, 1h de temps de calcul et un gap d'optimalité relative valant 10^{-3}.

valeur du problème $(OPF - L)$. Dans ce cas, il n'y a pas besoin d'effectuer le parcours de l'arbre de *branch-and-bound* pour faire remonter la borne inférieure. Cependant, il faut lancer l'heuristique afin d'obtenir une solution réalisable optimale. Grâce à l'efficacité de notre heuristique [22], la solution optimale est souvent trouvée dès la racine de l'arbre.

Pour les instances relatives à des réseaux de quelques nœuds, le gap d'optimalité est annulé au cours du parcours de l'arbre dans l'algorithme de *branch-and-bound*. C'est le cas pour les instances *WB2* et *WB5* où il y a le plus d'écart entre les bornes inférieure et supérieure au nœud racine. On remarque que la borne inférieure est réhaussée jusqu'à la valeur optimale du problème au cours des itérations de notre algorithme. Pour les instances relatives aux réseaux de plus grande taille, le gap entre la borne inférieure donnée par $(SDP - L)$ et la valeur de la meilleure solution connue, quand il existe, est plutôt faible. Cependant nous ne parvenons pas à le réduire en utilisant notre algorithme. En 1 heure de calculs, une solution réalisable est toujours obtenue, et peut etre sensiblement améliorée au cours du parcours de l'arbre. On remarque aussi, en comparant avec les résultats de la Table **??**, que l'instance *9mod* est maintenant résolue à l'optimum global par notre algorithme, ceci grâce à l'utilisation des contractions de bornes.

5.3 Résultats obtenus avec RC-OPF-Q sur $(OPF - Q)$

Dans cette section, on décrit les résultats obtenus en résolvant des instances $(OPF - Q)$ provenant des bibliothèques MATPOWER, de Bukhsh et PG-LIB à l'aide de notre algorithme adapté RC-OPF-Q et en appliquant les techniques de contractions de bornes des algorithmes OBBT-OPF et FBBT-OPF. Nous ne comparons pas nos résultats à ceux obtenus avec BARON, mais dans l'étude [12], il est possible de constater que ce logiciel ne parvient pass à résoudre des instances $(OPF - Q)$ dès que le réseau électrique contient une dizaine de nœuds électriques. Pour notre algorithme, la relaxation du rang $(SDP - Q)$ est résolue avec le logiciel SEDUMI. En utilisant les variables duales obtenues à l'issue de cette résolution, on construit la matrice de reformulation optimale S^*. Les relaxations quadratiques convexes $(\overline{OPF - Q_{S^*} - MCC})$ résolues en chaque nœud de l'arbre de *branch-and-bound* le sont avec le logiciel Xpress [48]. Il en est de même pour les relaxations $(MIN_k - Q)$ et $(MAX_k - Q)$ résolues pour toute variable k en chaque nœud de l'arbre de *branch-and-bound*. L'heuristique de recherche de solutions réalisables est lancée tous les trois nœuds de notre arbre. Elle est résolue avec le logiciel

KNITRO [20, 49].

Pour effectuer ces expérimentations, nous utilisons les paramètres suivants :

Tolérance de faisabilité absolue pour l'heuristique	10^{-5}
Gap d'optimalité final ϵ	10^{-3}
Paramètre de détermination de la valeur de coupe δ	0.5

La Table 5.5 présente les résultats obtenus pour 42 instances $(OPF - Q)$. Ici encore, pour les instances relatives à des réseaux de plus de 60 nœuds, nous n'utilisons pas les techniques de contractions de bornes s'appuyant sur la résolution de programmes quadratiques convexes, c'est-à-dire l'algorithme OBBT-OPF, puisque ces techniques sont trop longues à mettre en œuvre pour ces instances.

Dans ce tableau, chaque ligne correspond à une instance. La première colonne regroupe les noms des instances. Le numéro dans le nom d'une instance correspond au nombre de nœuds électriques dans le réseau étudié. La seconde colonne donne le gap d'optimalité pour $(SDP - Q)$, qui vaut : $\frac{\mathbf{UB}-\mathbf{LB}_{SDP}}{\mathbf{UB}}$, où **UB** est la meilleure solution réalisable trouvée pour l'instance et $\mathbf{LB}_{SDP}$ est la borne inférieure fournit par $(SDP - Q)$. Cela correspond au gap initial de RC-OPF-Q. La troisième colonne est le gap d'optimalité à la fin de l'algorithme de *branch-and-bound* RC-OPF-Q. Posons $\mathbf{LB}_{B\&B}$ la borne inférieure obtenue à la fin de l'algorithme. Le gap d'optimalité vaut : $\frac{\mathbf{UB}-\mathbf{LB}_{B\&B}}{\mathbf{UB}}$. La quatrième colonne est le nombre de nœuds traités au cours du parcours de l'arbre dans l'algorithme. La dernière colonne donne le temps de calcul en secondes.

On remarque des résultats similaires à ceux observés pour la résolution de $(OPF - L)$. En effet, les gaps initiaux dus à la relaxation SDP $(SDP - Q)$ sont le plus souvent nuls (c'est-à-dire inférieurs à 0.1%). Quand il y a un gap initial, on parvient à le fermer grâce à notre algorithme de *branch-and-bound* pour les plus petites instances. C'est notamment vrai pour les instances *WB2*, *WB5* et *9mod*. Pour les autres instances on ne parvient pas à réduire ce gap en 1h de calcul.

Comparaison avec l'état de l'art

Ici nous mettons en perspective nos résultats obtenus pour $(OPF - Q)$ avec ceux de l'état de l'art. Il est difficile de se comparer numériquement à l'état de l'art, car même si la communauté OPF

Instances	*gap SDP*	*gap B&B*	*Nœuds*	*Temps*
WB2	2.2%	0%	3	2s
WB3	0%	0%	1	1s
LMBM3	0%	0%	1	1s
3_lmbd	0%	0%	1	1s
3_lmbd_api	0%	0%	1	1s
WB5	17%	0%	69	15s
6ww	0%	0%	1	2s
9	0%	0%	1	1s
9mod	11%	0%	569	402s
14	0%	0%	1	14s
14_ieee	0%	0%	1	23s
14_ieee_api	0%	0%	1	37s
14_ieee_sad	0%	0%	1	24s
22loop	0%	0%	1	< 1s
30	0%	0%	1	69s
30_fsr	0%	0%	1	73s
30_fsr_api	0%	0%	1	1s
30_ieee	0%	0%	1	7s
30_ieee_api	0%	0%	1	24s
39	0%	0%	16	259s
39mod1	3.7%	3.7%	210	3600s
39mod2	0.15%	0.14%	438	3600s
39_epri	0%	0%	1	13s
39_epri_api	0%	0%	1	7s
57	0%	0%	1	2s
57_ieee	0%	0%	1	299s
57_ieee_api	0%	0%	1	470s
89pegase	0%	0%	1	2s
89_pegase	0%	0%	1	3s
118	0%	0%	1	4s
118mod	0%	0%	1	7s
118_ieee	0%	0%	1	5s
118_ieee_api	0%	0%	4184	3600s
118_ieee_sad	0%	0%	1	4s
162_ieee_dtc	2.7%	2.7%	2183	3600s
162_ieee_dtc_api	0%	0%	1	7s
162_ieee_dtc_sad	2.7%	2.7%	1318	3600s
300	0%	0%	671	3600s
300mod	0.14%	0.14%	370	3600s
300_ieee	0.33%	0.33%	1079	3600s
300_ieee_api	0%	0%	1332	3600s
1354pegase	0%	0%	1	253s

TABLE 5.5 – Résultats pour 42 instances $(OPF - Q)$ de RC-OPF-Q avec OBBT-OPF et FBBT-OPF et 1h de temps de calcul

travaille en général sur les mêmes instances, les contraintes effectivement présentes dans la modélisation mathématique peuvent être différentes. Il en résulte que les problèmes étudiés sont en fait différents, même s'ils sont désignés sous le même nom. Ces différences se situent au niveau des contraintes de limite technique. Ce ne sont pas forcément les mêmes qui sont ajoutées au problème.

Dans la suite on met en perspective nos résultats avec ceux de l'étude [50] portant sur la résolution exacte de l'OPF. On retrouve dans cette étude les instances de la bibliothèque PG-LIB étudiées dans ce manuscrit. Les modélisations faites de l'OPF dans cette étude correspondent à $(OPF - Q)$ auquel il faut ajouter certaines contraintes de limite technique. Ce sont les contraintes de puissance maximale sur les lignes du réseau :

$$|S_{i,j}| \leq \mathbf{S}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E} \qquad (12).$$

Ainsi que les contraintes limitant les différences angulaires des tensions de nœuds adjacents dans le graphe du réseau :

$$|ang(V_i) - ang(V_j)| \leq \boldsymbol{\theta}_{i,j}^{max}, \quad \forall (i,j) \in \mathcal{E} \qquad (15).$$

Nous rappelons que c'est à dessein que nous n'avons pas ajouté les limites sur les puissances traversant les lignes. En effet, dans les outils de conduite et d'étude du réseau utilisés par RTE, ces contraintes n'apparaissent pas RTE préfère à la place inclure les contraintes limitant le courant sur les lignes (contraintes(11), voir Section 0.2.4). Dans la section 2.2.4 nous montrons que ces contraintes peuvent être ajoutées à $(OPF - L)$ sans dénaturer la méthode RC-OPF. Elles peuvent s'ajouter de la même manière à $(OPF - Q)$. Sur le réseau de transport français se trouve des automates de protection qui déconnectent des équipements si un courant maximal est dépassé. C'est donc bien cette grandeur qu'il faut inclure dans le modèle $(OPF - Q)$ pour les études menées par RTE. De plus, les différences angulaires entre les tensions de nœuds voisins du réseau ne sont pas prises en compte par RTE.

Dans [50], les auteurs proposent un algorithme pour refermer le gap d'optimalité de l'OPF en utilisant une relaxation quadratique convexe couplée à un algorithme de réduction de bornes utilisant les techniques OBBT. Si l'on se compare avec cette étude, on remarque qu'elle ne prend en compte que des instances représentant des réseaux de moins de 300 nœuds, ce qui est également notre cas si l'on fait exception de l'instance *1354pegase*. Dans cette étude, le gap d'optimalité limite est de 1% contrairement à la notre où il vaut 0.1%. Là aussi nous avons choisi de nous concentrer sur une précision inférieure

Nom	Gap
9	$< 1 \times 10^{-7}$
9mod	7×10^{-4}
14	$< 1 \times 10^{-7}$
22loop	$< 1 \times 10^{-7}$
30	2×10^{-6}
39	3×10^{-5}
39mod1	1×10^{-3}
39mod2	1×10^{-3}
57	$< 1 \times 10^{-7}$
89pegase	7×10^{-7}
118	2×10^{-5}
118mod	2×10^{-6}

TABLE 5.6 – Gaps de dualité fournis par la relaxation SDP pour 12 instances d'OPF

à 1%, car du point de vue de RTE, une prècision de 1% sur l'optimalité n'est pas suffisante. Si l'on regarde notre Table 5.5, nous n'avons que deux instances de la bibliothèque PG-LIB avec un gap final en dessous de 1%. Il s'agit de *162_ieee_dtc* et de *162_ieee_dtc_api_sad*. La première d'entre-elles est aussi l'instance la moins refermée dans [50] avec un gap final de 1.53%. La seconde passe en dessous de 1% dans cette étude. Néanmoins, cela prouve que nous obtenons des résultats comparables, voire meilleurs que ceux d'une étude très récente (octobre 2019) de l'état de l'art.

