Mathematical contributions for the optimization and regulation of electricity production

Benjamin Heymann

NNT : 2016SACLX052

Thèse de doctorat de l'Université Paris-Saclay préparée á l'Ecole polytechnique

Ecole doctorale n°574
Ecole doctorale de mathématiques Hadamard
Spécialité de doctorat : Mathématiques Appliquées

par

M. Benjamin Heymann

Contributions mathématiques pour la régulation et l'optimisation de la production d'électricité

Thèse présentée et soutenue à l'Ecole polytechnique, le 23 Septembre 2016.

Composition du Jury :

M.	Didier Aussel	Professeur Université de Perpignan	(Rapporteur)
M.	Frédéric Bonnans	Directeur de recherche Inria et Ecole polytechnique	(Directeur)
M.	Michel De Lara	Professeur Ecole des Ponts ParisTech	(Examinateur)
M.	Emmanuel Gobet	Professeur Ecole polytechnique	(Président du jury)
M.	Roger Guesnerie	Professeur émérite Collège de France	(Examinateur)
M.	René Henrion	Professeur Weierstrass Institute	(Rapporteur)
M.	Alejandro Jofré	Professeur Universidad de Chile	(Co-directeur)

We are like tenant farmers chopping down the fence around our house for fuel when we should be using Nature's inexhaustible sources of energy–sun, wind and tide. I'd put my money on the sun and solar energy. What a source of power! I hope we don't have to wait until oil and coal run out before we tackle that.

Thomas A. Edison, *as quoted in Uncommon Friends : Life with Thomas Edison, Henry Ford, Harvey Firestone, Alexis Carrel and Charles Lindbergh (1987) by James Newton*

Version: November 26, 2016

Remerciements

Je souhaite avant tout remercier mes directeurs Frédéric Bonnans et Alejandro Jofré. Je leur suis trés reconnaissant pour ce qu'ils m'ont enseigné et pour le temps et l'enthousiasme qu'ils ont consacrés à mon travail de recherche.

Je tiens à remercier particulièrement Didier Aussel et René Henrion d'avoir accepté d'être les rapporteurs de ma thèse. Je tiens aussi à les remercier pour leurs suggestions et commentaires. Je suis très reconnaissant envers Roger Guesnerie, Michel De Lara et Emmanuel Gobet, que j'ai le privilège d'avoir parmi les membres de mon jury. Je souhaite aussi remercier Roger Guesnerie pour l'entretien qu'il m'a accordé au Collège de France.

Je souhaite exprimer ma gratitude envers ceux avec qui j'ai eu l'occation de collaborer ou d'échanger. J'adresse en très grand merci à Pierre Martinon pour ces temps de travail toujours enrichissants, ainsi que pour les innombrables conseils. Merci à Francisco Silva, avec qui j'ai beaucoup lu et beaucoup appris. Je tiens aussi à remercier Justina Gianatti, Daphné Giorgi et Olivier Tissot, ainsi que Philip Thompson et Faisal Wahid. Merci à nos collaborateurs de l'Universidad de Chile Guillermo Jiménez Estévez et Fernando Lanas. Enfin, merci à Michel de Lara et Pierre Carpentier, et plus généralement aux membres du CERMICS de l'Ecole des Ponts pour les nombreux échanges.

Je suis très reconnaissant envers Stéphane Gaubert, Robert Phillips et Harborne W. Stuart, professeurs à Polytechnique et à Columbia University, dont les enseignements ont été pour moi des révélations.

Je voudrais adresser mes remerciements aux équipes administratives en particulier à Jessica Gameiro pour l'Inria, à Nasséra Naar et Alexandra Noiret pour le CMAP, et à Maria Inés Rivera pour le CMM. Merci aussi à Sylvain Ferrand pour son aide précieuse. Je remercie également Francois Alouges et Matthieu Aussal pour leur accueil au sein de l'aventure Gimbal.

Je souhaite remercier Thomas Siebel et la fondation Siebel Scholar pour la reconnaissance dont j'ai bénéficié en début de troisième année. Merci aussi à Jacques Biot, Frank Pacard et Léo Liberti qui ont beaucoup oeuvré pour rendre ceci possible.

Cette thèse doit beaucoup au soutien de mes proches ! Je souhaite en particulier remercier les Taupins, Vincent, Juliette, Stéphane, Patrick, Anne-Lise, Sarah, Noémie et Esther. Un grand merci à mes grands-parents, dont les encouragements me poussent depuis que j'ai l'âge d'apprendre, à mes parents et à mes beaux-parents. Enfin, merci à Laure, qui est sans aucun doute la première endocrinologue à avoir saisi le lien entre les brassards à souris et la théorie de la commande optimale, et qui a aussi relu l'intégralité de mon travail.

Le travail résumé dans cette thèse est dédié à la mémoire de Théo Heymann. Que son souvenir soit une bénédiction.

École doctorale de mathématiques Hadamard (EDMH)

Titre : Contributions mathématiques pour la régulation et l'optimisation de la production d'électricité

Keywords : électricité, micro réseau, marchés en réseaux, optimisation, vieillissement, mécanismes d'incitation.

Résumé : Nous présentons notre contribution sur la régulation et l'optimisation de la production d'électricité.

La première partie concerne l'optimisation de la gestion d'un micro réseau. Nous formulons le programme de gestion comme un problème de commande optimale en temps continu, puis nous résolvons ce problème par programmation dynamique à l'aide d'un solveur développé dans ce but : BocopHJB. Nous montrons que ce type de formulation peut s'étendre à une modélisation stochastique. Nous terminons cette partie par l'algorithme de poids adaptatifs, qui permet une gestion de la batterie du micro réseau intégrant le vieillissement de celle-ci. L'algorithme exploite la structure à deux échelles de temps du problème de commande.

La seconde partie concerne des modèles de marchés en réseaux, et en particulier ceux de l'électricité. Nous introduisons un mécanisme d'incitation permettant de diminuer le pouvoir de marché des producteurs d'énergie, au profit du consommateur. Nous étudions quelques propriétés mathématiques des problèmes d'optimisation rencontrés par les agents du marché (producteurs et régulateur). Le dernier chapitre étudie l'existence et l'unicité des équilibres de Nash en stratégies pures d'une classe de jeux Bayésiens à laquelle certains modèles de marchés en réseaux se rattachent. Pour certains cas, un algorithme de calcul d'équilibre est proposé.

Une annexe rassemble une documentation sur le solveur numérique BocopHJB.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Title : Mathematical contributions for the optimization and regulation of electricity production

Keywords : electricity, microgrid, network markets, optimization, aging, mechanism design, auctions.

Abstract : We present our contribution on the optimization and regulation of electricity production.

The first part deals with a microgrid Energy Management System (EMS). We formulate the EMS program as a continuous time optimal control problem and then solve this problem by dynamic programming using BocopHJB, a solver developed for this application. We show that an extension of this formulation to a stochastic setting is possible. The last section of this part introduces the adaptative weights dynamic programming algorithm, an algorithm for optimization problems with different time scales. We use the algorithm to integrate the battery aging in the EMS.

The second part is dedicated to network markets, and in particular wholesale electricity markets. We introduce a mechanism to deal with the market power exercised by electricity producers, and thus increase the consumer welfare. Then we study some mathematical properties of the agents' optimization problems (producers and system operator). In the last chapter, we present some pure Nash equilibrium existence and uniqueness results for a class of Bayesian games to which some networks markets belong. In addition we introduce an algorithm to compute the equilibrium for some specific cases.

We provide additional information on BocopHJB (the numerical solver developed and used in the first part of the thesis) in the appendix.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Contents

Chapter 1

Introduction

This introduction, the only chapter of the thesis written in French, presents in broad terms the whole content of the manuscript.

Contents

1.1 Présentation générale

Ce manuscrit rassemble le travail des trois années de thèse que j'ai effectuées à l'Ecole polytechnique au sein de l'équipe Inria COMMANDS, sous la direction de Frédéric Bonnans, co-encadré par Alejandro Jofré de l'Universidad de Chile. Au cours de cette thèse, j'ai effectué trois séjours de plusieurs mois à Santiago du Chili pour travailler avec Alejandro Jofré et une équipe du centro de Energía. Mon travail porte sur certaines questions d'optimisation liées à la production d'électricité. Chacune des deux parties de cette thèse en aborde un aspect différent. L'annexe quant à elle apporte quelques informations complémentaires sur BOCOPHJB, le solveur numérique de contrôle optimal au développement duquel j'ai pu participer.

La première partie de cette thèse porte sur l'optimisation de la production d'électricité par un micro-réseau. Dans les deux premiers chapitres de cette partie, nous formulons le problème d'optimisation du micro-réseau comme un problème de commande optimale. Le micro-réseau (microgrid en anglais) est un paradigme de production et de distribution d'électricité décentralisées. Cette idée peut sembler dans un premier temps étonnante, car le bon sens recommanderait plutôt a priori de centraliser pour faire des économies d'échelle et prendre des décisions qui soient optimales globalement. Néanmoins ce concept suscite un intérêt croissant dans le domaine [74]. L'étude des avantages et inconvénients du micro-réseau en tant que moyen de production d'électricité n'entre pas dans ce projet de recherche.

Observons tout d'abord qu'il existe un scénario où l'intérêt du micro-réseau est indéniable : quand il ne s'agit pas d'un choix mais d'une nécessité. Ainsi les deux chapitres suivants portent sur l'optimisation d'un modèle de micro-réseau construit à partir d'un cas réel : le projet du village de Huatacondo dans le désert de l'Atacama, au Chili[1]. Ce village, parce qu'il est géographiquement isolé, n'est pas connecté au reste du réseau national chilien. Il s'agit d'une situation typique pour les lieux isolés et peu peuplés, tels que les endroits difficilement accessibles, les îles de petites tailles, etc. Le lecteur pourra se faire une idée du lieu avec les photographies[2] des illustrations 1.1, 1.2 et 1.3.

Figure 1.1 – Le village de Huatacondo, déconnecté du reste du reseau

Figure 1.2 – La batterie du projet de Huatacondo

Ensuite, le paradigme de micro-réseau comprend aussi un mode *connecté*. Dans ce cas, un prix de marché, par exemple, permet de décentraliser les prises de décision. En milieu urbain, un ensemble d'immeubles, une usine ou une flotte de véhicules hybrides peuvent se

[1] voir `https://building-microgrid.lbl.gov/huatacondo`
[2] Merci à Fernando Lanas de me les avoir fournies

Figure 1.3 – Les panneaux solaires du projet de Huatacondo

décrire comme un micro-réseau. Ainsi les idées présentées dans les deux premiers chapitres peuvent être adaptées pour le mode connecté.

La première partie se termine par un algorithme permettant de prendre en compte le vieillissement de long terme (Chapitre 4) de systèmes commandés. J'applique ensuite ce résultat à un exemple simplifié de micro-réseau. Ce travail est motivé par une question pratique à l'énoncé très simple : comment prendre en compte l'usure de la batterie dans le problème d'optimisation ?

Les idées et résultats de cette première partie sont construits sur la théorie du contrôle optimal en temps continu (appelé aussi commande optimale), avec une petite incursion dans la théorie du contrôle optimal stochastique dans le chapitre 3. Nous détaillons un peu plus bas notre contribution scientifique sur le sujet des micro-réseaux.

La deuxième partie de ma thèse porte sur un thème plus macroscopique : le marché de l'électricité. J'étudie un modèle d'enchères d'approvisionnement où un opérateur central organise la production pour minimiser le prix que le consommateur devra payer.

Observons que la réalité est très complexe. Les règles d'allocations et de rétributions, les contraintes techniques et réglementaires et le type d'agents qui y prennent part dépendent du marché considéré. En plus de l'aspect géographique, il existe plusieurs types de marchés (puissance, capacité, etc.) et ces marchés sont interconnectés que ce soit par des connexions physiques ou des produits financiers.

L'étude que nous proposons porte sur un modèle statique en réseau avec une demande inélastique. La demande est considérée comme un objet déterministe pour simplifier, le cas stochastique se traitant de façon similaire. Nous proposons un mécanisme d'incitation permettant:

- D'inciter les producteurs à révéler leurs vrais coûts de production,
- De minimiser le coût payé par le consommateur.