5.4 Génération d'instances difficiles

On observe au cours de nos expérimentations que la recherche de solutions admissibles en utilisant un solveur par points intérieurs est en général aisée, et fournit une solution souvent optimale, ce dès l'appel au nœud racine. De plus, la borne inférieure fournie par la résolution de $(SDP - L)$ est souvent exacte ou très proche de la valeur de la meilleure solution connue. C'est notamment le cas pour les instances des bibliothèques MATPOWER et Bukhsh comportant de 9 à 118 nœuds, voir la Table 5.4.

La Table 5.6 montre les gaps ($\frac{ub-lb}{ub}$) entre la valeur de la meilleure solution trouvée (ub) et celle de la borne inférieure (lb) fournie par la résolution de $(SDP - L)$ pour 12 instances $(OPF - L)$. Le nombre dans le nom de l'instance renseigne le nombre de nœuds dans le réseau électrique correspondant. Il n'y que pour des instances à très peu de nœuds dans le réseau électrique où l'on observe un gap important entre valeur optimale et la valeur de la relaxation $(SDP - L)$ (par exemple 16% pour WB5,

instance à 5 nœuds, voir la Table 5.4). De plus, quand une instance dépasse la dizaine de nœuds dans le réseau électrique, la borne inférieure n'évolue guère au cours de l'arbre de *branch-and-bound*, voir le cas de l'instance *39mod1* dans la Table 5.4.

Les constats ci-dessus nous poussent à penser que, dans une optique de valorisations de différentes techniques de branchement au cours de l'arbre de *branch-and-bound*, il faut disposer de plus d'instances présentant un grand gap à la racine, c'est-à-dire d'une forte inexactitude de la borne obtenue par relaxation ($SDP - L$). En d'autres termes, il faut générer une banque d'instances ($OPF - L$) difficiles à résoudre, même à faible précision, pour notre méthode. Pour ce faire, l'idée générale est de modifier des instances de manière déterministe ou aléatoire en espérant obtenir de plus grands gaps de dualité, en pratique de quelques pourcents (donc supérieur à 10^{-2}).

Avant d'aller plus loin il est nécessaire de définir précisément les notions de gap qui seront présentes par la suite. Comme nous recherchons des instances où la relaxation ($SDP - L$) est non exacte, nous nous intéressons au gap à la racine. Le gap à la racine est l'écart relatif entre la valeur optimale, vp, du problème et la borne fournit par la relaxation $SDP - L$, lb. Il se calcule comme ceci : $\frac{vp-lb}{vp}$. Cependant, il n'est pas possible de connaître la valeur optimale vp du problème sans l'avoir résolu, c'est pourquoi on va introduire la notion de gap potentiel. Une solution admissible fournit une borne supérieure ub du problème. Cette solution conduit à un gap potentiel : $\frac{ub-lb}{ub}$. Le gap potentiel est donc toujours supérieur au gap à la racine.

On utilise ces gaps potentiels pour sélectionner *a priori* les instances pour lesquelles on cherchera à calculer l'optimum global, et donc pour lesquelles on disposera *a posteriori* d'une évaluation plus rigoureuse du gap à la racine.

Maintenant nous présentons une manière de générer de nouvelles instances à partir d'instances déjà à disposition. Cette approche est aussi utilisée dans [12] pour générer d'autres instances à partir d'instances plus classiques afin d'étudier le comportement d'un algorithme de *branch-and-bound* pour résoudre l'OPF. Notre méthode consiste à venir modifier les coûts de production linéaires. Chaque unité de production va se voir attribuer un nouveau coût de production choisi aléatoirement (pour chaque unité) entre 1 et 10. Les 10 valeurs possibles sont équiprobables. Nous avons choisi de reproduire

Nom	Gap
9	$[1; 5]$
9mod	$[1; 5]$
14	$[20; 40]$
22loop	$\{2\}$
30	$[1; 3.25]$
39	$\{0.3\}$
39mod1	$\{0.3\}$
39mod2	$\{0.3\}$
57	$[20; 40]$
89pegase	$\{1\}$
118	$[20; 40]$
118mod	$[20; 40]$

TABLE 5.7 – Intervalle des coefficients linéaires de production active

de nouvelles instances de cette manière plutôt qu'en modifiant les données de consommation et de production comme dans [12] car cela nous semble plus à même de générer de nouvelles situations de réseau réalistes. La Table 5.7 présente les intervalles dans lesquels se situent les coefficients de coût de production initiaux pour les 12 instances précédentes.

Avec cette méthode on peut introduire un facteur 10 entre le prix le plus élevé et le prix le plus faible. Cela représente un écart un peu plus élevé que ce que l'on constate dans nos instances de départ, mais cela ne nous semble pas irréaliste pour autant. L'enjeu est maintenant de repérer les instances générées aléatoirement disposant d'un gap potentiel suffisant.

Cette méthode a été utilisée sur chacune des 20 instances d'OPF suivantes :

— 6ww, 9, 14, 30, 39, 57, 89pegase, 118, 300 et 1354pegase que l'on trouve dans la bibliothèque MATPOWER [3].

— WB2, LMBM3, WB3, WB5, 9mod, 22loop, 39mod1, 39mod2, 118mod et 300mod qui sont décrites dans [2].

Le détail de l'application et des résultats obtenus danss la recherche d'instances difficiles avec notre méthode se trouvent en annexe. Nous donnons ci dessous une synthèse des résultats. Pour la plupart des instances, les gaps à la racine sont inférieurs à 1%, qui est notre valeur cible. Il n'y a que sur les instances 57 et 300mod que nous avons obtenu des résultats probants. On notera néanmoins que

Instances	*LB SDP*	*LB B&B*	*UB*	*Gap SDP*	*Gap B&B*	*#nœuds*
84	2359.582	2433.053	2433.889	3.05%	0.03%	5608
260	3483.577	3542.080	3542.642	1.67%	0.02%	5546
267	5268.374	5315.100	5333.004	1.21%	0.34%	4522
299	6147.227	6241.130	6267.343	1.92%	0.42%	5061
391	1893.940	1917.854	1917.963	1.25%	0.01%	5794
628	2318.442	2478.559	2483.231	6.64%	0.19%	5603
683	3131.342	3204.889	3205.613	2.23%	0.02%	5717
829	3525.509	3596.291	3597.400	2.00%	0.03%	5440
868	3979.457	4062.699	4067.721	2.17%	0.12%	5146
974	4448.716	4518.970	4535.649	1.92%	0.37%	4781
Moyenne	-	-	-	2.41%	0.16%	5322

TABLE 5.8 – Résultats de `RC-OPF` avec les contractions de bornes sur 10 instances difficiles et un temps limite de 120h.

sur des instances construites pour avoir plusieurs optimums locaux (voir [2]), comme 22loop, 39mod1 et 300mod, l'algorithme de *branch-and-bound* parvient à en trouver plusieurs et donc à améliorer la meilleure solution connue au cours de son exécution. Les instances obtenues à partir de l'instance 57 semblent être les meilleures candidates pour venir tester de futurs renforcements des techniques de branchements au cours de l'algorithme de *branch-and-bound*.

Résultats sur les instances difficiles de $(OPF - L)$

La Table 5.8 donne les résultats obtenus avec notre algorithme `RC-OPF` utilisé pour résoudre les 10 instances difficiles $(OPF - L)$ générées à partir de l'instance 57 de MATPOWER. Pour ces expérimentations, nous avons utilisé les deux algorithmes de contraction de bornes OBBT-OPF et FBBT-OPF à l'intérieur de `RC-OPF`. Notons que la contrainte (3.1) est bien ajoutée aux problèmes quadratiques convexes (MIN_k) et (MAX_k). Nous avons pour toutes les instances laissé notre algorithme calculer jusqu'à la fin du temps limite de 120h. Chaque ligne du tableau correspond à une instance. La colonne *LB SDP* donne la borne inférieure provenant de la relaxation $(SDP - L)$. La colonne *LB B&B* donne la borne inférieure à la fin de notre algorithme `RC-OPF`. La colonne *UB* donne la meilleure borne supérieure trouvée. La colonne *Gap SDP* correspond au gap entre la borne $(SDP - L)$ et la valeur de la meilleure solution connue. La colonne *Gap B&B* correspond au gap entre la borne inférieure obtenue à la fin de notre algorithme et la valeur de la meilleure solution connue. La colonne *#nœuds*

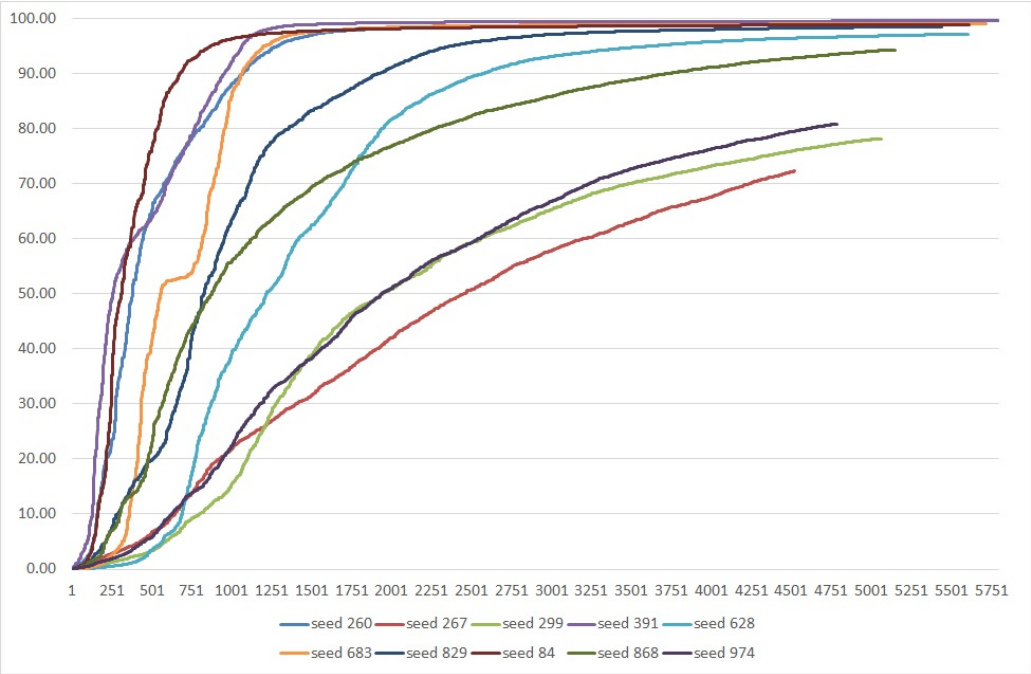

FIGURE 5.2 – Réduction du gap au cours des itérations pour les 10 instances difficiles créées à partir de l'instance 57

correspond au nombre de nœuds parcourrus dans l'arbre de notre algorithme `RC-OPF`.

On observe que le gap est bien réduit, et même réduit en dessous de 1% à l'aide de notre algorithme par rapport à l'utilisation de la relaxation $(SDP - L)$. Ces résultats nous permettent de montrer que notre algorithme peut réduire des gaps d'optimalité en faisant augmenter la meilleure borne inférieure connue sur un problème $(OPF - L)$. Ces résultats sont obtenus par l'utilisation conjointe de la reformulation optimale introduite dans la Section 2.2.3 et des contractions de bornes de la Section 3.1.1.

La Figure 5.2 montre l'évolution, pour les 10 instances difficiles générées aléatoirement de la réduction du gap entre la borne inférieure de provenant de $(SDP - L)$ et la meilleure valeur connue du problème. Cette évolution se fait au cours des itérations dans l'arbre de notre algorihtme de *branch-and-bound*. Cette réduction s'exprime en pourcent, c'est pourquoi toutes les courbes commencent à 0 et ne peuvent dépasser 100. Plus explicitement, soit UB la meilleure valeur trouvée d'un problème, soit LB_0 la borne inférieure connue au nœud racine de l'arbre et soit LB la borne inférieure connue

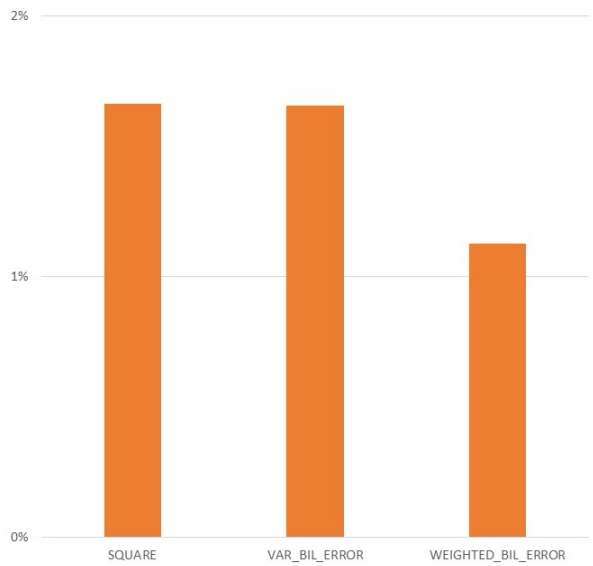

FIGURE 5.3 – Gap final moyen pour les 10 instances difficiles à 57 nœuds après 12h de calculs

sur ce problème à l'itération k. Le point de la courbe correspondant à ce problème et d'abscisse k aura pour ordonnée $100 \times \left(1 - \frac{UB-LB}{UB-LB_0}\right)$. On observe que pour 6 sur 10 de ces instances, le gap est rapidement réduit. On remarque aussi trois instances pour lesquelles le gap se réduit plus lentement et qui sont donc vues comme les plus difficiles générées.

Comparaisons des méthodes de branchement dans la méthode `RC-OPF`

Nous nous intéressons ici à comparer les différentes méthodes de branchement envisagées dans la Section 2.4.2 sur ces instances difficiles. Nous comparons trois méthodes de branchement :

— La méthode dite « erreur bilinéaire » que l'on note `VAR_BIL_ERROR`,

— La méthode similaire à la précédente mais donnant la priorité aux termes diagonaux, on la note `SQUARE`,

— La méthode dite d'« erreur bilinéaire pondérée ». Elle est notée `WEIGHTED_BIL_ERROR`.

Pour évaluer l'impact du choix d'une méthode de branchement, nous utilisons nos 10 instances difficiles construites à partir de l'instance *57*. La Figure 5.3 montre le gap final moyen obtenu par `RC-OPF` utilisé avec FBBT-OPF et OBBT-OPF sur les 10 instances difficiles à 57 nœuds.

Nous observons que les méthodes `SQUARE` et `VAR_BIL_ERROR` donnent des résultats similaires. Nous interprétons cela comme le fait que le fait de donner la priorité aux termes diagonaux ne change pas beaucoup la variable sélectionnée car il est possible que la variable sélectionnée par la méthode `VAR_BIL_ERROR` le soit du fait d'un terme diagonal. Nous remarquons néanmoins qur la méthode `WEIGHTED_BIL_ERROR` donne de bien meilleurs résultats, confirmant notre intuition initiale. C'est pourquoi c'est cette méthode qui est choisie par défaut dans `RC-OPF`.

Conclusion

Dans ce manuscrit, nous avons commencé par présenter les trois variantes de problèmes d'optimisation des flux de puissance (OPF) étudiées : $(OPF - L)$, $(OPF - Q)$ et $(OPF - UC)$. Nous avons ensuite rappelé l'état de l'art sur la résolution exacte de ces variantes, ainsi que sur la résolution exacte de problèmes quadratiques non-convexes. Nous avons notamment décrit la méthode MIQCR pour la résolution de problèmes quadratiques en variables mixtes. Dans le chapitre 2 nous avons d'abord traité le cas de la première variante, dite à coûts linéaires, $(OPF - L)$. Nous avons montré comment construire une reformulation quadratique optimale pour cette variante, phase 1 de notre méthode de résolution. Nous avons prouvé que cette construction était plus efficace que celle de la méthode MIQCR, tout en ayant la même qualité de borne. Nous avons proposé un algorithme de résolution exacte qui résout la reformulation optimale à l'aide d'un arbre de *branch-and-bound*. Ceci constitue la phase 2 de notre méthode de résolution, résumée dans notre algorithme `RC-OPF`. Nous avons terminé ce chapitre avec la généralisation de notre méthode au cas de la variante $(OPF - Q)$, et à la description de notre algorithme adapté `RC-OPF-Q`. Dans le chapitre 3, nous avons présenté les contractions sur les domaines des variables utilisées dans notre algorithme de résolution exacte. Nous avons notamment montré comment les contractions de type OBBT, pour *Optimality-Based Bound Tightening* sont plus efficaces lorsque l'on utilise la reformulation optimale de notre méthode `RC-OPF`. Nous avons également présenté de nouvelles contractions linéaires fondées sur la faisabilité. Toutes ces contractions peuvent être utilisées dans la phase 2 de `RC-OPF` ou de `RC-OPF-Q`. Dans le chapitre 4, nous avons constaté que notre méthode `RC-OPF` utilisait l'ordre 1 de la hiérarchie de Lasserre comme fondement pour sa reformulation optimale. Nous nous sommes demandé et avons répondu à la question : comment étendre naturellement cette approche aux ordres supérieures de la hiérarchie. Le chapitre 5, a donné des résultats numériques obtenus en appliquant notre méthode sur des instances de $(OPF - L)$ puis $(OPF - Q)$. Nous avons illustré l'impact décisif de nos techniques de contractions de bornes OBBT-OPF, ainsi que ceux d'autres paramètres de résolution. Nous avons mis en perspective nos résultats avec de récentes études issues de l'état de l'art. Nous avons aussi créé des instances dont la résolution exacte est « difficile » et sur lesquelles nous avons montré l'efficacité de notre algorithme de résolution, et affiné nos choix algorithmiques.

La Figure 5.4 illustre la synthèse des principales contribution de cette thèse, à savoir la conception et l'implémentation des deux méthodes de résolution `RC-OPF` et `RC-OPF-Q` respectivement dédiées à la

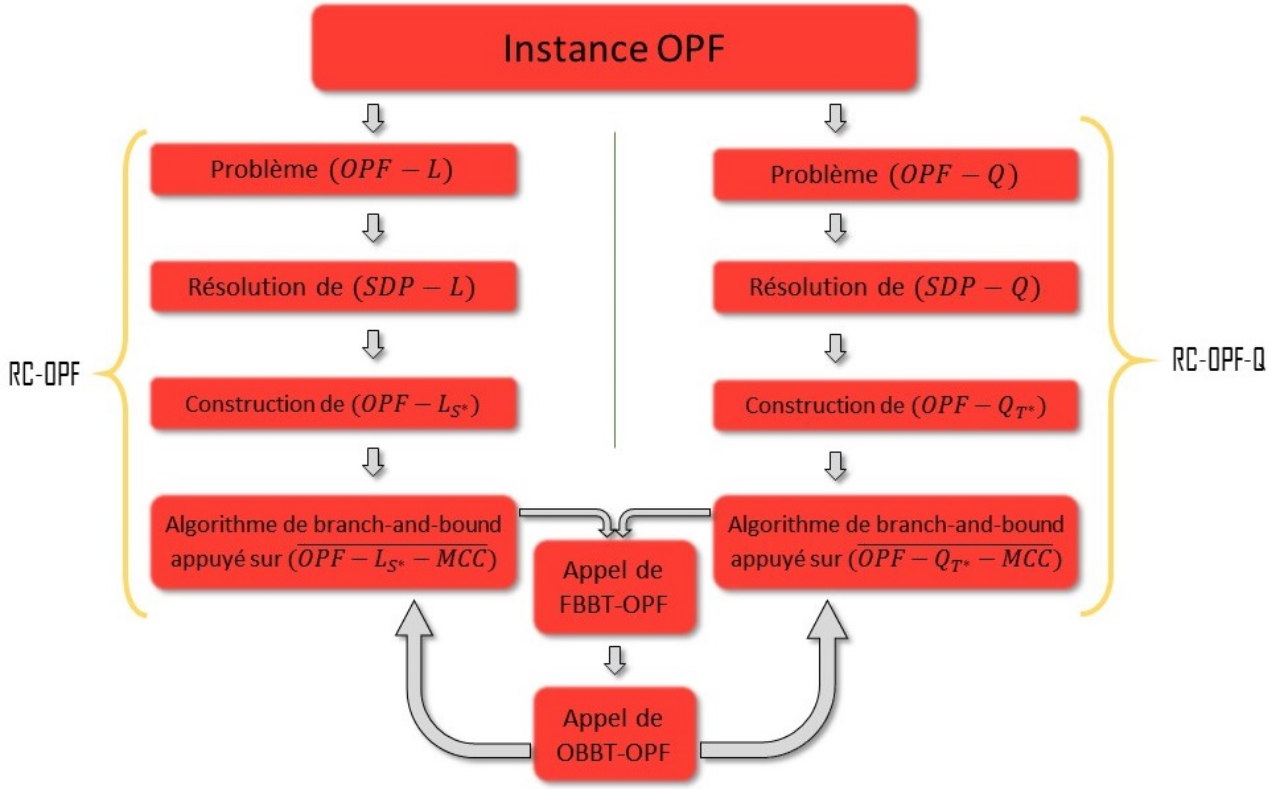

FIGURE 5.4 – Synthèse des contributions principales

résolution des problèmes $(OPF - L)$ et $(OPF - Q)$. La Table 5.9 dresse un panorama exhaustif des contributions liées à la résolution de $(OPF - L)$ et de $(OPF - Q)$ avec `RC-OPF` et `RC-OPF-Q`.

On peut ajouter comme contributions les deux éléments suivants :

— Généralisation de la méthode `RC-OPF` à des problèmes polynomiaux avec des ordres supérieurs de la hiérarchie,

— Génération de nouvelles instances $(OPF - L)$ pour lesquelles la relaxation du rang est inexact avec un gap d'optimalité supérieur à 1%.

La résolution de la variante $(OPF - UC)$ n'a pas été ici traitée. Cependant, on peut dessiner en perspective deux approches pour résoudre de manière optimale $(OPF - UC)$ en utilisant l'algorithme `RC-OPF-Q`. La première, et la plus simple de ces deux approches, consiste à construire la reformulation de la même manière que dans le cas de `RC-OPF-Q`. Ensuite, on résout cette reformulation dans un algorithme de *branch-and-bound* comme celui de `RC-OPF`, mais où l'on relaxe en chaque nœud le caractère binaire des variables b. Il suffit alors d'ajouter des branchements 0-1 sur ces variables pour concevoir un algorithme de résolution exacte convergeant. La seconde approche plus prospective consisterait en la généralisation de la technique de reformulation optimale au cas de la variante $(OPF - UC)$. Il faut noter que dans [51] les auteurs montrent que la variante $(OPF - UC)$ conserve le caractère creux que possède le graphe du réseau, comme c'est le cas pour les variantes $(OPF - L)$ et $(OPF - Q)$. Ceci laisse présager que la résolution des problèmes SDP ou quadrariques convexes nécessaires à la mise en œuvre de `RC-OPF` pour $(OPF - UC)$ restera accessible en terme de temps de calcul.