Nous étudions également la structure de ce marché à la lumière de quelques éléments de théorie des jeux. Ce travail emprunte beaucoup à la théorie des enchères et à la théorie des jeux Bayesiens. Le manuscrit contient plusieurs références à des résultats classiques d'optimisation convexe.

La construction du solveur numérique BocopHJB ne fait pas l'objet d'une discussion spécifique au sein de ce manuscrit. J'aborde quelques points clefs de la théorie sous-jacente (le Principe de Programmation Dynamique et l'équation de Hamilton-Jacobi-Bellman) dans la suite de cette introduction. Le guide utilisateur ajouté en annexe fournit l'information nécessaire à son utilisation, et le guide d'exemples illustre les types de problèmes que l'on peut résoudre avec le logiciel. Quelques extraits du code de micro-réseau simplifié présenté au Chapitre 4 sont également donnés en annexe.

Comme ce travail est une compilation de projets d'articles, les chapitres peuvent se lire indépendamment les uns des autres. Les noms de mes co-auteurs sont précisés au début de chaque chapitre. Les deux parties correspondent à des projets différents et indépendants.

1.2 Quelques mots sur l'électricité

L'énergie (et donc la production d'électricité) est un sujet sociétal important. Pour preuve, la période pendant laquelle ce manuscrit a été élaboré fut le témoin de la COP21. Durant cette conférence, les dirigeants du monde entier ont débattu d'une politique pour lutter contre le réchauffement climatique. Les problématiques liées à l'énergie et la production d'électricité furent un des grands axes discutés au cours de ces échanges.

Par ailleurs, l'électricité est aussi un sujet complexe. Le réseau électrique est un système gigantesque faisant interagir de nombreux agents à des échelles différentes dans un environnement incertain et sous des horizons temporels variés, auxquels s'ajoute un lot de contraintes techniques et légales.

Enfin, l'industrie de l'électricité est en pleine mutation. Au cours de ma deuxième année de thèse, le marché mondial du panneau solaire a augmenté de 25%. Les énergies renouvelables sont en plein essor (cf. graphique 1.4 et 1.5). L'objectif de la France est un mix énergétique comprenant 20% de renouvelables d'ici 2020. L'ADEME (Agence de l'Environnement et de la Maîtrise de l'Energie) a même envisagé (plus comme expérience de pensée que comme plan politique) des scénarios avec des mix à plus de 80%. Dans ce contexte, les quelques phrases données en épigraphe, attribuées à Thomas Edison (à qui l'on attribue l'invention de la lampe à incandescence) prennent des accents de prophétie.

Dans certains pays, la part de renouvelables dépasse les 50% (Norvège 98%, Brésil 77%, Canada 63%, Costa Rica 88%). De grands projets sont en cours dans d'autres pays (au Chili par exemple). La catastrophe de Fukushima a durablement changé le rapport des sociétés au nucléaire.

Des changements importants s'annoncent pour les années à venir:

- augmentation des sources intermittentes dans le mix énergétique,

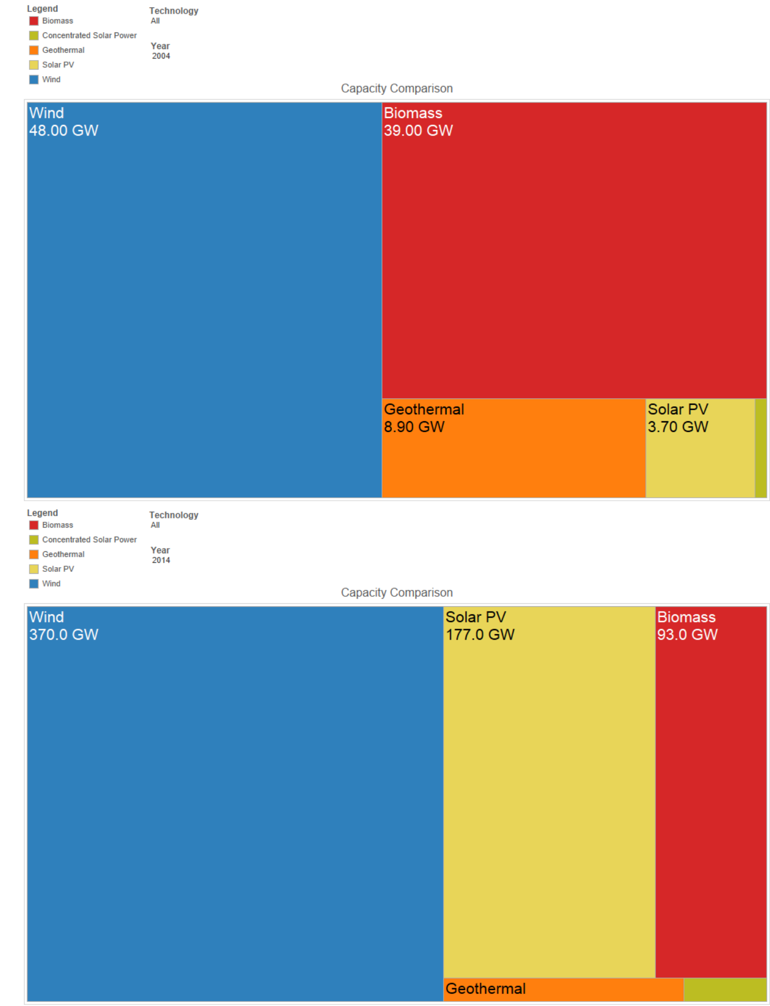


Figure 1.4 – Capacité en renouvelables (monde) en 2004 (en haut) et 2014 (en bas) Source: `http://www.ren21.net/resources/charts-graphs/`

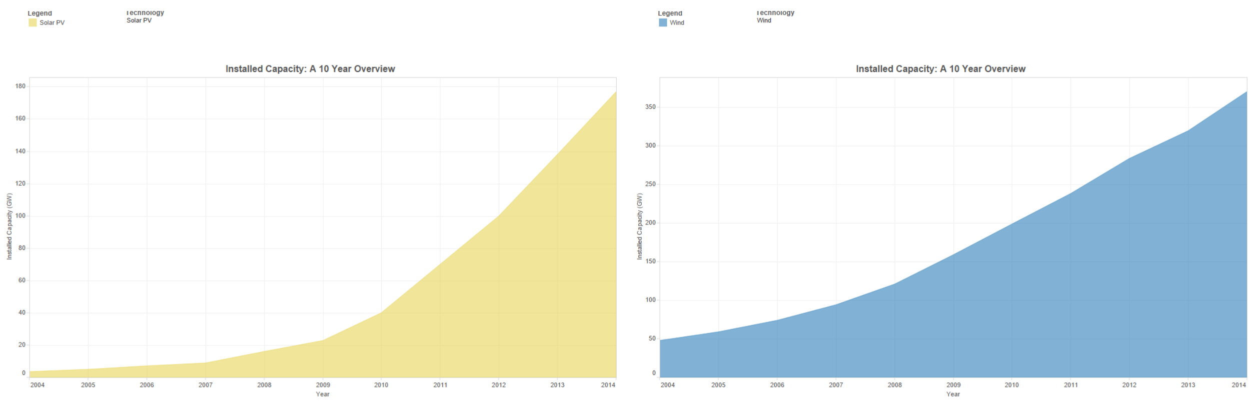


Figure 1.5 – Capacité en solaire (gauche) et éolien (droite) entre 2004 et 2014 (monde). Source: `http://www.ren21.net/resources/charts-graphs/`

- avènement du véhicule électrique
- réseaux intelligents,
- production par le consommateur.

Ces changements, qui sont portés par le progrès technologique (loi de Moore sur les panneaux solaires, batteries de plus en plus performantes) et l'impératif écologique, introduisent un nouveau degré de complexité. Par exemple, il est bien plus difficile de définir un coût marginal pour des énergies intermittentes que pour une centrale à charbon. Dès lors, la définition économique d'un prix s'en trouve bouleversée. L'augmentation du parc de véhicules électrique se traduira vraisemblablement par une augmentation des capacités de stockage d'énergie disponibles, les batteries des véhicules pouvant être utilisées pour stocker le surplus de production des heures creuses. Ce nouveau moyen de stockage potentiel que représente la flotte électrique demandera beaucoup de travail d'intégration.

1.3 Résumé et contributions de la première partie

La première partie de ma thèse porte sur l'optimisation de la production d'un micro-réseau. Le modèle de micro-réseau a été construit au cours d'une collaboration avec des ingénieurs du centro de Energia de l'Universidad de Chile. Ceux-ci disposent d'un micro-réseau (évoqué plus haut) isolé, à Huatacondo dans le désert de l'Atacama. Ce village produit localement l'électricité qu'il consomme. Pour subvenir à ses propres besoins en électricité, le village dispose d'un panneau solaire, d'un générateur diesel et d'une batterie.

La figure 1.6 est une représentation symbolique d'un tel micro-réseau. Le panneau solaire produit de l'énergie quand il y a du soleil. Si à un moment donné la production d'énergie intermittente est supérieure à la demande, on peut stocker cette énergie dans la batterie pour l'utiliser plus tard. Si au contraire, cette production est insuffisante pour satisfaire la demande instantanée, on peut puiser dans les réserves énergétiques de la batterie ou utiliser le générateur diesel. Lorsque le générateur est utilisé, la consommation de diesel s'accompagne d'un coût supplémentaire. Le rôle d'un EMS (Energy Management System) est de satisfaire la demande en électricité à moindre coût tout en satisfaisant des contraintes de fonctionnement. De fait, un EMS a en général pour composant principal un algorithme d'optimisation.

A priori, il y a donc deux variables d'ajustement possible pour le problème, mais si on considère l'équation d'équilibre des puissances :

$$P_{diesel} + P_{batterie} + P_{solaire} = P_{demande}, \tag{1.1}$$

on se rend compte que si on fixe la puissance diesel, la puissance de la batterie est fixée aussi. On en déduit qu'une formulation avec un contrôle (le diesel) et un état contrôlable

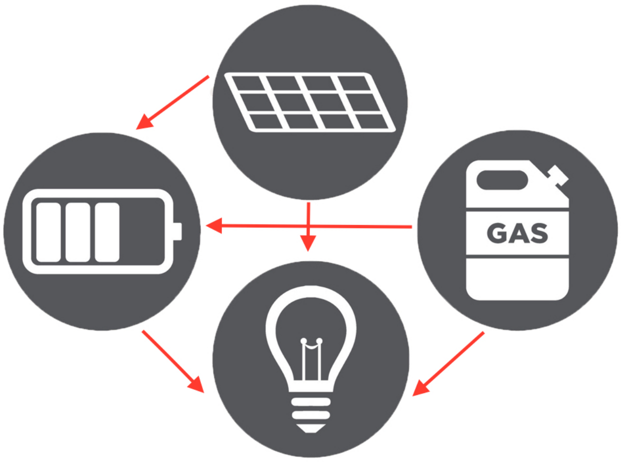


Figure 1.6 – Représentation synthétique de la microgrid de Huatacondo. En haut la demande en électricité, en bas la production non contrôlable, et au milieu les deux *variables d'ajustement* : la batterie et le générateur diesel. Puisque par bilan des puissances, la demande doit être égale à la somme des productions, et que seuls les deux éléments de la ligne du milieu (diesel et batterie) correspondent à des quantités contrôlables, on peut comprendre avec cette figure que la seule variable de décision sera la puissance du générateur diesel.

(correspondant à l'énergie stockée dans la batterie) doit être possible.

Plusieurs aspects sont à prendre en compte lors de l'optimisation: les contraintes physiques (typiquement de charge et de décharge de la batterie) et opérationnelles (par exemple éviter de se retrouver avec une batterie complètement vide). Dans la littérature on trouve surtout des formulations en temps discret du problème d'optimisation du micro-réseau. Je propose dans cette thèse une représentation continue du problème, ce qui permet d'utiliser d'autres outils d'analyse (principe du maximum de Pontryagin, régularité[3], travail sur le problème initial non linéaire), et de résolution (équations aux dérivées partielles, Monte-Carlo, etc.). Le chapitre 2 introduit une formulation déterministe du problème de gestion de micro-réseau. Comme l'aléa n'est pas pris en compte directement dans cette formulation, cette approche peut s'utiliser en horizon glissant.

Le chapitre 3 étend cette formulation à un cadre stochastique. Comme la production solaire est très prévisible dans un désert, l'aléa porte sur la demande, qui est modélisée avec un terme Brownien et un terme de retour à la moyenne. Une telle approche demande d'estimer un nombre relativement faible de paramètres. Nous comparons la stratégie obtenue avec une stratégie à horizon glissant sur un profil historique de demande.

Le chapitre 4 propose un algorithme pour la résolution d'un problème de contrôle optimal

[3]en particulier, cf. l'hypothèse 3 dans le chapitre sur le vieillissement.

avec vieillissement et données périodiques. L'intérêt d'un tel algorithme dans le cadre d'un micro-réseau est clair: le prix d'achat de la batterie représente une part importante du budget du micro-réseau et, de plus, sa durée de vie dépend de l'usage que l'on en fait. Au fur et à mesure qu'une batterie vieillit, son efficacité décroit. Il faudrait donc idéalement prendre en compte le vieillissement de cette batterie dans le problème d'optimisation du micro-réseau.

Mais ce vieillissement se fait sur plusieurs années, et n'a donc pas d'effet mesurable à l'échelle d'une journée. Quel niveau de vieillissement viser dans la résolution du problème de l'EMS sur un horizon de 24 heures ? Comment éviter que la prise en compte de ce vieillissement rende le problème trop difficile à résoudre numériquement dans les délais impartis ? Pour le prendre en compte dans le problème d'optimisation, une technique consiste à pénaliser l'usage de la batterie avec un terme de pénalisation dans le critère à optimiser. Si le coefficient de pénalisation est nul, cela revient à ne pas prendre en compte le vieillissement dans le critère d'optimisation. À l'inverse, si ce coefficient est très grand, cela revient à s'interdire d'utiliser la batterie, et de n'utiliser que le générateur diesel comme variable d'ajustement pour l'équilibre des puissances. En général ce coefficient de pénalisation est choisi de façon plus ou moins arbitraire. Nous proposons une règle pour décider de sa valeur. Nous modélisons l'état de santé de la batterie avec une variable d'*âge*. Or, si l'on considère le problème d'optimisation du micro-réseau sur plusieurs années, il faut prendre en compte l'état de santé de la batterie comme une variable d'état, au même titre que la quantité d'énergie stockée dans la batterie. Sur un intervalle temporel grand, il existe donc un profil optimal d'âge.

L'algorithme de poids adaptatifs permet de choisir le coefficient de pénalisation (le poids) de telle sorte que le contrôle optimal résultant pour un horizon d'une journée provoque un niveau de vieillissement qui soit proche de l'optimum. Les résultats sont d'abord présentés de façon générale, en dehors du contexte du micro-réseau. J'expérimente ensuite l'algorithme sur un modèle analytique de micro-réseau.

Nous venons de voir les principaux axes abordés dans cette première partie. Les principales contributions sont les suivantes

Chapitre 2: Continuous Optimal Control Approaches to Microgrid Energy Management

- formulation du problème de l'EMS comme d'un problème de contrôle optimal non linéaire en temps continu,
- modélisation des états allumé et éteint du générateur diesel,
- introduction d'une simplification algorithmique pour le cas des coûts de production concaves,

- comparaison de deux méthodes de résolution du problème en temps continu avec une approche mixte linéaire en nombres entiers (MILP).

Chapter 3: A Stochastic Continuous Time Model for Microgrid Energy Management Ce chapitre est paru comme article de conférence pour the European Control Conference 2016 (ECC2016), Aalborg Danemark 29 juin-1er juillet 2016.

- extension au cas stochastique du modèle déterministe. Le modèle proposé pour la demande nécessite l'estimation d'un nombre limité de paramètres.
- comparaison avec une approche déterministe à horizon glissant sur un profil historique.

Chapter 4: Long Term Aging : An Adaptative Weights Dynamic Programming Algorithm

- introduction de l'algorithme de poids adaptatifs,
- estimation de l'erreur,
- discussion sur la complexité algorithmique,
- expérience numérique sur le vieillissement d'un micro-réseau.

1.4 Résumé et contributions de la seconde partie

Dans la second partie, nous proposons un mécanisme de régulation permettant de diminuer le pouvoir de marché des producteurs d'électricité, au profit du consommateur. On considère un modèle de marché en réseau. Le marché est modélisé par un graphe. En chaque point du graphe:

- il y a un producteur d'électricité,
- la demande est connue.

Un opérateur central alloue la production de façon à minimiser ses coûts. L'électricité peut transiter à travers le réseau, mais cela engendre des pertes en ligne (effet Joule). Ces pertes sont à la charge du consommateur.

Les producteurs voulant maximiser leur profit, ils sont tentés de surévaluer leurs coûts de production pour augmenter leurs marges. Escobar et Jofré ont montré dans [35] que les pertes en ligne donnent aux producteurs un pouvoir de marché, et que la surévaluation est dans ce cadre là une attitude raisonnable des producteurs.

Ainsi dans le cas d'un réseau de deux nœuds avec des producteurs symétriques de coût marginal de production c, on peut montrer que ceux-ci annonceront à la place un coût de

$$c^* = \frac{c}{1-2rd} > c, \tag{1.2}$$

où r est le coefficient de perte en ligne, et d la demande aux deux nœuds du réseau.

Après un chapitre introductif (chapitre 5), nous présentons un mécanisme d'incitation pour des coûts de productions linéaires (chapitre 6). Nous généralisons ce mécanisme dans le chapitre 7 au cas de coûts de production linéaires par morceaux dans un réseau avec des contraintes et des externalités plus générales. Nous discutons aussi des propriétés du programme d'allocation que résout l'opérateur central et en déduisons un algorithme de calcul d'allocation "décentralisé". Le dernier chapitre contient un résultat d'unicité de l'équilibre de Nash en stratégie pure du jeu Bayésien induit par ce modèle de marché, dans le cas de fonctions de coût linéaires. Nous introduisons ensuite un système d'équations différentielles couplées pour lequel cet équilibre de Nash est un point stationnaire. Nous en déduisons un schéma numérique pour calculer les stratégies d'équilibre. Pour la famille de jeux étudiée, ce schéma présente de meilleures propriétés de convergence numérique que l'algorithme d'itérations sur les meilleures réponses. Il nous a permis de calculer les stratégies d'équilibre du marché à partir desquelles on obtient le coût moyen payé par la société quand le mécanisme d'incitation n'est pas implémenté. De cette manière on peut estimer l'économie que représente pour le consommateur l'implémentation du mécanisme d'incitation (cf. chapitre 6).

Pour cette deuxième partie, les contributions apportées sont:

Chapter 5: Mechanism Design and Auctions for Electricity Network

- ce chapitre est principalement une introduction du contexte général pour ce qui va suivre,
- on propose en illustration un calcul de la solution du problème d'allocation de l'opérateur dans le cas de deux nœuds et des coûts de production linéaire par morceaux (deux morceaux).

Chapter 6: Cost-Minimizing regulations for a wholesale electricity market

- nous introduisons le mécanisme d'incitation dans le cas de coûts linéaires,
- nous calculons numériquement sur un exemple ce que gagne le consommateur si ce mécanisme d'incitation est implémenté,
- nous montrons que, contrairement à ce que pourraient nous faire croire les résultats numériques, ce que paie consommateur n'est pas une fonction affine du coefficient de perte en ligne r.

Chapter 7: Mechanism design and allocation algorithms for network markets with piece-wise linear costs and quadratic externalities

- nous généralisons le résultat du chapitre précédent pour des fonctions linéaires par morceaux, des externalités et un réseau général,
- nous observons quelques propriétés de la solution du problème d'allocation de l'opérateur central,
- nous en déduisons en particulier que la quantité moyenne produite par un producteur est une fonction régulière de son coût de production,
- nous en déduisons aussi un algorithme de point fixe pour calculer la solution du problème d'allocation du principal,
- nous comparons les performances de cet algorithme avec les performances d'une résolution avec le solveur CVX,
- pour le cas binodal, nous introduisons un autre algorithme plus rapide.

Chapter 8: On a class of bidding games whose dynamics converges to the unique pure equilibrium

- nous identifions une classe de jeux Bayésien pour laquelle nous montrons l'existence d'un équilibre de Nash en stratégie pure,
- nous donnons des conditions suffisantes d'unicité,
- nous proposons un algorithme pour calculer les stratégies d'équilibre,
- nous appliquons ces résultats au cas du marché binodal avec coûts de production linéaires.

1.5 BocopHJB

BocopHJB est un solveur numérique codé en C++ que nous avons développé pendant la première année de thèse. BocopHJB permet la formulation et la résolution de problèmes de contrôle optimal en temps continu.

Il repose sur le principe de programmation dynamique énoncé par Richard Bellman dans les années 1950. On considère formellement le problème de contrôle optimal suivant:

$$V(x_0, t_0) = \inf_u \int_t^T \ell(u(t), x(t))\mathrm{d}t + \phi(x_T), \tag{1.3}$$

où x est un état contrôlé par la commande u, ℓ est un coût intégral et ϕ un coût final. La quantité V correspond au coût total minimum que l'on puisse obtenir en partant de l'état

x_0 au temps t_0. On dit que V est la fonction valeur du problème. C'est une fonction de x_0 et de t_0. Le principe de programmation dynamique de Bellman exprime la chose suivante:

Supposons qu'un contrôle u est optimal entre le temps t_0 et le temps T et nous amène à x au temps t. Alors si on part de x, ce contrôle est optimal entre le temps t et le temps T.
Le lecteur intéressé par ce principe pourra notamment lire le livre de Isaacs sur les jeux différentiels [54].

BocopHJB est, comme Bocop, un logiciel de contrôle optimal développé au sein de l'équipe Commands. Celui-ci ne repose pas sur de la programmation dynamique, mais sur ce que l'on appelle la méthode directe, qui consiste à discrétiser *directement* le problème et résoudre ensuite le problème d'optimisation discrétisé. Le guide utilisateur et quelques exemples sont donnés en annexe de la thèse. BocopHJB se présente comme un code source en C++ (non compilé). Pour résoudre un problème de contrôle avec BocopHJB, il suffit d'indiquer les différentes données du problème (dynamique, objectif, contraintes) dans des fonctions C++ correspondantes. Les paramètres de discrétisation et de simulation sont à indiquer dans des fichiers textes. Une fois le problème écrit, l'utilisateur doit compiler le code, puis le lancer.

BocopHJB résout alors le problème par programmation dynamique. Plus techniquement, BocopHJB calcule la solution de l'équation de Hamilton-Jacobi-Bellman associée au problème de contrôle optimal en utilisant un schéma semi-Lagrangien. L'équation de Hamilton-Jacobi-Bellman (HJB, d'où BocopHJB tient son nom) est une équation aux dérivées partielles non linéaire qui s'obtient formellement en combinant un développement de Taylor de la fonction valeur (ou d'Itô dans un cadre stochastique avec un mouvement Brownien) avec le principe de programmation dynamique:

$$\partial_t V(x,t) + \inf_u \ell(x,u) + \partial_x V(x,t) b(x,u) = 0, \tag{1.4}$$

où b est la dynamique du système commandé. Pour des références sur l'équation HJB, le lecteur pourra consulter [20], [17], [88] et [76].