Un autre sujet de perspective serait l'amélioration de notre algorithme `RC-OPF` en faisant un appel non systématique aux techniques de contraction de bornes. Par exemple, il n'est sans doute pas nécessaire de faire en chaque nœud de l'arbre de *branch-and-bound* et pour chaque variable, la résolution de deux problèmes d'optimisation (MIN_k) et (MAX_k). Il pourrait être judicieux de travailler sur la conception d'un ordre de parcours des variables à réduire. En effet, en faisant les réductions dans un certain ordre, on peut diminuer le nombre d'opérations à effectuer pour obtenir le même niveau de contraction. Ceci pourrait grandement augmenter l'efficacité de notre algorithme et lui permettrait de traiter des instances plus grandes d'OPF.

	Résolution de $(OPF - L)$ et de $(OPF - Q)$ avec `RC-OPF` et `RC-OPF-Q`
PHASE 1 : Calcul de la meilleure reformulation quadratique convexe	1. Conception des familles de reformulations quadratiques équivalentes $(OPF - L_S)$ et $(OPF - Q_T)$ respectivement pour $(OPF - L)$ et $(OPF - Q)$, 2. Enoncé et démonstration des théorèmes permettant d'obtenir une reformulation optimale au sein des deux familles $(OPF - L_S)$ et $(OPF - Q_T)$. Ces reformulations sont optimales au sens où elles donnent ensuite la meilleure borne inférieure par résolution des relaxations respectivement $(\overline{OPF - L_S})$ et $(\overline{OPF - Q_T})$, 3. Extension au cas où l'on intègre les limites en courant aux problèmes $(OPF - L)$ et $(OPF - Q)$, 4. Démonstration que les reformulations obtenues sont aussi optimales dans le cas où l'on ajoute les inégalités de McCormick, donc en considérant les relaxations $(\overline{OPF - L_S - MCC})$ et $(\overline{OPF - Q_T - MCC})$ respectivement à la place des relaxations $(\overline{OPF - L_S})$ et $(\overline{OPF - Q_T})$.
PHASE 2 : Résolution de la reformulation avec un algorithme de *branch-and-bound*	1. Conception d'un algorithme de *branch-and-bound* s'appuyant sur la relaxation $(\overline{OPF - L_S - MCC})$ ou $(\overline{OPF - Q_T - MCC})$ pour résoudre la reformulation optimale $(OPF - L_S)$ ou $(OPF - Q_T)$, 2. Conception de trois méthodes différentes pour sélectionner la variable de branchement au sein de cet algorithme, 3. Implémentation des algorithmes de contraction OBBT-OPF et FBBT-OPF, 4. Explication de la synergie entre les contractions de type OBBT et la méthode de reformulation quadratique optimale, 5. Enoncé et démonstration des contractions non linéaires conçues pour l'OPF.

TABLE 5.9 – Mes contributions.

Bibliographie

[1] P. Belotti, J. Lee, L. Liberti, F. Margot et A. Wächter, “Branching and bounds tightening techniques for non-convex minlp,” *Optimization Methods & Software*, vol. 24, n°. 4-5, p. 597–634, 2009.

[2] W. A. Bukhsh, A. Grothey, K. I. M. McKinnon et P. A. Trodden, “Local solutions of the optimal power flow problem,” *IEEE Transactions on Power Systems*, vol. 28, n°. 4, p. 4780–4788, Nov 2013.

[3] R. D. Zimmerman, C. E. Murillo-Sanchez et R. J. Thomas, “Matpower : Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, n°. 1, p. 12–19, Feb 2011.

[4] PGLib-OPF, “Power grid lib - optimal power flow v18.08,” 2018. [En ligne]. Disponible : https://git.io/pglib-opf

[5] M. Ruiz, J. Maeght, A. Marié, P. Panciatici et A. Renaud, “A progressive method to solve large-scale ac optimal power flow with discrete variables and control of the feasibility,” p. 1–7, Aug 2014.

[6] B. Kocuk, S. S. Dey et X. A. Sun, “New Formulation and Strong MISOCP Relaxations for AC Optimal Transmission Switching Problem,” *ArXiv e-prints*, oct. 2015.

[7] A. Mitsos, “Global solution of nonlinear mixed-integer bilevel programs,” *Journal of Global Optimization*, vol. 47, n°. 4, p. 557–582, 2010. [En ligne]. Disponible : http://dx.doi.org/10.1007/s10898-009-9479-y

[8] J. Carpentier, “Contribution to the economic dispatch problem,” *Bulletin de la Societe Francaise des Electriciens*, vol. 3, n°. 8, p. 431–447, 1962.

[9] J. Lavaei et S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, n°. 1, p. 92–107, Feb 2012.

[10] S. Sojoudi et J. Lavaei, "Exactness of semidefinite relaxations for nonlinear optimization problems with underlying graph structure," *SIAM Journal on Optimization*, vol. 24, n°. 4, p. 1746–1778, 2014. [En ligne]. Disponible : http://dx.doi.org/10.1137/130915261

[11] C. Coffrin, H. L. Hijazi et P. Van Hentenryck, "Strengthening convex relaxations with bound tightening for power network optimization," dans *International Conference on Principles and Practice of Constraint Programming*. Springer, 2015, p. 39–57.

[12] A. Gopalakrishnan, A. U. Raghunathan, D. Nikovski et L. T. Biegler, "Global optimization of optimal power flow using a branch & bound algorithm," p. 609–616, Oct 2012.

[13] B. Kocuk, S. S. Dey et X. A. Sun, "Inexactness of sdp relaxation and valid inequalities for optimal power flow," *IEEE Transactions on Power Systems*, vol. 31, n°. 1, p. 642–651, Jan 2016.

[14] S. H. Low, "Convex Relaxation of Optimal Power Flow, Part I : Formulations and Equivalence," *ArXiv e-prints*, mai 2014.

[15] Z. Qiu, G. Deconinck et R. Belmans, "A literature survey of optimal power flow problems in the electricity market context," p. 1–6, March 2009.

[16] D. Bienstock et A. Verma, "Strong NP-hardness of AC power flows feasibility," *ArXiv e-prints*, déc. 2015.

[17] B. Zhang et D. Tse, "Geometry of feasible injection region of power networks," p. 1508–1515, 2011.

[18] S. Sojoudi et J. Lavaei, "Network topologies guaranteeing zero duality gap for optimal power flow problem," *submitted for publication*, 2011.

[19] J. Lavaei, D. Tse et B. Zhang, "Geometry of power flows and optimization in distribution networks," *IEEE Transactions on Power Systems*, vol. 29, n°. 2, p. 572–583, 2013.

[20] R. H. Byrd, J. Nocedal et R. A. Waltz, "Knitro : An integrated package for nonlinear optimization," p. 35–59, 2006. [En ligne]. Disponible : http://dx.doi.org/10.1007/0-387-30065-1_4

[21] "Ipopt." [En ligne]. Disponible : https://www.coin-or.org/Ipopt/documentation/

[22] C. Josz, S. Fliscounakis, J. Maeght et P. Panciatici, "AC Power Flow Data in MATPOWER and QCQP Format : iTesla, RTE Snapshots, and PEGASE," *ArXiv e-prints*, mars 2016.

[23] C. Coffrin et P. V. Hentenryck, “A linear-programming approximation of ac power flows,” *INFORMS Journal on Computing*, vol. 26, n°. 4, p. 718–734, 2014. [En ligne]. Disponible : http://dx.doi.org/10.1287/ijoc.2014.0594

[24] G. P. McCormick, “Computability of global solutions to factorable nonconvex programs : Part i — convex underestimating problems,” *Mathematical Programming*, vol. 10, n°. 1, p. 147–175, 1976. [En ligne]. Disponible : http://dx.doi.org/10.1007/BF01580665

[25] C. Coffrin, H. L. Hijazi et P. Van Hentenryck, “The qc relaxation : A theoretical and computational study on optimal power flow,” *IEEE Transactions on Power Systems*, vol. 31, n°. 4, p. 3008–3018, 2016.

[26] B. Kocuk, S. S. Dey et X. A. Sun, “Strong socp relaxations for the optimal power flow problem,” *Operations Research*, vol. 64, n°. 6, p. 1177–1196, 2016. [En ligne]. Disponible : http://dx.doi.org/10.1287/opre.2016.1489

[27] C. Bingane, M. F. Anjos et S. Le Digabel, “Tight-and-cheap conic relaxation for the ac optimal power flow problem,” *IEEE Transactions on Power Systems*, vol. 33, n°. 6, p. 7181–7188, 2018.

[28] B. Kocuk, S. S. Dey et X. A. Sun, “Matrix minor reformulation and socp-based spatial branch-and-cut method for the ac optimal power flow problem,” *arXiv preprint arXiv :1703.03050*, 2017.

[29] B. Ghaddar, J. Marecek et M. Mevissen, “Optimal power flow as a polynomial optimization problem,” *IEEE Transactions on Power Systems*, vol. 31, n°. 1, p. 539–546, Jan 2016.

[30] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre et C. L. DeMarco, “Implementation of a large-scale optimal power flow solver based on semidefinite programming,” *IEEE Transactions on Power Systems*, vol. 28, n°. 4, p. 3987–3998, Nov 2013.

[31] M. Fukuda, M. Kojima, K. Murota et K. Nakata, “Exploiting sparsity in semidefinite programming via matrix completion i : General framework,” *SIAM Journal on Optimization*, vol. 11, n°. 3, p. 647–674, 2001. [En ligne]. Disponible : http://dx.doi.org/10.1137/S1052623400366218

[32] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima et K. Murota, “Exploiting sparsity in semidefinite programming via matrix completion ii : implementation and numerical results,” *Mathematical Programming*, vol. 95, n°. 2, p. 303–327, 2003. [En ligne]. Disponible : http://dx.doi.org/10.1007/s10107-002-0351-9

[33] L. Vandenberghe et M. S. Andersen, “Chordal graphs and semidefinite optimization,” *Foundations and Trends® in Optimization*, vol. 1, n°. 4, p. 241–433, 2015. [En ligne]. Disponible : http://dx.doi.org/10.1561/2400000006

[34] J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” *SIAM Journal on Optimization*, vol. 11, n°. 3, p. 796–817, 2001.

[35] ——, “An explicit equivalent positive semidefinite program for nonlinear 0-1 programs,” *SIAM Journal on Optimization*, vol. 12, n°. 3, p. 756–769, 2002.

[36] ——, “Convexity in semialgebraic geometry and polynomial optimization,” *SIAM Journal on Optimization*, vol. 19, n°. 4, p. 1995–2014, 2009. [En ligne]. Disponible : http://dx.doi.org/10.1137/080728214

[37] C. Josz et D. K. Molzahn, “Moment/Sum-of-Squares Hierarchy for Complex Polynomial Optimization,” *ArXiv e-prints*, août 2015.