Il est important de préciser ici que cette dérivation est formelle, et qu'en général, un travail théorique est nécessaire pour donner un sens à cette équation. En effet, rien n'assure a priori que la fonction valeur V soit dérivable. Dans ce cas, le sens de (1.4) n'est pas clair dans un cadre classique. La bonne notion de solution est celle de solution de viscosité. Le lecteur pourra se référer au User Guide [30] pour une présentation détaillée de la théorie. L'idée est la suivante : en chaque point, remplacer la fonction valeur par des fonctions test régulières qui l'approchent par en dessus ou par en dessous. Ensuite au lieu de vérifier directement que V satisfait l'identité (1.4), on utilise les fonctions test (et une relation très proche de (1.4))

Le schéma numérique par lequel nous résolvons l'équation HJB est un schéma semi-Lagrangien, qui revient à discrétiser directement le principe de programmation dynamique. Le lecteur pourra consulter [31] pour plus de détail sur ce schéma numérique.

La résolution de l'équation de Hamilton-Jacobi-Bellman par schéma semi-Lagrangien consiste donc à calculer, en partant du temps final, les fonctions valeurs. Les calculs peuvent prendre du temps si l'état est de dimension "grande" (typiquement supérieur à 3). C'est ce que l'on appelle la malédiction de la dimension: le nombre d'opérations à effectuer lors de la résolution numérique explose avec la dimension de l'état. C'est l'une des principales limites de l'approche par programmation dynamique. Une fois que le calcul des fonctions valeurs pour chaque pas de temps est terminé, il reste à calculer effectivement le contrôle optimal. Pour cela on exploite à nouveau le principe de programmation dynamique, en remarquant que si V est connue, u s'obtient "très facilement". On part donc de l'état initial et on construit la trajectoire optimale à l'aide des fonctions valeurs calculées par schéma semi-Lagrangien.

1.6 Perspectives

Nous identifions ici quelques pistes d'approfondissement:

- pour le modèle d'EMS des chapitres 2 et 3, étendre la méthode à des processus avec sauts, étudier d'autres approches de résolution,
- étendre l'algorithme de poids adaptatifs au cas stochastique avec des contraintes en espérance,
- identifier d'autres applications de cet algorithme (en économie ?), qui ne nécessite en fait que des hypothèses de monotonie par rapport à une variable d'état monotone dans le temps,
- faire une étude plus poussée du gain lorsque le mécanisme d'incitation est implémenté (dans le cas général),
- identifier d'autres domaines d'application de ce mécanisme,
- le dernier chapitre s'intéresse à une partie restreinte du sujet qu'il aborde, mais une étude plus approfondie pourrait possiblement ouvrir le champ à un contexte plus général.

Part I

Microgrid optimization and dynamic programming

Chapter 2

Continuous Optimal Control Approaches to Microgrid Energy Management

This is a joint work with Frédéric Bonnans, Pierre Martinon, Francisco Silva, Fernando Lanas and Guillermo Jiménez-Estévez.

Contents

We propose a novel method for the microgrid energy management problem by introducing a nonlinear, continuous-time, rolling horizon formulation. The method is linearization-free and gives a global optimal solution with closed loop controls. It allows for the modelling of switches.

We formulate the energy management problem as a deterministic optimal control problem (OCP). We solve (OCP) with two classical approaches: the direct method and Bellman's Dynamic Programming Principle (DPP). In both cases we use the optimal control toolbox Bocop for the numerical simulations. For the DP approach we implement a semi-Lagrangian scheme adapted to handle the optimization of switching times for the on/off modes of the diesel generator. The DP approach allows for accurate modelling and is computationally cheap. It finds the global optimum in less than a second, a CPU time similar to the time needed with a Mixed Integer Linear Programming (MILP) approach used in previous works. We achieve this result by introducing a trick based on the Pontryagin Maximum Principle (PMP). The trick increases the computation speed by several orders and improves the precision of the solution.

For validation purposes, we performed simulations on datasets from an actual isolated microgrid located in northern Chile. The result shows that the DP method is very well suited for this type of problem.

2.1 Introduction

Distributed Energy Resources (DER) play a key role as an energy supply alternative. Moreover, DER are in most of the cases renewable energy sources and bring positive environmental impacts and contribute to sustainability. In order to integrate in a massive way DER into interconnected power systems or make use of DER as a power source for isolated locations, microgrids appear as a suitable technical solution.

A microgrid is a group of interconnected loads and DER that acts as a single controllable entity. It can operate connected to the main network or autonomously (isolated)[89]. In either case, an Energy Management System (EMS) is required to coordinate the different units that compose it. The EMS solves an optimization problem and, as described in [74], this problem falls into the category of mixed integer nonlinear programming (MILP).

Depending on the philosophy established for the EMS and the different components (generation units, loads, storage devices) incorporated into the microgrid, the objective function may be nonlinear. Moreover, the operation of some of these components involves start up / shut down set points that are typically represented as binary functions of time in the problem formulation. Constraints represent in particular operational limitations of storage devices and generation units (i.e., batteries (dis)charging patterns).

There has been different formulations to handle this problem, the most common being through MILP, for which the complexity mostly stems from the modelling of nonlinearities: battery charging/discharging pattern and the diesel engine efficiency [74] for instance. Heuristic techniques have been also applied to the microgrid EMS problem, such as Genetic Algorithms (GA) [28], [47], Particle Swarm Optimization (PSO) [28], and Ant Colony Optimization (ACO) [27].

Finally, recent works which focus on microgrid energy management systems have incorporated a more detailed modelling of the energy storage system. This energy management system considers the importance of the cost associated with its replacement, so that extending the life span of the battery is part of the objective. In this context, GA have been implemented to solve the problem [96], and other predictive control approaches such as the ones described in [39], [78], [51] and [58].

Other authors have made use of Dynamic Programming (DP) to solve the EMS problem. Kanchev et. al. [56] use DP but look for GHG emissions reductions, [64] focus its objective on the Energy Storage System (ESS) management. These cases do not consider the economic efficiency of the whole microgrid. In [95] the EMS problem is focused on buildings decision incorporating uncertainty modeled with Markov chains under a discrete approach. Babazadeh [9] makes use of DP to handle the wind power management in a microgrid environment. In [71] DP has been developed to solve out the maximum profit an owner might achieve from energy trading in a day, either in isolated or connected mode, but do not consider in an explicit way effects and management of batteries. Finally, [53] applies a multi path dynamic programming (MPDP) approach to solve a power scheduling considering load/generation changes and time of use (TOU) tariff for a low voltage DC microgrid incorporating energy storage battery, fuel cell and PV.

An important conceptual difference between previous works and the present study is that we start with the optimal control problem formulated in continuous time. This gives access to a broader set of theoretical and numerical tools (e.g. Pontryagin's Principle and the Direct Approach).

The microgrid model presented in this work handles some challenges involved with the microgrid EMS, such as units modelling, ESS management, CPU solving time for real applications and the switching of the generator mode (on or off) among others, with a continuous time optimal control formulation. This approach keeps the original non-linear model for the numerical optimization, which enhances the solutions accuracy. The proposal considers two solution methods:

The direct method starts with a time discretization to transform the continuous optimal control problem into a Nonlinear Programming (NLP) problem. The NLP is then solved with any usual technique (see for instance [16]).

The DP method relies on Bellman' s Principle and uses a discretization of both time and space to compute the value function. This information then allows the reconstruction of the optimal trajectory using feedback controls (See for example [12]).

We perform numerical simulations for both methods using the optimal toolboxes Bocop and BocopHJB [19, 18]. The proposed methods are validated with data from a real microgrid operating in Huatacondo, an isolated northern Chilean village that relies completely on the microgrid concept for its electricity supply, which is described in section II. The present study uses a similar model to the one presented in [75], so that the comparison is relevant. We show results for the three approaches: MILP, direct method and DP.

Note that this work focuses on the comparison of the three techniques, but does not intend to deal with their implementation as building block of upper level algorithms, such as Model Predictive Control (MPC). Likewise, the demand and load modelling is out of the scope of this article. In addition, in all this work the microgrid is considered in disconnected mode, but a similar approach in connected mode could be envisioned (using a market price for instance).

The main contributions of this work are:

- the introduction of a continuous time non-linear framework for the microgrid energy management problem,
- for the dynamic programming approach, the modelling of the generator switching,
- the combination of the Pontryagin Maximum Principle and the Dynamic Programing Principle to get a surprising improvement of the computing time
- a comparison of the continuous time non-linear framework (with two resolution techniques, DP and direct method) with a MILP formulation.

The paper is organized as follows. Section II describes the microgrid system and the optimal control formulation for its energy management. Section III explains the numerical methods we use to solve the optimal control problem. Section IV presents the numerical simulations with the direct and DP methods. Section V comments the results of the simulations. The conclusion sums up the main results and presents ongoing research in the continuation of this work.

2.2 Model Presentation

2.2.1 General Aspects

Description of the Microgrid

The following model is based on a real microgrid operating in Huatacondo, an isolated village in northern Chile that relies entirely on the microgrid concept for its electricity supply. The microgrid we are considering includes a photovoltaic power plant (PV), a diesel generator and a battery energy storage system (BESS). It uses a mix of fuel and renewable energy sources. The solar panel produces electricity without any additional cost, but the generation pattern cannot be controlled and depends on the daily weather. The BESS can store energy for later use, but has limited capacity and power. The diesel generator has a minimal and a maximal output levels, and has a fixed start-up cost. All these are local generation units, i.e. situated physically near the electric consumption point, and electric losses due to distribution are not considered.

The aim is to find the optimal planning that meets the power demand and minimizes the operational costs, which, in this case, mainly relates to the diesel consumption. We follow the problem description from [75].

Optimal Control Formulation

We consider a fixed horizon $T = 48$ hours. For $t \in [0, T]$, we denote by $P_S(t)$ the solar power from the photovoltaic panels, $P_D(t)$ the diesel generator power and $P_L(t)$ the electricity load. The state of charge $SOC(t)$ of the BESS evolves according to the dynamics

$$S\dot{O}C(t) = \frac{1}{Q_B}\left(P_I(t)\rho_I - P_O(t)/\rho_O\right), \tag{2.1}$$

where Q_B is the maximum capacity of the battery, $P_I, P_O > 0$ are the input and output power of the BESS, and ρ_I, ρ_o are the efficiency ratios for the charge and discharge processes, assumed constant. Observe that (2.1) writes equivalently

$$S\dot{O}C(t) = \frac{1}{\tilde{Q}_B}\left(P_I(t)\tilde{\rho} - P_O(t)\right), \tag{2.2}$$

where $\tilde{Q}_B = \rho_O Q_B$ and $\tilde{\rho} = \rho_I \rho_O$.

We introduce the slack variable P_{slack} that represents the excess power ($P_{slack} < 0$), which has to be shed, or the missing power in the microgrid ($P_{slack} > 0$), which turns into unmet demand. The addition of this variable ensures the mathematical feasibility of the problem. Positive P_{slack} will be penalized by $C_{US}P^+_{slack}$, where C_{US} is a positive constant (see Table 2.1).

The underlying power equilibrium equation is

$$P_D + P_O + P_S + P_{slack} - P_L - P_I = 0. \tag{2.3}$$

Taking into account the demand and the various power production devices, we obtain that P_O and P_I can be written as nonlinear functions of (t, P_D, P_{slack}):

$$\begin{aligned} P_O(t, P_D, P_{slack}) &= -\min(0, P_S(t) + P_D - P_L(t) + P_{slack}), \\ P_I(t, P_D, P_{slack}) &= \max(0, P_S(t) + P_D - P_L(t) + P_{slack}). \end{aligned} \tag{2.4}$$

We model the fuel consumption of the diesel generator by the following strictly concave function

$$\int_0^T K P_D(t)^{0.9} \mathrm{d}t, \tag{2.5}$$

with $K = 0.471$. The fuel consumption curve was extrapolated from the datasheet provided

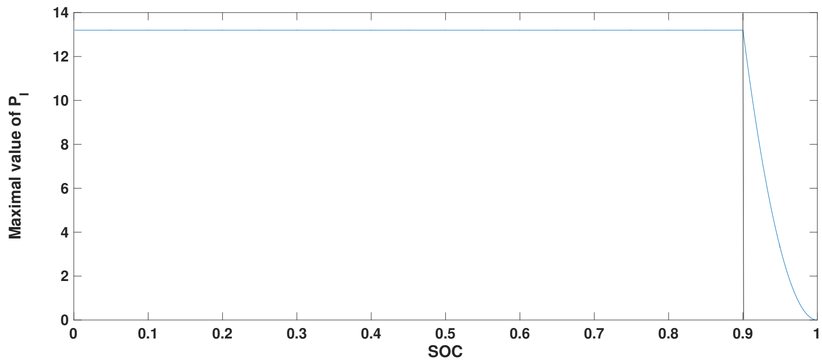


Figure 2.1 – Battery Charge Constraint

by the diesel generator manufacturer as in [75].

For physical reasons, the system is subject to the following constraints at every time $t \in [0, T]$:

$$SOC(t) \in [0.2, 1], \tag{2.6}$$

$$P_D(t) \in \{0\} \cup [5, 120], \tag{2.7}$$

$$\begin{cases} P_I(t, P_D(t), P_{slack}) \in [0, 13.2] & \text{if } SOC(t) < 0.9, \\ P_I(t, P_D(t), P_{slack}) \le 1320(1 - SOC(t))^2 & \text{otherwise}, \end{cases} \tag{2.8}$$

$$P_O(t) \in [0, 40]. \tag{2.9}$$

Note that (2.4) implies that (2.8)-(2.9) are constraints on P_D. The state constraint (2.6) expresses the maximum and minimum charge of the battery. Constraints (2.7) to (2.9) are control constraints. The minimal and maximal power for the diesel generator are given by (2.7). The charging and discharging limits for the battery are stated in (2.8) and (2.9). The charging limit depends on the state of charge, and is therefore a mixed control-state constraint, as illustrated on Fig. 2.1.

Since the operations time frame is larger than the optimization horizon, we impose a constraint on the final time to avoid the battery depletion. We impose this constraint either with a periodicity condition $SOC(0) = SOC(T)$ (direct method) or a penalization term $g(SOC(T))$ (DP method).

In summary the optimal control problem can be written under the following abstract formulation (see [24, Chapter 2])

$$(OCP)\begin{cases} \min_u \int_0^T \ell(u(t))\mathrm{d}t + g(x(T)) \\ \qquad \dot{x}(t) = F(u(t),t) \\ \qquad x(0) = x_0 \\ \qquad u(t) \in U_{x(t)} \\ \qquad x(t) \in \mathcal{C}. \end{cases} \tag{2.10}$$

In the notation above, x, u, F, U_x and $\mathcal{C}$ correspond respectively to the state variable (SOC), the control variable (P_D and P_{slack}), the dynamics of the system (2.1) and (2.4), the control constraints on $u(t)$ (see (2.7), (2.8) and (2.9)), and the state constraints (2.6). The time horizon T is 48 hours. The objective function is the sum of a final cost g (introduced in Section 2.2.3 to impose a periodicity condition) and the integral of ℓ defined in (2.5) (to which we add the penalization associated with the slack variable). Those functions take value from the state space and the control space respectively to $\mathbb{R}$.

2.2.2 Switching Cost

Turning the diesel generator on consumes fuel. We model this by considering that the diesel generator has two modes: when off, the only admissible control is $P_D = 0$, whereas when it is on, $P_D \in [P_{\min}, P_{\max}]$. At any time, one can *switch* from one mode to the other by paying the corresponding switching cost. This cost is zero to turn the generator off, and is equal to C_D to turn the generator on. It should be stressed that while the modelling of the switching cost is made straightforward by the Dynamic Programming approach, it is challenging for the Direct Method approach.

2.2.3 Periodicity Condition

To avoid the battery depletion at the end of the time horizon, we add a periodicity constraint on the state

$$SOC(0) = SOC(T). \tag{2.11}$$

The implementation of the constraint is straightforward for the MILP model and the Direct Method. The actual initial value is then optimized by the algorithm.

For the dynamic programming approach we model the periodicity condition by taking a similar approach to the "big M method" in linear programming:

$$\begin{aligned} g(SOC(T)) &= M \quad \text{if} SOC(T) < SOC_0, \\ g(SOC(T)) &= 0 \quad\ \ \text{if } SOC(T) \geq SOC_0. \end{aligned}$$

For the simulations, we set $SOC_0 = 0.7$.

2.3 Presentation of the numerical methods

We give here a brief presentation of the two resolution approaches we are considering and explain how to apply them in order to solve (2.10). The reader will find more on these approaches in [16] and [37].

2.3.1 The Direct Method Approach

Presentation

In the Direct Method we apply a time discretization to the dynamics equation. The optimal control problem is rewritten as a finite-dimensional optimization problem. The *decision* variables of this discretized problem are the values of the control variables at each time step. Since we solve the discretized problem by locally convergent algorithms, we cannot guarantee that the numerical solution (if any) is close to a global optimum. On the other hand, this approach often provides efficient solutions for large scale optimal control problems, with limited computing times.
Summary of the time discretization, using the Euler formula:

$t \in [0, T]$	$\rightarrow$	$\{t_0 = 0, \dots, t_N = T\}$
$x(\cdot), u(\cdot)$	$\rightarrow$	$Z = \{x_0, \dots, x_N, u_0, \dots, u_{N-1}\}$
Criterion	$\rightarrow$	$\min\ h \sum_{i=0}^{N-1} \ell(u_i) + G(x_N)$
Dynamics	$\rightarrow$	$x_{i+i} = x_i + hf(x_i, u_i) \quad i = 0, \dots, N$
Controls	$\rightarrow$	$u_i \in U_{x_i} \quad i = 0, \dots, N-1$
States	$\rightarrow$	$x_i \in \mathcal{C} \quad i = 0, \dots, N$

We therefore obtain a nonlinear programming problem on the discretized state and control variables

$$(NLP) \left\{ \begin{array}{l} \min_Z \ F(Z) \\ LB \leq C(Z) \leq UB. \end{array} \right.$$

The optimal control toolbox Bocop solves the discretized nonlinear optimization problem with the Ipopt solver [93] that implements a primal-dual interior point algorithm.

Modelling Remarks

We come back to our setting. This method allows a periodicity constraint of the form $SOC(0) = SOC(T)$ where the actual value is optimized by the algorithm. On the other hand, the constraint (2.7) is changed into $P_D \in [0, 120]$ because switching are hard to deal with within this framework. Another drawback is that switching costs are binary decisions which are not easily handled within this framework.

2.3.2 Dynamic Programming Approach

We propose a semi-Lagrangian scheme to solve the DP, in particular because it is adapted for problems with switching modes. We refer the reader to the monograph [37] and the references therein for an introduction to semi-Lagrangian schemes applied to optimal control problems. In addition, the Pontryagin Maximum Principle (PMP), see [77], provides additional information on the optimal solution. The combination of the Dynamic Programming Principle and the Pontryagin Maximum Principle reduces the computing time of the method significantly.

Brief Presentation of the Theory

Let $V(t, x_0)$ denote the value of problem (2.10) with initial time t and initial condition x_0. In R. Bellman's words [12] *"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."* In mathematical terms, V satisfies for $h \in (0, T-t)$:

$$V(t, x_0) = \inf \left\{ \int_t^{t+h} \ell(u_s) \mathrm{d}s + V(t+h, x(t+h)) \right\}, \tag{2.12}$$

the infimum being taken over the set of admissible controls. In our case, we will use an extended version of the DP approach that handles the switchings (See [37] for more details).

Semi-Lagrangian Scheme

The Semi-Lagrangian scheme consists of solving a discretized version of (2.12) over the space backward in time (see [37] for an overview). We have chosen this scheme to solve the problem because it has good stability properties, it allows large time steps and it is easy to implement. Let us motivate the scheme by first discretizing in time (2.12). Given a time step h and N such that $Nh = T$, let us set $t_k = kh$ $(k = 0, \dots, N)$. Denoting by V^k the "approximated" value function at t_k we have

$$V^k(x) = \min_{u \in U_x} \left\{ h\ell(u) + V^{k+1}(x + hF(u, t_k)) \right\}. \tag{2.13}$$

We derive the Semi-Lagrangian scheme from (2.13) by discretizing in space the state variable x and introducing interpolation operators in order to approximate $V^{k+1}(x+hF(u, t_k))$ in terms of its values in the space grid. The scheme is solved backward in time and, under standard conditions, it converges to the solution V of (2.12). We use the implementation of BocopHJB (see [19, 18]) for the numerical experiment.

The PMP Trick

The formulation has a property that greatly reduces the computing time. For the sake of simplicity we do not detail the aspects related to the state constraints. If $\bar{u}$ is the optimal control, denote by $\bar{x}$ the optimal state and by $\bar{p}$ the costate associated to the dynamics constraint $\dot{x}(t) = F(t, u(t))$. Defining the Hamiltonian $H(u, p, t) := pF(u, t) + \ell(u)$ the PMP says that for all $t \in [0, T]$ we have

$$H(\bar{u}(t), \bar{p}(t), t) \leq H(v, \bar{p}(t), t) \text{ for all } v \in U_{\bar{x}(t)}.$$

Since the dynamics is continuous and piecewise affine, the Hamiltonian is the sum of a continuous, piecewise affine and of a continuous strictly concave functions, and therefore is continuous, piecewise strictly concave. Therefore it can attain its minimum only at one of the extreme points of the pieces. Taking into account the constraints, we have at most five possible optimal controls, as illustrated in figure 2.2. Moreover, the values of these controls can be computed explicitly, since they do not depend on $\bar{p}$. Therefore, when doing the minimization in (2.13), we test only these controls instead of discretizing the control space, gaining both in speed and precision. So:

- if the Diesel is off (mode 0), we simply take $P_D = 0$.
- if the Diesel is on (mode 1), we test the five cases
  - $P_D = 5$ (minimum power),
  - $P_D = 120$ (maximum power),
  - P_D such that $S\dot{O}C = 0$ (battery unused),
  - P_D such that $P_i = P_i^{max}(SOC)$ (maximal charge),
  - P_D such that $P_0 = 40$ (maximal discharge).

The specific structure of the problem permits to reduce the computing time. More precisely, the candidates for the optimal control do not depend on the costate p and therefore can be evaluated and tested when computing the value function. In the general case, the control that minimizes the Hamiltonian is expressed from both the state and costate, the latter being unavailable in the DP approach (the costate actually corresponds to the gradient of the Value Function under suitable regularity assumptions).

Remark 1 (Slack variable)**.** *In the five cases above, we adjust the slack variable if needed to get an admissible Diesel output P_D.*

We now propose a pseudo-algorithm for the numerical resolution:

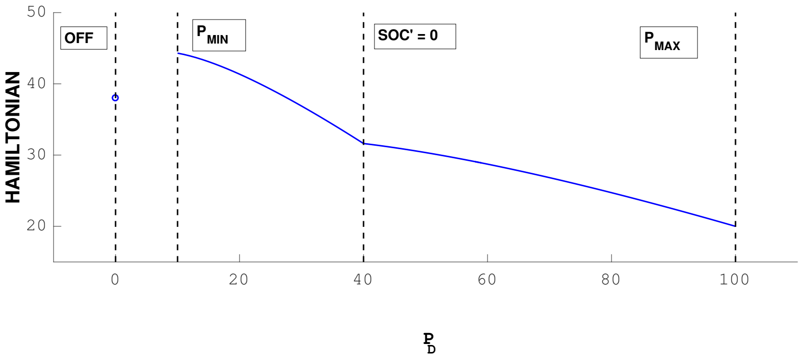


Figure 2.2 – The PMP trick illustrated

Data: h, I_{SOC}

Result: V^k, $kh = 0 \dots T$

for $kh \in T \dots 0$ **do**

  for $m \in \{ON,OFF\}$ **do**

    for $SOC \in I_{SOC}$ **do**

      $V^k_m(SOC) = \min\{ \min_{P_D \in G(SOC,m,kh)} h\ell(P_D) + V^{k+1}(x + hF(P_D, kh)),$
      $\min_{P_D \in G(SOC,\bar{m},kh)} h\ell(P_D) + V^{k+1}(x + hF(P_D, kh)) + C_{switch}(m)\}$

    end

  end

end

The parameter h corresponds to the time discretization size, and I_{SOC} is the state discretization grid. The result is the value function V^k for each time step k. The functions F and ℓ correspond to the dynamics and running cost as expressed in the abstract optimal control problem formulation (2.10) The mode $m \in \{\text{ON,OFF}\}$ corresponds to the fact that the diesel can be already working or turned off. We have denoted by $\bar{m}$ the negation of m. In case of switch, a cost $C_{switch}(m)$ has to be added to the cost to go function. This cost is the startup cost if the generator is turned on (see Table 2.1), and 0 else. The set $G(SOC, m, t)$ corresponds to the potential optimal controls we deduced from the *PMP trick*.