[38] C. Josz et D. K. Molzahn, “Lasserre hierarchy for large scale polynomial optimization in real and complex variables,” *SIAM Journal on Optimization*, vol. 28, n°. 2, p. 1017–1048, 2018.

[39] C. Josz, “Application of Polynomial Optimization to Electricity Transmission Networks,” *PhD. Thesis*, août 2016.

[40] M. Tawarmalani et N. V. Sahinidis, “A polyhedral branch-and-cut approach to global optimization,” *Mathematical Programming*, vol. 103, p. 225–249, 2005.

[41] C. Chen, A. Atamtürk et S. S. Oren, “A spatial branch-and-cut method for nonconvex qcqp with bounded complex variables,” *Mathematical Programming*, p. 1–29, 2017.

[42] N. V. Sahinidis, “Global optimization and constraint satisfaction : The branch-and-reduce approach,” p. 1–16, 2002.

[43] H. S. Ryoo et N. V. Sahinidis, “A branch-and-reduce approach to global optimization,” *Journal of Global Optimization*, vol. 8, n°. 2, p. 107–138, 1996.

[44] A. Billionnet et S. Elloumi, “Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem,” *Mathematical Programming*, vol. 109, n°. 1, p. 55–68, 2007. [En ligne]. Disponible : http://dx.doi.org/10.1007/s10107-005-0637-9

[45] A. Billionnet, S. Elloumi et A. Lambert, “Extending the qcr method to general mixed-integer programs,” *Mathematical Programming*, vol. 131, n°. 1, p. 381–401, 2012. [En ligne]. Disponible : http://dx.doi.org/10.1007/s10107-010-0381-7

[46] ——, “Exact quadratic convex reformulations of mixed-integer quadratically constrained problems,” *Mathematical Programming*, vol. 158, n°. 1, p. 235–266, 2016. [En ligne]. Disponible : http://dx.doi.org/10.1007/s10107-015-0921-2

[47] R. Fourer, D. M. Gay et B. W. Kernighan, “A modeling language for mathematical programming,” *Management Science*, vol. 36, n°. 5, p. 519–554, 1990.

[48] X.-M. Release, “Reference manual,” *Dash Associates*, 1999.

[49] “Artelys knitro - nonlinear optimization solver.” [En ligne]. Disponible : http://www.artelys.com/en/optimization-tools/knitro

[50] S. Gopinath, H. Hijazi, T. Weisser, H. Nagarajan, M. Yetkin, K. Sundar et R. Bent, “Towards closing the acopf chapter,” *arXiv preprint arXiv :1910.03716*, 2019.

[51] E. Dumon, M. Ruiz, H. Godard et J. Maeght, “Sdp resolution techniques for the optimal power flow with unit commitment,” dans *2017 IEEE Manchester PowerTech*. IEEE, 2017, p. 1–6.

Annexe A

Recherche d'instances difficiles

Comment calculer un gap potentiel

Une fois qu'un tirage a été réalisé sur les coûts de production active, on obtient une instance $(OPF - L)$ modifiée. On évalue son gap potentiel comme suit :

1. On résout la relaxation $(SDP - L)$ sur cette instance avec l'option SDPOPF de MATPOWER [3] (Code rédigé par Dan Molzahn [30]) avec les options suivantes pour obtenir une instance $(OPF - L)$:

   ```
   mpopt.sdp_pf.eps_r = -inf;         % Do not enforce a minimum resistance
   mpopt.sdp_pf.min_Pgen_diff = 0;    % Disable enforcing fixed value when small range for Pgen
   mpopt.sdp_pf.min_Qgen_diff = 0;    % Disable enforcing fixed value when small range for Qgen
   mpopt.sdp_pf.max_line_limit = inf; % Disable elimination of large line limits
   mpopt.sdp_pf.max_gen_limit  = inf; % Disable elimination of large generation limits
   ```

   Ces options sont expliquées plus en détail dans l'article de référence pour les jeux de données RTE et PEGASE [22] traitant d'instances $(OPF - L)$. Ce code utilise le modeleur YALMIP et nécessite un solveur de programmes SDP, on utilise ici SEDUMI. La tolérance d'optimalité est laissée par défaut à 10^{-9}.

   Si la condition de rang 1 est respectée par la solution de $(SDP - L)$, on note que la relaxation est exacte pour cette instance. On sait alors que son gap de dualité est nul, sinon on note que la relaxation $(SDP - L)$ n'est pas déclarée optimale.

2. On recherche une solution réalisable avec le solveur KNITRO [49, 20] avec MATPOWER. Cette solution nous fournit une borne supérieure. C'est avec elle que l'on calcule le gap potentiel comme défini plus haut. Si la condition de rang 1 était respectée par la relaxation $(SDP - L)$, on réalise quand même l'appel à KNITRO pour vérifier l'exactitude des bornes.

3. On recherche aussi une solution réalisable avec le solveur MIPS (MatLab Interior Point Solver) de MATPOWER, afin d'avoir de la redondance au cas où un solveur non linéaire échouerait.

On notera que KNITRO et MIPS résolvent la même formulation fournie par MATPOWER. C'est la formulation polaire de $(OPF - L)$. De plus, les deux solvers débutent avec le même point de départ fourni par MATPOWER. On utilise MIPS avec les options par défaut de MATPOWER, tandis qu'on utilise les options suivantes avec KNITRO :

```
xtol    0
feastol 10^{-10}
opttol  10^{-6}
maxit   1000
```

Application

Cette méthode a été utilisée pour tirer 1000 instances de chacune des 20 instances d'OPF suivantes :

— 6ww, 9, 14, 30, 39, 57, 89pegase, 118, 300 et 1354pegase que l'on trouve dans la bibliothèque MATPOWER [3].

— WB2, LMBM3, WB3, WB5, 9mod, 22loop, 39mod1, 39mod2, 118mod et 300mod qui sont décrites dans [2].

Pour chaque tirage, nous avons conservé la graine de laquelle il est issue, ainsi que la valeur de la relaxation $(SDP - L)$, des solutions obtenues avec KNITRO et MIPS, le gap potentiel évalué avec la solution de KNITRO et finalement si la relaxation $(SDP - L)$ a été déclarée exacte ou non. Les mêmes calculs de bornes ont aussi été réalisés sur l'instance initiale non modifiée.

La Table A.1 détaille les résultats obtenus instance par instance. Pour chacune d'entre elles, on indique :

— Le nombre d'instances aléatoires avec un gap potentiel au dessus de 1%.

— Le gap potentiel de la dixième instance ayant le plus grand gap potentiel supérieur à 1%. On l'appelle le seuil. S'il y a moins de 10 instances avec un gap potentiel supérieur à 1%, on donne le plus petit gap potentiel supérieur à 1%. Une valeur de seuil de 0 signifie que cette instance n'a fournit aucune instance aléatoire avec un gap potentiel, et donc un gap de dualité, supérieur à 1%.

Les deux instances WB2 et WB3 ne contiennent qu'un générateur, sur ces instances notre méthode

Instances	Gaps potentiels	Seuils	Commentaires
WB2	XX	XX	Aucune relaxation $(SDP - L)$ optimale, KNITRO et MIPS échouent à trouver des solutions admissibles. Toute instance a donc un gap potentiel non évalué.
LMBM3	0	0	$(SDP - L)$ toujours optimale.
WB3	0	0	$(SDP - L)$ toujours optimale.
WB5	268	25%	$(SDP - L)$ déclarée optimale dans 60% des cas. KNITRO et MIPS échouent sur 135 instances.
6ww	0	0	$(SDP - L)$ toujours optimale.
9	0	0	$(SDP - L)$ n'est jamais déclarée optimale. Gaps potentiels < 0.
9mod	1	1.7%	$(SDP - L)$ jamais déclarée optimale. Le seul gap potentiel au dessus de 1% est celui donné par l'instance non modifiée.
14	0	0	$(SDP - L)$ déclarée optimale dans 73 % des cas. Gaps potentiels $< 0.018\%$
22loop	9	46 %	Seules 9 $(SDP - L)$ sont déclarées non optimales.
30	0	0	$(SDP - L)$ n'est jamais déclarée optimale. Gaps potentiels $< 0.000002\%$.
39	0	0	92 % de $(SDP - L)$ déclarées non optimales. Gaps potentiels $< 0.042\%$.
39mod1	9	1.8%	$(SDP - L)$ n'est jamais déclarée optimale, KNITRO et MIPS échouent sur une instance.
39mod2	9	1.8 %	$(SDP - L)$ n'est jamais déclarée optimale, KNITRO et MIPS échouent sur une instance.
57	11	1.2 %	72 % de $(SDP - L)$ déclarées non optimales.
89pegase	0	0	0.4 % de $(SDP - L)$ déclarées non optimales. Gaps potentiels < 0.
118	0	0	96% de $(SDP - L)$ déclarées non optimales. Gaps potentiels $< 0.032\%$.
118mod	0	0	37% de $(SDP - L)$ déclarées non optimales. Gaps potentiels $< 0.032\%$.
300	0	0	$(SDP - L)$ n'est jamais déclarée optimale. Gaps potentiels $< 0.29\%$.
300mod	277	4.05%	$(SDP - L)$ n'est jamais déclarée optimale.
1354pegase	0	0	$(SDP - L)$ n'est jamais déclarée optimale. Gaps potentiels $< 0.051\%$.

TABLE A.1 – Résultats par instance. 1000 tirages par instance.

ne modifie pas la solution optimale, mais ne fait que modifier sa valeur. Notre méthode de modifications d'instances n'est donc pas pertinente pour ces deux instances. Par conséquent, on écarte ces deux instances de notre analyse. Pour les autres, il y a 6 instances (parmi 18) où l'on obtient des gaps potentiels supérieurs à 1 %, qui est notre valeur cible. Ces instances sont : WB5, 22loop, 39mod1, 39mod2, 57 et 300mod.

De manière générale, on remarque que la relaxation $(SDP - L)$ est souvent déclarée non exacte [1] même si les gaps potentiels obtenus, qui bornent le gap de dualité, sont inférieurs à notre cible. C'est le cas pour une grande majorité des instances générées à partir des 8 instances : 9, 9mod, 30, 39, 118, 118mod, 300 et 1354pegase. Il arrive que le gap potentiel soit négatif, cela signifie que la solution fournie par KNITRO a une valeur inférieure à celle de la relaxation $(SDP - L)$. Ces gaps potentiels sont inférieurs à 10^{-6} en valeur absolue. Les écarts numériques dues aux tolérances de faisabilité et d'optimalité nous semblent à même d'expliquer ce phénomène.

Concernant l'efficacité de la recherche de bornes supérieures, on remarque que globalement KNITRO converge systématiquement vers une solution admissible. Sauf pour WB2, 13.5% des cas sur WB5 et 1 une fois sur 1000 pour 39mod1 et 39mod2. Si l'on compare KNITRO et MIPS, on note que MIPS échoue toujours là où KNITRO échoue. De plus, MIPS échoue aussi dans 3254 cas supplémentaires provenant des instances WB5, 9mod, 39mod1, 39mod2, 118mod, 300, 300mod. On peut aussi comparer la qualité des bornes obtenues, dans la grande majorité des cas, les bornes sont très proches (écart relatif inférieur à 10^{-6}). Notons tout de même les éléments suivants :

— Dans 564 cas, sur 9mod, l'écart relatif est d'au moins 1% en défaveur de MIPS.
— Dans 39 cas, sur 39mod1 et 39mod2, l'écart relatif est d'au moins 1% en défaveur de MIPS.
— Dans 42 cas, sur 300mod, l'écart relatif est d'au moins 1% en défaveur de MIPS.