2.4 Numerical simulations

2.4.1 Comments on the Inputs: Solar Power and Power Load

We test the algorithms on two historical data sets. Both data sets correspond to representative 48-hour periods, one data set was obtained with winter data, the other one with summer data. Figures 2.3 and 2.4 show the load power and the solar power for the two

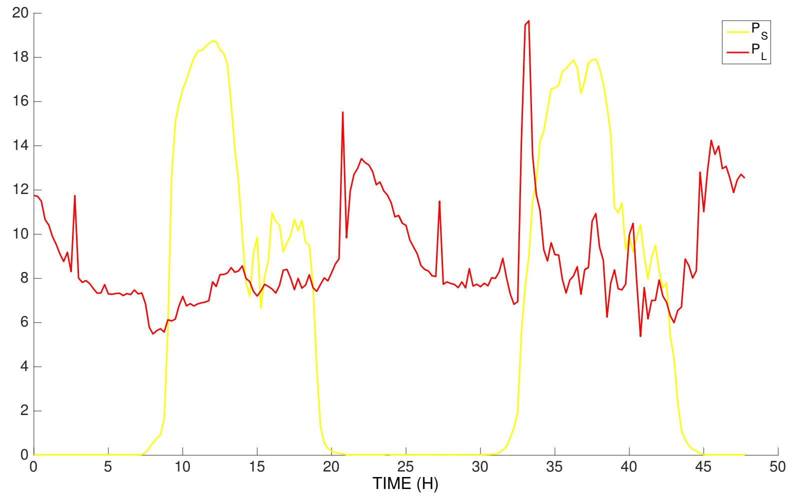


Figure 2.3 – Summer data in kW

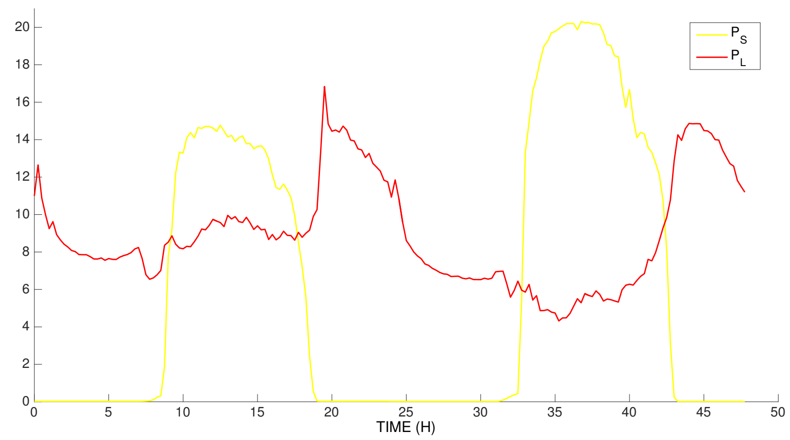


Figure 2.4 – Winter data in kW

days of each period.

Since the actual microgrid is situated in the Atacama desert, we assume the production from the photovoltaic panels to be reliably predictable. The demand, on the other hand, has a greater variability. While it is modeled as deterministic in this initial work, the extension of this model to a stochastic demand setting is the focus of another work.

2.4.2 Optimal Solutions for the Different Methods

In addition to the direct and DP methods, we present the results obtained with the MILP approach from [75] as baseline for comparison. The six solutions are illustrated in Figures 2.5 and 2.6 for the DP approach, in Figures 2.7 and 2.8 for the direct approach and in

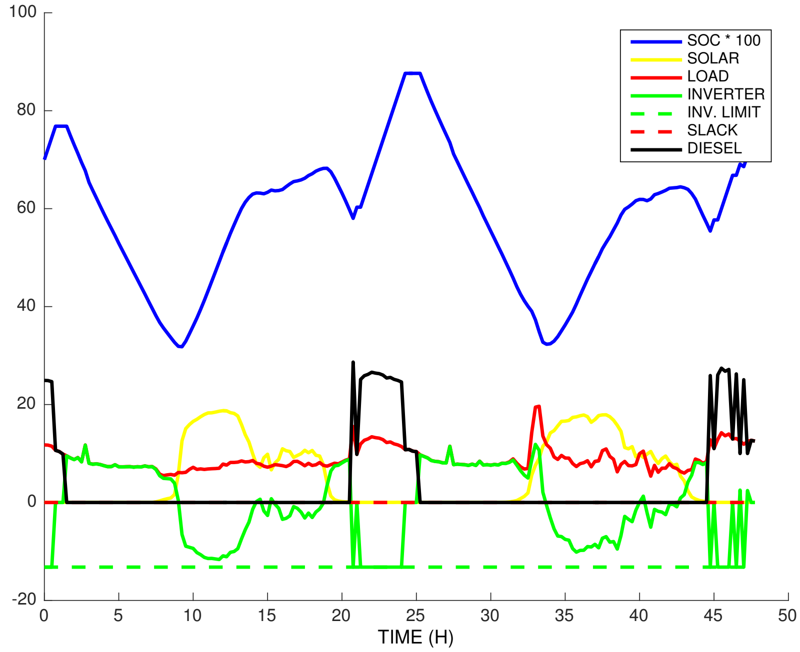


Figure 2.5 – Summer DP Simulation

Figures 2.9 and 2.10 for MILP method. The numerical results are summarized in Table 2.2.

Name	**Notation**	**Value**	**Unit**
Min Diesel Power	P_{min}	5	kW
Max Diesel Power	P_{max}	120	kW
Unserved Energy Cost	C_{US}	250	CLP/kWh
Diesel Start-Up Cost	-	1000	CLP
Diesel Price	C_D	500	CLP

Table 2.1 – Model parameters, CLP means Chilean Pesos

	MILP	DIRECT	DP
Diesel range	[18.66,29.69]	[2.63,14.23]	[8.15,28.66]
Switchings	2	3	2
Total Cost	34785	36244	34378
Cpu Time	3.92 s	4.41 s	0.88 s
SOC(0)=SOC(T)	0.7	0.641	0.7
SOC range	[0.20,0.89]	[0.38,0.75]	[0.30,0.83]
Slack Range	[0,0]	[0,0]	[0,0]

Table 2.2 – Results: MILP, direct and DP (summer case)

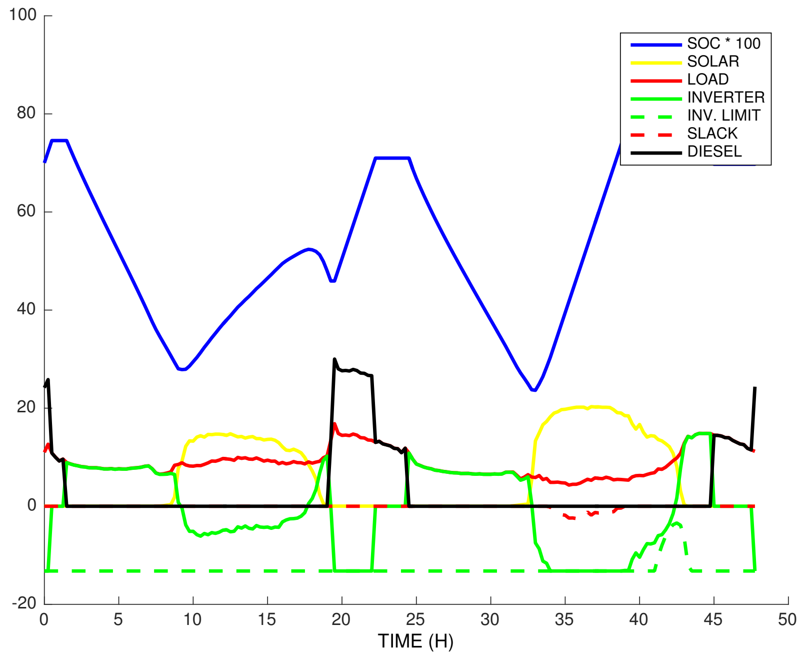


Figure 2.6 – Winter DP Simulation

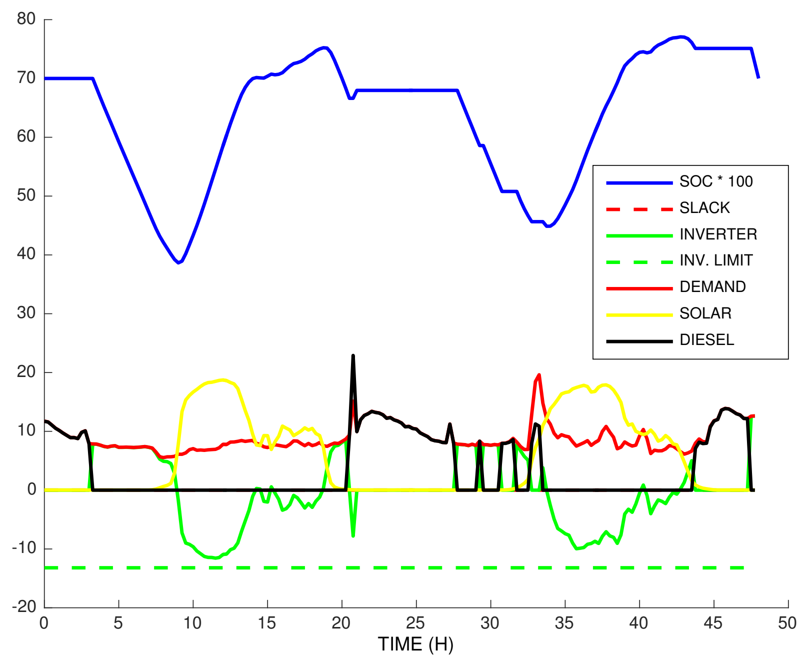


Figure 2.7 – Summer Direct Simulation

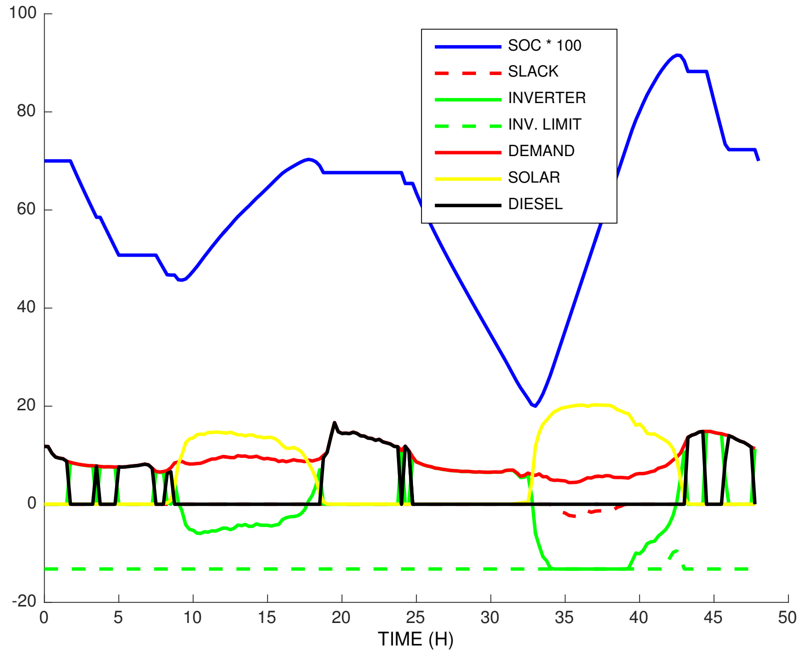


Figure 2.8 – Winter Direct Simulation

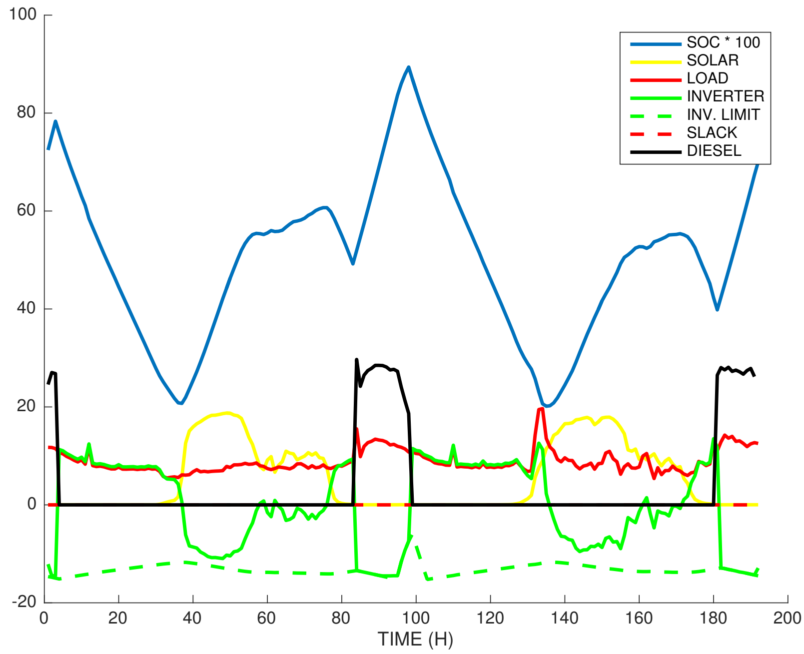


Figure 2.9 – Summer MILP Simulation

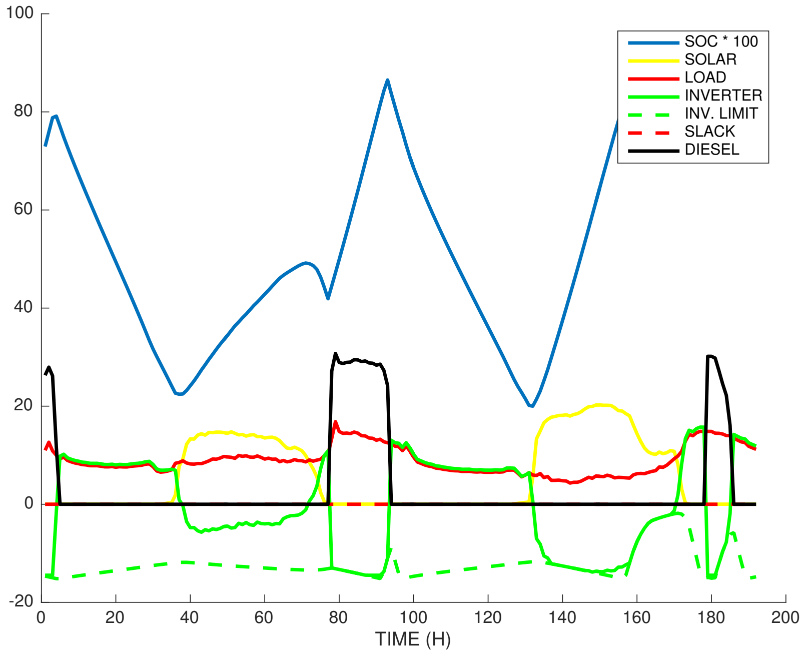


Figure 2.10 – Winter MILP Simulation

General Observations

- The Solar Power fills the demand, with any excess power used to charge the battery.
- The Diesel is always off when solar power is available, and is switched on once a day during the evening peak in demand. The Diesel output is often greater than power demand: it is also used to charge the battery.
- The battery fills the gaps between production and demand especially at night.
- MILP and DP solutions are quite close, while direct solution shows some clear differences (different initial/final SOC, no minimal power, spurious switchings).

Diesel Range

A qualitative difference between MILP and DP/direct is the existence of time intervals with a constant SOC, while the diesel exactly matches the power load. In the MILP solutions the diesel is either off or saturating the charge limit.

First we note that there is a tradeoff between low average production cost and low storage cost per energy unit. On the one hand, since the diesel cost function is concave, the diesel generator should run at maximal capacity to minimize the average unit price of power produced. On the other hand one incurs some losses when storing energy because the battery is not perfectly efficient. One has to set the diesel output equal to the net demand (so that nothing gets in or leaves the battery) to minimize those losses. This tradeoff explains why on the DP and Direct solutions we observe two kinds of non zero diesel levels (low: just sufficient to satisfy the demand, and thus keep a constant SOC, or high: maximum physical production level). The question is then why we do not observe the low level on the MILP simulation. We do not have a straightforward answer, as different phenomena could contribute to this observation. First the MILP problem is a linearized version of the DP problem, so it may happen that this linearized version does not produce the same output. Second the MILP numerical solution is only locally optimal and so may differ from the MILP actual solution.

2.4.3 Comparison of the methods

We highlight below the differences between the three optimization methods.

Global optimum. Both the MILP and direct approaches are local methods and may converge to a local solution, depending on the provided starting point and the choice of the stopping criterion (gap). On the other hand, the DP approach performs a global optimization over all possible (discretized) trajectories, and therefore always finds the global optimum. This is an advantage for the user since one does not have to find a "suitable" starting point. Also, the DP solution provides a feedback optimal control, whereas MILP and direct solutions are open-loop.

Switching cost. Both MILP and DP approaches take into account the switching costs for the diesel generator. They typically find solutions with one switch per day, located during the peak of power demand in the evening. On the other hand, the direct approach has free switchings, which explains why it may find solutions with many on-off oscillations.

Nonlinear model. The MILP method requires a piecewise linear reformulation of the nonlinear functions in the model, here for example the charging power limit or the cost of diesel consumption. Both direct and DP methods use the original nonlinear model. This simplifies the actual implementation, and may imply more accurate solutions.

Periodicity constraint and minimal diesel power. Compared to MILP and DP, the direct method optimizes the value of the initial/final SOC. On the other hand, it does not take into account the minimal power output for the diesel generator.

Computing time. For this problem the computing time is a few seconds for MILP and direct method, and less than one second for the DP approach. Note that DP is outperforming the two other approaches because of the PMP trick and the fact that the state is one dimensionnal. An interesting question is how well each method would scale for higher dimensions. MILP and direct approach are iterative methods, so changing the problem size may lead to a different convergence, making it difficult to assess the evolution of the CPU time. For the DP approach, on the other hand, the number of operations is always known and the CPU time can be predicted reliably. The CPU time should increase linearly in the number of time steps. Due to the state discretization, however, adding new state variables to the problem would have a significant impact on performance (the so-called curse of dimensionality). In terms of high performance computing, parallelization is possible with MILP and DP methods, not so easily with the direct method.

2.5 Conclusion and perspectives

We have applied two methods from the continuous optimal control field to the optimal energy management of a microgrid, namely the direct and DP approaches. Numerical simulations indicate that the DP method is very well suited to this problem as it is a linearization-free method that provides global optimal solution with closed loop control. It allows for the modelling of the switches and it is as fast as the MILP approach. We were able to obtain the global optimum in less than one second of CPU time, while taking into account the switching cost for the diesel generator. Solutions are close to the ones obtained in [75] with a MILP formulation, the main difference being the existence of time intervals where the battery stays at a constant SOC. In comparison with the two other approaches, the use of Pontryagin's Maximum Principle combined with Dynamic Programming reduces the computing time. The numerical experiments were performed with the optimal control toolbox Bocop.

From a theoretical standpoint, the continuous model offers a very large collection of mathematical results. The PMP trick introduced here is an example of insight one can get from a continuous time mathematical analysis. (Observe that a Maximum Principle exists for the discrete case, but only for convex Hamiltonian).

Ongoing works on this topic include the extension to a stochastic model for the power demand, and the study of the long-term aging of the battery.

Chapter 3

A Stochastic Continuous Time Model for Microgrid Energy Management

This is a joint work with J. Frédéric Bonnans, Francisco Silva, and Guillermo Jiménez Estévez It was accepted as a conference paper for the 2016 European Control Conference

Contents

We propose a novel stochastic control formulation for the microgrid energy management problem and extend previous works on continuous time rolling horizon strategy to uncertain demand. We modelize the demand dynamics with a stochastic differential equation. We decompose this dynamics into three terms: an average drift, a time-dependent mean-reversion term and a Brownian noise. We use BocopHJB for the numerical simulations.

This optimal control toolbox implements a semi-Lagrangian scheme and handles the optimization of switching times required for the discrete on/off modes of the diesel generator. The scheme allows for an accurate modelling and is computationally cheap as long as the state dimension is small. As described in previous works, we use a trick to reduce the search of the optimal control values to six points. This increases the computation speed by several orders. We compare this new formulation with the deterministic control approach introduced in [49] using data from an isolated microgrid located in northern Chile.

3.1 Introduction

A microgrid is a small network of loads and energy ressources controlled by an Energy Management System (EMS). It can be either connected to the main network or isolated. The coordination of the microgrid units requires the resolution of an optimization problem. This problem is described in the literature as the microgrid management problem. Palma-Behnke et al. introduce in [75] a microgrid EMS based on a rolling horizon strategy for which the microgrid management problem is formulated as a Mixed Integer Programming (MIP) problem. Heymann et al. show in [49] that this MIP formulation could be replaced by a continuous Optimal Control (OC) formulation. We extend this last approach to the stochastic case by introducing a stochastic dynamics for the load.

In [55] the microgrid energy management problem is formulated as a two-stage stochastic programming model based on optimization principle. Then, the optimization model is decomposed into a mixed integer quadratic programming problem by using discrete stochastic scenarios to approximate the continuous random variables. A scenarios generation approach based on a time-homogeneous Markov chain model is proposed to simulate time-series of renewable energy generation and load demand. Similar approaches are considered in [63] and [32], especially in [63] uncertainty is characterized by a scenarios generation approach based on autoregressive moving average (ARMA) model according to the probability density function of each random variable. In [73], uncertainty is addressed using a two-stage decision process combined with a receding horizon approach. The first stage decision variables (unit commitment) are determined using a stochastic mixed-integer linear programming formulation, whereas the second stage variables (optimal power flow) are refined using a nonlinear programming formulation. Other approaches appear, such as the one described in [94] where uncertainties related to renewable distributed generation are modeled by proper probability distribution functions and are managed by reserve provided by both DGs and loads.

The reader may refer to [76] to get an overview of stochastic control theory and the applications of the dynamic programming principle, and to [31] and [25] for more details about the discretization scheme we use. As in [49] we solve the microgrid management problem using Bellman's Dynamic Programming Principle (DPP). We compute an approximation of the value function using a time and space discretization, and then we use this value function to reconstruct an optimal control. The DPP approach presents numerous advantages. First, no starting point is required to initiate the optimization algorithm. Second, the algorithm computes a global optimum, as opposed to other methods relying for instance on first order optimality conditions that only compute local optima. Third, we can deal with integer variables such as the on/off status of a device (in our case the diesel generator). Fourth, we use directly the original non-linear model. This simplifies the implementation, and may also give more accurate solutions. Fifth, since we derive the optimal controls from the value function, those controls are in feedback form. Last but not least, if the state dimension is low (in our case, two), it is competitive from a computational perspective. In particular, the computational burden is linear in the number of time steps. We perform numerical simulations with BocopHJB, a C++ open source numerical solver for stochastic optimal control problems (see [18]). We point out that this solver does not solve stochastic problems with scenario trees but instead solves an associated deterministic second order partial differential equation. The data for our problem come from the Huatacondo microgrid. Huatacondo is an isolated village in the Atacama desert (northern Chile). The village relies completely on the microgrid for its energy supply.

This paper is organized as follows: Section 3.2 describes the microgrid and the demand model, and then formulates the stochastic optimal control problem. Section 3.3 presents the numerical method as well as the simulation methodology. We explain the parameters estimation in Section 3.4, and in Section 3.5 we display and comment the simulations results. Finally, the conclusion sums up the main results and presents ongoing research in the continuation of this work.