Dans la suite, nous nous intéressons aux instances générées aléatoirement, présentant les dix meilleurs gaps potentiels au delà de 1%, à partir des 6 instances où notre méthode a obtenu des gaps potentiels au delà de 1%. Pour chacune de ces instances, nous lançons notre algorithme de *branch-and-bound*, afin d'affiner le gap potentiel en s'approchant de l'optimalité et donc de confirmer les instances obtenues présentant des gaps de dualité intéressant. En effet, L'algorithme de *branch-and-bound* fait

1. C'est-à-dire que la matrice obtenue en résolvant la relaxation $(SDP - L)$ n'est pas de rang 1.

progressivement augmenter la borne inférieure, initialement obtenue par la relaxation $(SDP - L)$. Il permet également d'obtenir d'autres solutions réalisables. Notre algorithme `RC-OPF` a été lancé avec le paramètrage suivant :

- Pas de limite sur le nombre de nœuds dans l'arbre de *branch-and-bound*,
- Le solveur utilisé pour résoudre les relaxations aux nœuds est le simplexe dual de XPRESS,
- On recherche des solutions admissibles (et donc des bornes supérieures) à partir du problème reformulé non convexe avec l'algorithme de points intérieurs de KNITRO (en coordonnées cartésiennes donc),
- La recherche de solutions admissibles a lieu une fois tous les 3 nœuds explorés,
- Les contractions de bornes avec le presolve AMPL de la méthode FBBT-OPF sont désactivées,
- L'algorithme de contraction de bornes OBBT-OPF est désactivé,
- Les autres options sont laissées par défaut, on rappelle les valeurs par défaut des plus sensibles d'entre elles :
  - Critère d'arrêt de l'algorithme : Ecart relatif de $10^{-5} = 0.001\%$ entre bornes inférieures et supérieures,
  - Temps de calcul limite : 3600s = 1h,
  - Paramètre de branchement $\delta = 0.5$,
  - Tolérance relative sur la faisabilité et l'optimalité pour KNITRO : 10^{-6}.

Résultats détaillés avec l'algorithme de *branch-and-bound*

On présente les résultats détaillées pour les 6 instances fournissant des gaps potentiels au delà de 1 %. Pour chacune de ces instances, on a lancé notre algorithme de *branch-and-bound* sur les 10 instances modifiées aléatoirement présentant les plus grands gaps potentiels au-delà de 1%.

WB5

Rappelons que l'instance WB5 dispose d'un gap initial de plus de 16 %, il n'est donc guère étonnant de trouver facilement des gaps élevés à partir de cette instance. Les Figures A.1 et A.2 montrent l'évolution des bornes supérieures et inférieures au cours de notre algorithme de *branch-and-bound* pour deux instances aléatoires. On rappelle aussi les bornes obtenues grâce à la relaxation $(SDP - L)$ et à KNITRO (MIPS échoue ici à trouver une borne). La Figure A.1 correspond à l'instance aléatoire

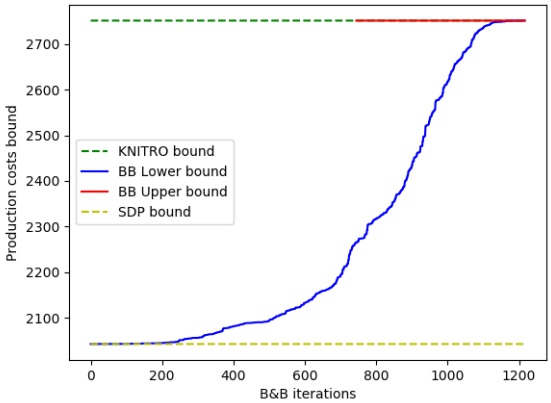

FIGURE A.1 – Evolution des bornes sur l'instance aléatoire 137 depuis WB5

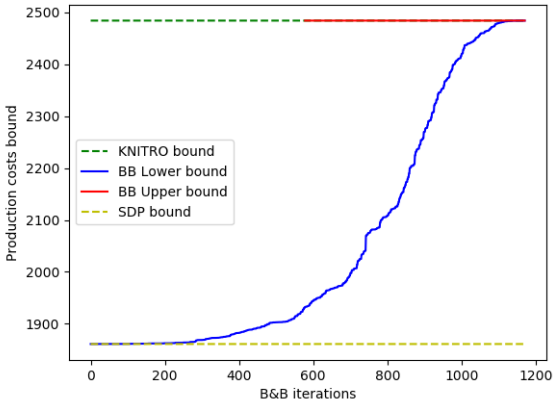

FIGURE A.2 – Evolution des bornes sur l'instance aléatoire 195 depuis WB5

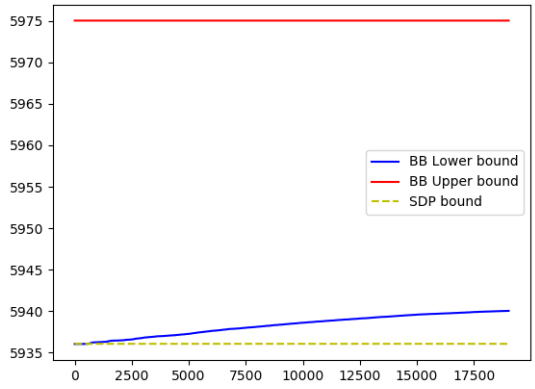

FIGURE A.3 – Evolution des bornes sur l'instance aléatoire 236 depuis 22loop

137 qui présente le plus grand gap potentiel. La Figure A.2 correspond à l'instance aléatoire 195 qui présente le dixième plus grand gap potentiel. On observe que la borne inférieure au cours de l'algorithme remonte jusqu'à la solution initiale obtenue par KNITRO. On a donc prouvé pour ces instances que le gap potentiel correspondait bien au gap de dualité, ce qui conduit à valider l'obtention de nouvelles instances avec des gaps de dualité importants. Cependant on cherche plutôt des gaps de dualité sur des instances plus grandes, plus réalistes que WB5. De plus, cette instance ne disposant que de deux générateurs, notre méthode de modifications des coûts ne permet pas de générer beaucoup de configurations différentes. En effet il n'y a que 100 combinaisons de prix (10 valeurs possibles pour chaque générateur), tout en sachant que des configurations différentes mènent peut être au même plan de tension optimal.

22loop

Pour cette instance, on a repéré 9 instances aléatoires avec des gaps potentiels au-dessus de 1%, qui avait des valeurs plutôt élevés (toutes au dessus de 45%). On pouvait faire l'hypothèse que ces gaps potentiels étaient bien plus élevés que le gap de dualité à cause de la convergence de KNITRO et MIPS vers des optima locaux non globaux. Le lancement de notre *branch-and-bound* sur ces 9 instances modifiées a permis de confirmer cela. En effet les nouveaux gaps potentiels obtenus sont tous inférieurs à 0.7%. Ces instances ne présentent donc pas de gap de dualité suffisament intéressant.

À titre d'informations, la Figure A.3 présente l'évolution des bornes au cours de notre algorithme,

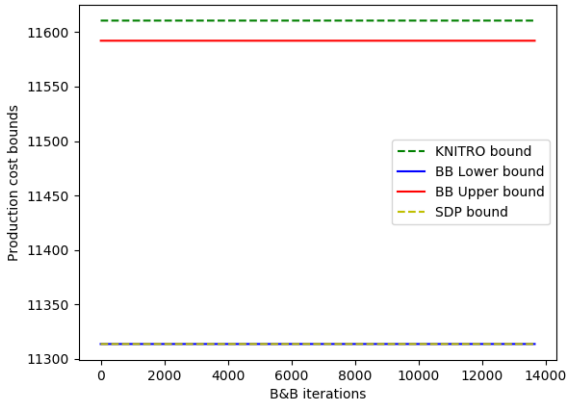

FIGURE A.4 – Evolution des bornes sur l'instance aléatoire 30 depuis 39mod1

en plus de la borne de la relaxation $(SDP - L)$ initiale, pour l'instance présentant le plus grand gap potentiel.

39mod1 et 39mod2

Dans notre modélisation $(OPF - L)$ consistant à ignorer les coûts quadratiques en puissance active, les instances 39mod1 et 39mod2 sont identiques à une constante près. En effet, 39mod2 est la même instance que 39mod1, mais où les coûts quadratiques et constants ont été supprimés. Ces deux instances fournissent alors les mêmes instances modifiées. Dans la suite, on se limite à l'étude de 39mod1.

On obtient des résultats différents pour chacune des 9 instances aléatoires générées à partir de 39mod1 ayant des gaps potentiels au-dessus de 1%. On résume ces résultats dans la Table A.2. Dans cette table, chaque ligne correspond à une instance aléatoire et renseigne le nouveau gap potentiel obtenu. Les bornes inférieures n'évoluent pas significativement au cours de l'algorithme.

Les Figures A.4 à A.12 montrent l'évolution des bornes sur ces 9 instances. Contrairement à ce qui advient pour WB5, on est ici incapable de certifier que les gaps potentiels sont égaux aux gaps de dualité. Néanmoins ils en sont la meilleure approximation connue. L'appel à l'algorithme de *branch-and-bound* a permis d'éliminer 5 instances qui passent en dessous du seuil de 1% de gap potentiel. Pour les instances 30, 211, 437 et 643 le doute persiste quant à l'existence d'un gap de dualité de plus de 1%.

Graine	Gap potentiel	Commentaires
30	2.4%	La borne supérieure fournie par `RC-OPF` est en dessous de celle initialement fournie par KNITRO et reste constante.
211	4%	La borne supérieure fournie par `RC-OPF` s'améliore rapidement et reste au dessus de celle initialement fournie par KNITRO. À noter que pour cette instance, la solution de la relaxation en un nœud de l'arbre de *branch-and-bound* fournit une solution admissible pour le problème originel vérifiant les contraintes à 10^{-3} près (en relatif) et donnant un gap potentiel de 0.3%.
437	2.3%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO et passe rapidement en dessous.
537	0.12%	La borne supérieure fournie par `RC-OPF` est en dessous de celle initialement fournie par KNITRO et reste constante.
643	3.0%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO puis rejoint tardivement celle ci.
712	0.2%	La borne supérieure fournie par `RC-OPF` démarre en dessous de celle initialement fournie par KNITRO puis diminue rapidement.
766	0.4%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO puis passe rapidement en dessous avec un nouveau gap potentiel inférieur à 1%.
776	0.1%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO et la rejoint rapidement. À noter que sur cette instance, on a obtenu une borne supérieure KNITRO différente entre l'étape de recherche des instances à gap potentiel intéressant et la génération des graphes d'évolution des bornes. Entre ces deux étapes, ce sont de versions différentes de KNITRO qui ont été utilisées pour l'appel via MATPOWER.
924	0.1%	Semblable au cas 766.

Table A.2 – Résultats pour 9 instances aléatoires obtenues à partir de 39mod1.

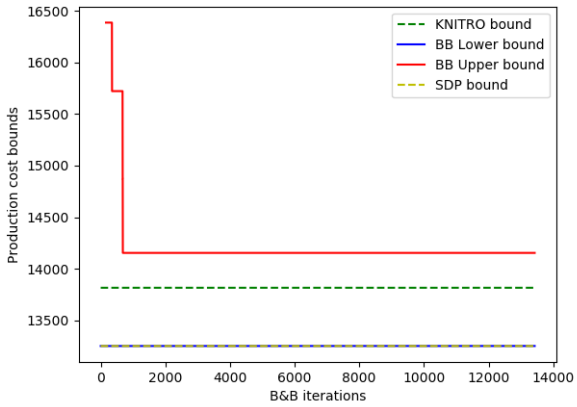

FIGURE A.5 – Evolution des bornes sur l'instance aléatoire 211 depuis 39mod1

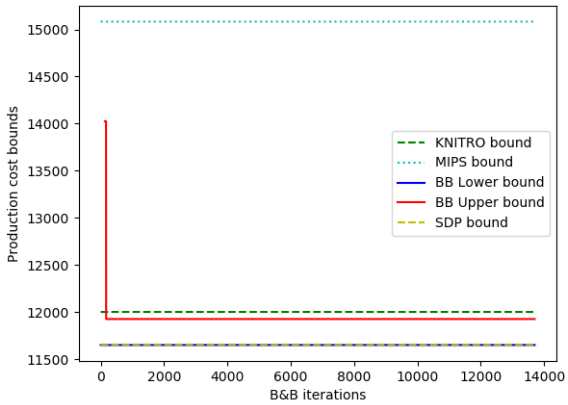

FIGURE A.6 – Evolution des bornes sur l'instance aléatoire 437 depuis 39mod1

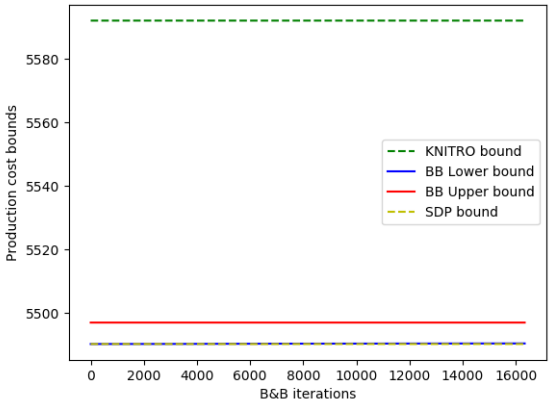

FIGURE A.7 – Evolution des bornes sur l'instance aléatoire 537 depuis 39mod1

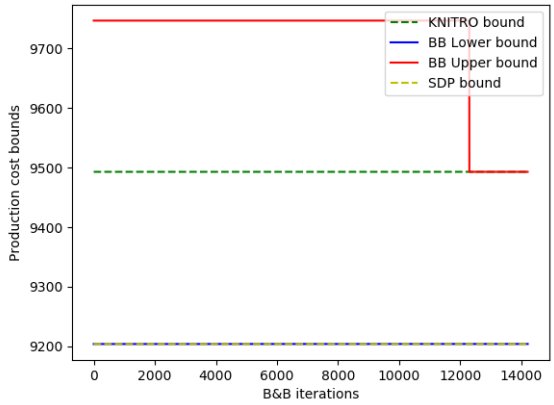

FIGURE A.8 – Evolution des bornes sur l'instance aléatoire 643 depuis 39mod1

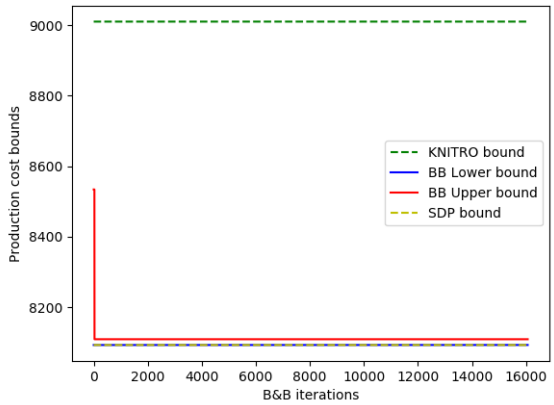

FIGURE A.9 – Evolution des bornes sur l'instance aléatoire 712 depuis 39mod1

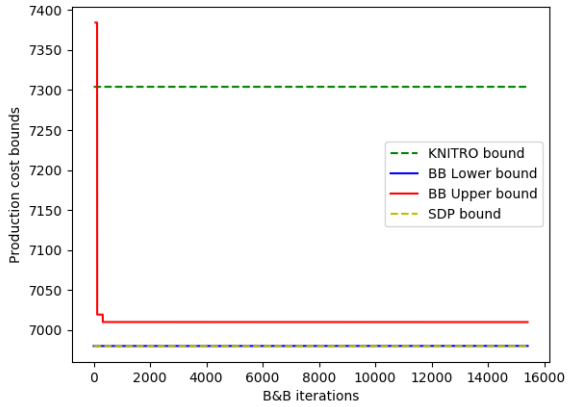

FIGURE A.10 – Evolution des bornes sur l'instance aléatoire 766 depuis 39mod1

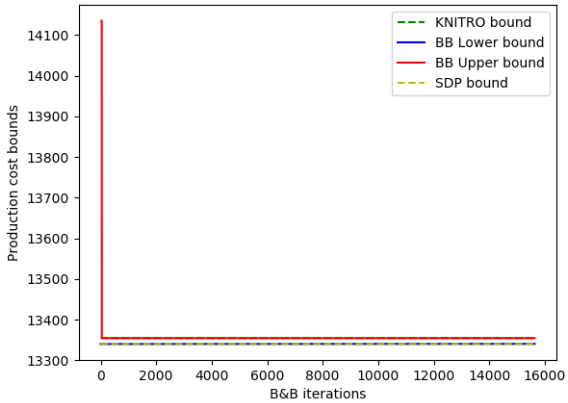

FIGURE A.11 – Evolution des bornes sur l'instance aléatoire 776 depuis 39mod1

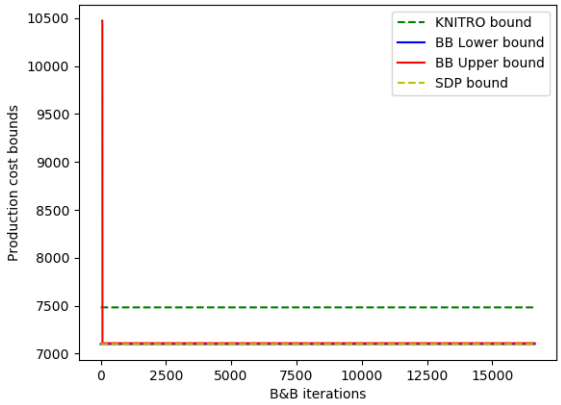

FIGURE A.12 – Evolution des bornes sur l'instance aléatoire 924 depuis 39mod1

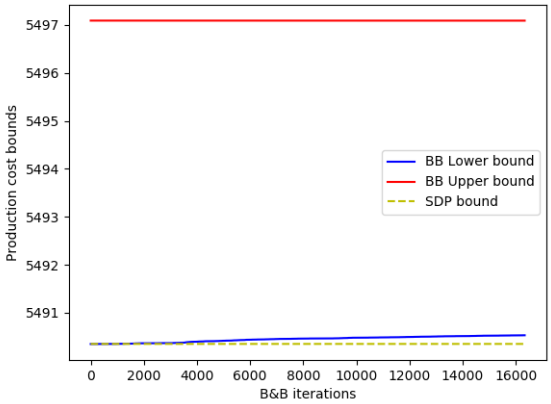

FIGURE A.13 – Evolution des bornes sur l'instance aléatoire 537 depuis 39mod1

La capacité de l'algorithme de *branch-and-bound* à trouver d'autres solutions admissibles, meilleures que celle obtenue par KNITRO, au cours de son execution renforce l'intérêt pour cette méthode de résolution exacte. Cela permet aussi de relativiser l'enjeu de prouver l'optimalité de la solution renvoyée par MATPOWER via KNITRO en faisant remonter une borne inférieure.

Au sujet de l'évolution de la borne inférieure au cours de l'algorithme `RC-OPF`, à titre d'illustration, on peut regarder la Figure A.13. Pour cette Figure, le gap entre la borne supérieure et la borne $(SDP - L)$ est de 0.12%.

57

Pour les 10 instances construites à partir de 57, on observe un comportement uniforme illustrée par la figure A.14. À savoir que l'algorithme de *branch-and-bound* retrouve la solution de KNITRO, et ne l'améliore pas, tandis que la borne inférieure n'évolue pas et reste bloquée à la valeur de la relaxation $(SDP - L)$. En augmentant le temps limite de calcul à 12h, on obtient le même résultat. Dès lors, on ne peut pas certifier que le gap potentiel est bien le gap de dualité. Néanmoins, comme on échoue systématiquement à trouver de nouvelles bornes supérieures, on peut se dire que l'enjeu ici réside plutôt dans la remontée dans la borne inférieure. Les 10 instances sélectionnées sur leur gap potentiel sont donc de bonnes candidates pour évaluer de futurs méthodes de branchement.

Graine	Gap potentiel	Commentaires
31	1.8%	La borne supérieure fournie par `RC-OPF` commence au dessus de celle initialement fournie par KNITRO, passe rapidement en dessous puis diminue progressivement.
45	3.0%	La borne supérieure fournie par `RC-OPF` est en dessous de celle initialement fournie par KNITRO et reste constante.
242	2.7%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO et passe rapidement en dessous.
437	3.1%	La borne supérieure fournie par `RC-OPF` démarre en dessous de celle initialement fournie par KNITRO puis diminue rapidement.
520	4.5 %	La borne supérieure fournie par `RC-OPF` démarre au même niveau que celle initialement fournie par KNITRO puis diminue.
536	3.1%	Similaire à 437.
625	4.2%	La borne supérieure fournie par `RC-OPF` démarre au même niveau que celle initialement fournie par KNITRO puis diminue rapidement.
650	4.2%	La borne supérieure fournie par `RC-OPF` commence au dessus de celle initialement fournie par KNITRO et la rejoint rapidement.
948	2.4%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO puis passe rapidement en dessous.
980	2.5%	La borne supérieure fournie par `RC-OPF` démarre au dessus de celle initialement fournie par KNITRO, diminue progressivement et finit en dessous.

TABLE A.3 – Résultats pour 10 instances aléatoires obtenues à partir de 300mod.

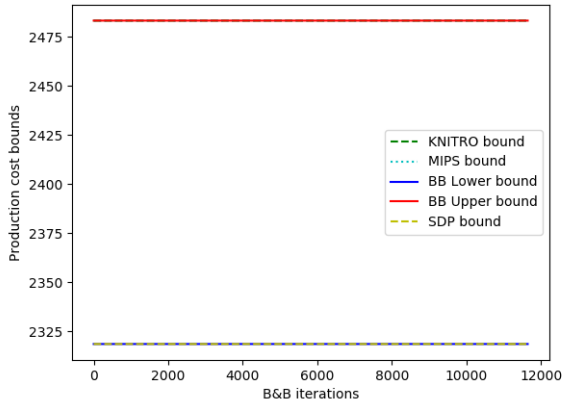

FIGURE A.14 – Evolution des bornes sur l'instance aléatoire 628 depuis 57

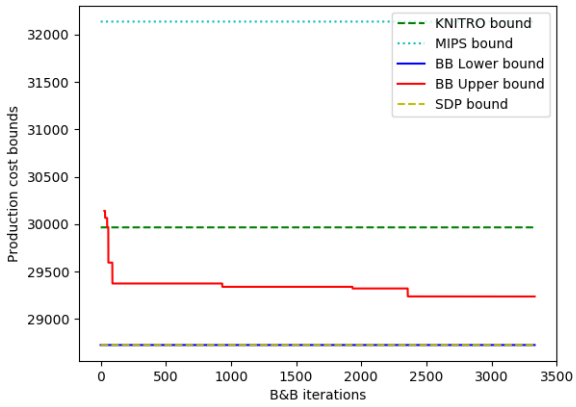

FIGURE A.15 – Evolution des bornes sur l'instance aléatoire 31 depuis 300mod

300mod

Les instances provenant de 300mod sont très diverses, comme celles provenant de 39mod1. De même que ci-dessus, nous détaillons les résultats instance par instance dans la Table A.3, puis nous affichons tous les graphes d'évolution des bornes (Figures A.15 à A.24).

On observe que dans 9 cas sur 10, on fait diminuer le gap potentiel avec notre algorithme en déterminant une meilleure solution réalisable. Dans le dernier cas, on obtient la même borne. Dans tous les cas, on échoue à faire remonter la borne inférieure. 9 gaps potentiels sont donc réduits et tous les gaps potentiels restent au dessus de notre cible de 1%. Là encore il n'y a pas de certitudes quant à

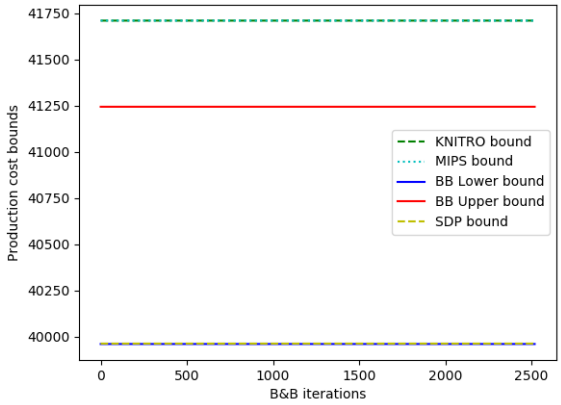

FIGURE A.16 – Evolution des bornes sur l'instance aléatoire 45 depuis 300mod

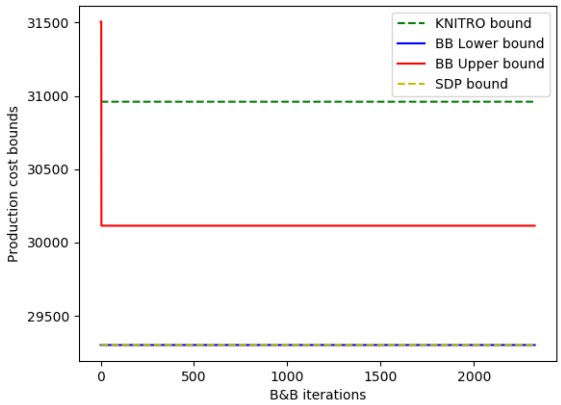

FIGURE A.17 – Evolution des bornes sur l'instance aléatoire 242 depuis 300mod

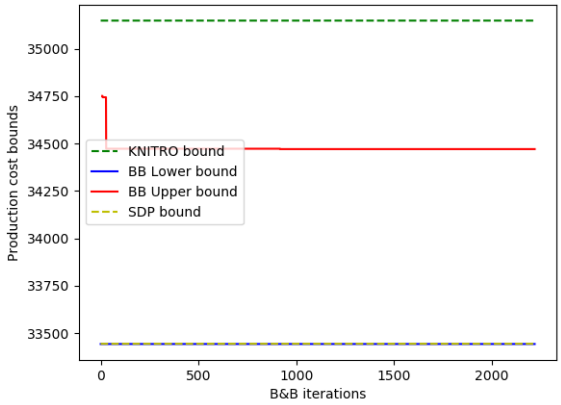

FIGURE A.18 – Evolution des bornes sur l'instance aléatoire 437 depuis 300mod

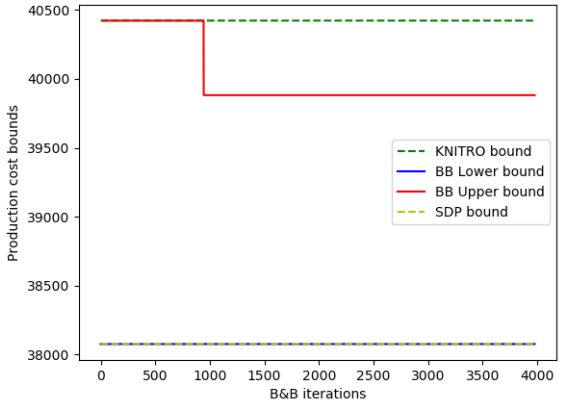

FIGURE A.19 – Evolution des bornes sur l'instance aléatoire 520 depuis 300mod

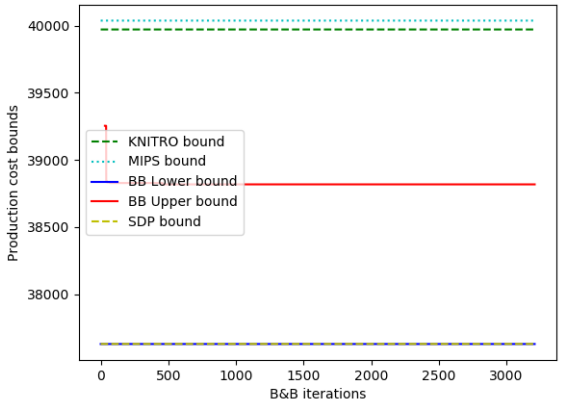

FIGURE A.20 – Evolution des bornes sur l'instance aléatoire 536 depuis 300mod

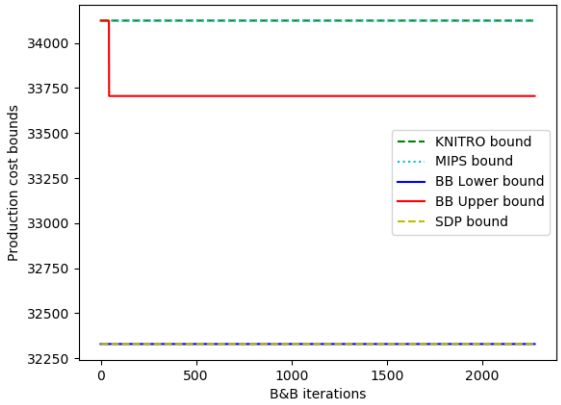

FIGURE A.21 – Evolution des bornes sur l'instance aléatoire 625 depuis 300mod

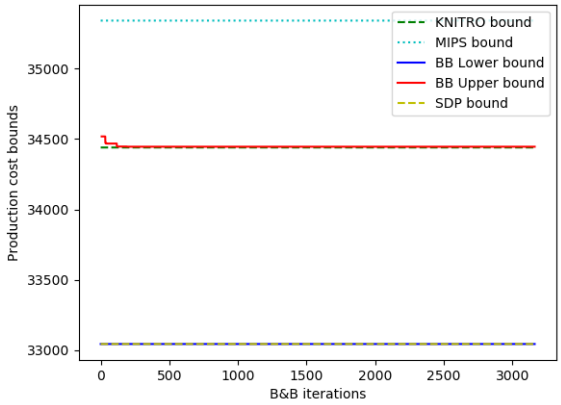

FIGURE A.22 – Evolution des bornes sur l'instance aléatoire 650 depuis 300mod

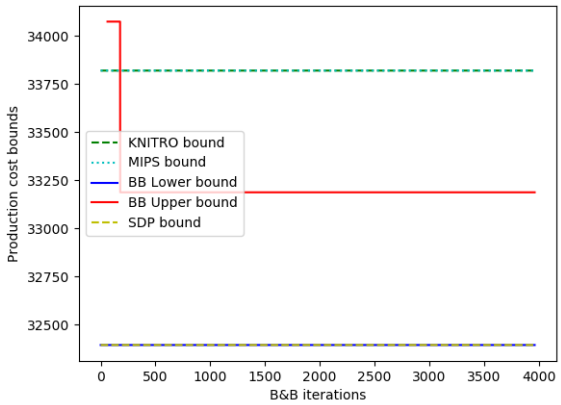

FIGURE A.23 – Evolution des bornes sur l'instance aléatoire 948 depuis 300mod

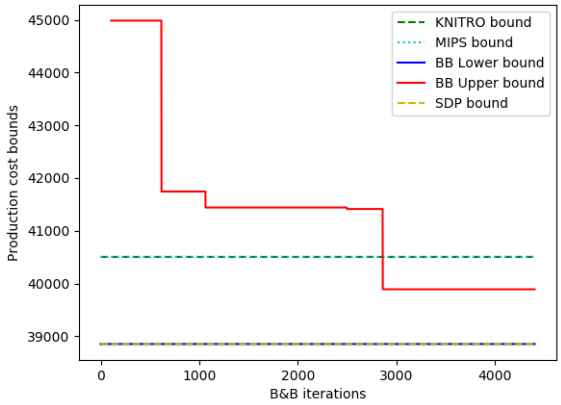

FIGURE A.24 – Evolution des bornes sur l'instance aléatoire 980 depuis 300mod

l'écart entre nos gaps potentiels et le gap de dualité, mais au vu de nos connaissances, ces 10 instances restent de bonnes candidates pour tester notre algorithme de *branch-and-bound* `RC-OPF`.

Hadrien GODARD

Résolution exacte du problème de l'Optimisation des Flux de Puissance

Résumé : Cette thèse a pour objet la résolution exacte d'un problème d'optimisation des flux de puissance (OPF) dans un réseau électrique. Dans l'OPF, on doit planifier la production et la répartition des flux de puissances électriques permettant de couvrir, à un coût minimal, la consommation en différents points du réseau. Trois variantes du problème de l'OPF sont étudiées dans ce manuscrit. Nous nous concentrerons principalement sur la résolution exacte des deux problèmes $(OPF - L)$ et $(OPF - Q)$, puis nous montrerons comment notre approche peut naturellement s'étendre à la troisième variante $(OPF - UC)$. Cette thèse propose de résoudre ces derniers à l'aide d'une méthode de reformulation que l'on appelle RC-OPF. La contribution principale de cette thèse réside dans l'étude, le développement et l'utilisation de notre méthode de résolution exacte RC-OPF sur les trois variantes d'OPF. RC-OPF utilise également des techniques de contractions de bornes, et nous montrons comment ces techniques classiques peuvent être renforcées en utilisant des résultats issus de notre reformulation optimale.

Mots clés : Réseaux-Électriques Énergie, Optimisation des Flux de Puissance, Optimisation non-Linéaire, Reformulation Quadratique Convexe, Programmation Quadratique, Programmation Semi-Définie, Réduction de bornes.

Abstract : Alternative Current Optimal Power Flow (ACOPF) is naturally formulated as a non-convex problem. In that context, solving (ACOPF) to global optimality remains a challenge when classic convex relaxations are not exact. We use semidefinite programming to build a quadratic convex relaxation of (ACOPF). We show that this quadratic convex relaxation has the same optimal value as the classical semidefinite relaxation of (ACOPF) which is known to be tight. In that context, we build a spatial branch-and-bound algorithm to solve (ACOPF) to global optimality that is based on a quadratic convex programming bound.

Keywords : Electrical-Network Energy, Optimal Power Flow, Non-Linear Optimization, Quadratic Convex Reformulation, Quadratic Programming, Semi Definite Programming, Bound Tightening.