3.2 Model Presentation

3.2.1 System Description

On the supply side, the microgrid includes a photovoltaic power plant, a diesel generator and a Battery Energy Storage System (BESS). The photovoltaic power plant is a non dispatchable unit. Since one can accurately predict the climate in the desert region of the microgrid, we assume that we know the future production of renewable energy. This is why it is deterministic in this work. The diesel generator and the battery play the role of the dispatchable units. The marginal cost of the energy the generator produces is decreasing, i.e. the fuel consumption cost is a strictly concave function of the energy produced by the diesel generator. We derived this production function in [49] from the constructor data sheet. The diesel generator is either on or off. When on, the diesel generator cannot work below a given threshold (due to its physical properties). When switched on, the generator needs an additional amount of energy to warm up, which is modeled as a fixed switching cost. Since any ON switching is followed by an OFF switching and conversely, we account for half this switching cost for any switch, ON or OFF. We can store the energy surplus in the BESS when production is greater than demand and supply this energy to the system when demand is greater than production. This storage is not free as some energy is lost in the charge/discharge process. We will not take into account the battery aging in this work. On the demand side, the load comes from the villagers domestic needs. In our model proposal the randomness comes from the demand side. Since the village is small, the load is volatile. We modelize the load dynamics with a Stochastic Differential Equation (SDE). The grid is isolated, so there cannot be any flows from or to the outer world. We neglect the transmission losses because the village is small. Our objective is to find a strategy that minimizes the operating cost (diesel and switching cost) and produce enough electricity for the village.

3.2.2 Load Model

The microgrid historical load might suggest a random model with several jumps. Nonetheless such type of models requires a large number of parameters: the sizes and probabilities of jumps need to be fitted. As a first step toward the integration of stochastic modeling within our framework we propose a simpler model based on a Brownian motion. Our proposal is similar to an Ornstein-Uhlenbeck process (which is the continuous time equivalent of the AR(1) model) because it is a Brownian dynamics with a mean reversion. The difference is that the mean and volatility parameters are time dependent. We model the load process $L(t)$ with the Stochastic Differential Equation (SDE)

$$\mathrm{d}L(t) = (\dot{\Lambda}(t) + b(\Lambda(t) - L(t))\mathrm{d}t + \sigma(t, L)\mathrm{d}W(t), \tag{3.1}$$

where $\Lambda(t)$ is a deterministic load process (in kW), $b \geq 0$ is a unitless mean reversion coefficient, $\sigma(t, L)$ is the volatility (in $kW.t^{-0.5}$), $W(t)$ is a Wiener process and $L(0) = L_0$. The volatility σ has a bounded support in $[0, T] \times [0, L_{max}]$. Since Λ is bounded and $b \geq 0$, the load L remains bounded (and positive):

$$L(t) \leq \max(\sup_t \{\Lambda(t)\}, L_{max}). \tag{3.2}$$

This allows us to refer to section 3 of [25] for the mathematical properties of the system. Setting $Y(t) = L(t)e^{bt}$ and applying Itô's formula we get that $L(t)$ is equal to

$$\Lambda(t) + e^{-bt}(L_0 - \Lambda(0)) + \int_0^t e^{b(\tau - t)}\sigma(\tau, L(\tau))\mathrm{d}W(\tau).$$

So, $L(t)$ has expectation $\Lambda(t) + (L_0 - \Lambda(0))e^{-bt}$ and, by Itô isometry, a variance of at most $\sup(\sigma^2)(1 - e^{-2bt})/(2b)$. In Section 3.4 we will discuss the computation of σ and b and provide an empirical justification of the model.

3.2.3 Notations

We denote by t_0 the initial time and by T the time horizon. The state variables will be represented with capital letters. We denote by C the state of charge of the BESS and by L the load. We point out that only C is controlled, since the dynamics (3.1) of L does not depend on any decision. The diesel generator mode (on or off) at time t is represented with the variable $m(t) \in \{0, 1\}$ (0 for off and 1 for on). The control variables will be represented with lower-case letters. We write d the diesel generator output and s an artificial slack variable (to ensure the feasibility of the problem). The variable s represents the excess ($s < 0$) or missing ($s > 0$) power. We penalize decisions with a non zero slack variable by an integrable cost proportional to the absolute value of s. We impose s to be non positive if the diesel generator is off and bigger than a fixed constant if it is positive. We denote by $n(t)$ the counting variable equal to the number of switches that occurred between time t_0 and t. It is non decreasing over time and for all t, $n(t) \in \mathbb{N}$. We associate to each switch (OFF to ON and ON to OFF) a cost K equal to half the real cost needed to fire the diesel generator on. We write P_S the quantity of renewable energy produced at time t. As explained in §3.2.1, this is a deterministic function of time since we assume we have a reliable deterministic forecast. If we denote by P_I and P_O the quantities that go in and out of the BESS, then the power equilibrium equation writes

$$d + P_O + P_S + s - L - P_I = 0, \tag{3.3}$$

so that P_I and P_O can be written as non linear functions of s, L, P_S and d:

$$\begin{aligned} P_O &= -\min(0, P_S + d - L + s) \\ P_I &= \max(0, P_S + d - L + s). \end{aligned} \tag{3.4}$$

We denote by Q_B the maximum capacity of the battery, while ρ_I and ρ_O are the efficiency ratios for the charge and discharge processes. We write ℓ the cost function associated to the diesel consumption. The final cost function g ensures a minimal value of the final state of charge:

$$\begin{cases} g(C) = 0 & \text{if } C \geq C_0 \\ g(C) = M & \text{otherwise.} \end{cases} \tag{3.5}$$

where M is a large penalty parameter. Setting $C_0 = C(t_0)$ we ensure that the system finishes the day with as much energy in the BESS as what was stored at t_0.

3.2.4 Stochastic Control Formulation

We now define the value of the microgrid management problem as

$$V^{m_0}(t_0, C_0, L_0) :=$$

$$\inf_{n,d,s} \mathbb{E} \left(Kn(T) + g(C(T)) + \int_{t_0}^{T} \ell(d(t), s(t))\mathrm{d}t \right) \tag{3.6}$$

subject to, for all t:

$$\dot{C}(t) = F_C(L, d, s, t) \tag{3.7}$$

$$\mathrm{d}L(t) = (\dot{\Lambda}(t) + b(\Lambda(t) - L(t))\mathrm{d}t + \sigma(t, L(t))\mathrm{d}W(t) \tag{3.8}$$

$$m(t) = \frac{1 + (-1)^{n(t)}(2m_{t_0} - 1)}{2} \tag{3.9}$$

$$(C(t_0), L(t_0), m(t_0)) = (C_0, L_0, m_0) \tag{3.10}$$

$$\begin{cases} d(t) = 0 \text{ and } s(t) \leq 0 & \text{if } m(t) = 0, \\ d(t) \in I_d & \text{otherwise} \end{cases} \tag{3.11}$$

$$C(t) \in I_c \tag{3.12}$$

$$P_O \in I_{P_O} \tag{3.13}$$

$$\begin{cases} P_I \in I_{P_I} & \text{if } C(t) < 0.9, \\ P_I \leq A(C(t) - 1)^2 & \text{otherwise} \end{cases} \tag{3.14}$$

where $F_C(L, d, s, t) = \frac{1}{Q_B}(P_I\rho_I - P_O/\rho_O)$. We point out that by many ways this stochastic optimal control problem is similar to the deterministic model presented in [49]. Here the decision variables are the diesel output at any instant $d(t)$, the slack variable $s(t)$ and the value of the counting function $n(t)$. Note that optimizing over the counting functions is equivalent to optimizing over the switching times. Implicitly we impose those decisions to be non anticipative, i.e. progressively measurable with respect to the filtration generated by $W(t)$. Constraints (3.7) and (3.8) are respectively the power balance for the battery and the load dynamics. Relation (3.9) expresses the current mode as a function of the initial mode and the number of switches that occurred since t_0. If this number is even, $m(t) = m_0$ and if it is odd, $m(t) = 1-m_0$. Constraint (3.10) is the initial condition. Constraint (3.11) corresponds to the modeling of the diesel generator mode ($ON = 1$ or $OFF = 0$). If OFF, the diesel generator cannot produce anything and $d = 0$, else, the physics of the generator imposes d to be in $I_d = [d_1, d_2]$, with $d_1 > 0$. Last, constraints (3.12), (3.13) and (3.14) correspond to physical properties and limitations of the battery, with P_I and P_O defined by equation (3.4). The sets I_{P_O} and I_{P_I} are segments of $\mathbb{R}_+$ and I_C is included in $[0, 1]$. The parameters A and M are positive constants. Table 3.1 contains the numerical values we use.

3.2.5 Technical Remark

We already noticed that L is bounded over $[0, T]$. Thus the number of switches n is bounded on any scenario and the slack variable s is uniformly bounded over the scenari (this is of course true for d since $d \in \{0\} \cup I_d$). To our knowledge, there are no general well posedness results for stochastic control with state constraints. Nonetheless, since the controls are bounded and the diffusion is orthogonal to the outer normal of the state constraint we can argue that the viscosity approach developed in [57] for second order fully nonlinear elliptic equations with state constraints could be extended to our case with finite horizon and switching times.

3.3 Numerical Optimization Method

3.3.1 Dynamic Programming

The Dynamic Programming Principle (DPP) states that (see [76])

$$\begin{aligned} & V^{m_0}(t_0, C_0, L_0) = \\ & \inf_{d \in D_{m_0}, \tau \in \mathbb{T}_t, s} \mathbb{E} \int_{t_0}^{\tau} \ell(d, s) \mathrm{d}t + \min\{V^{m_0}(\tau, C(\tau), L(\tau)), \\ & \quad K + V^{1-m_0}(\tau, C(\tau), L(\tau))\} \end{aligned} \tag{3.15}$$

where the optimization is performed over (7)-(14) and $D_0 = \{0\}$, $D_1 = I_d$, and $\mathbb{T}_t$ is the set of stopping times in $[t_0, T]$. The time dependency of d and s is implicite in the integrand. Note that from (3.15) and applying Itô's formula we get that the value function formally satisfies the Hamilton-Jacobi-Bellman equation (see for instance [76])

$$\begin{aligned} \max \Big\{ -V_t^i - 0.5 V_{LL}^i \sigma^2 - V_L^i(\dot{\Lambda} + b(\Lambda - L)) + H_i \ , \\ V^i - (K + V^{1-i})) \Big\} = 0, \end{aligned} \tag{3.16}$$

where

$$H_0 = \sup_{s \leq 0} -\{\ell(0, s) + V_C^1 F_C(L, 0, s, t)\} \tag{3.17}$$

$$H_1 = \sup_{d \in I_d, s} -\{\ell(d, s) + V_C^0 F_C(L, d, s, t)\}. \tag{3.18}$$

We now explain a weaker version of a trick introduced in [49] for the deterministic case. We assume $s = 0$, i.e. there is a good balance between production capacity and load. If the diesel generator is off then by definition $d = 0$. Otherwise, the dynamics of the system is locally (3.16) for $i = 1$, so that heuristically, the control should maximize the Hamiltonian H_1 defined at (3.18). Since we maximize a piecewise convexconvexe function, the optimal controls can take a limited number of values that can be explicitly computed.

- if the diesel is off ($m = -1$), we simply take $d = 0$.
- if the diesel is on ($m = 1$), we test the five cases
  - d is set to the minimum power,
  - d is set to the maximum power,
  - d such that $F_C = \dot{C} = 0$ (battery unused),
  - d such that P_i is maximal (maximal charge),
  - d such that P_0 is maximal (maximal discharge).

From a computational perspective it is sufficient to test those values instead of discretizing the control space.

3.3.2 Algorithm

We solve the Hamilton-Jacobi-Bellman equation (3.15) with BocopHJB [18]. This open-source software solves second order finite horizon Hamilton-Jacobi-Bellman equations with a semi-Lagrangian scheme and allows for the use of switches. The semi-Lagrangian scheme is obtained by discretizing (3.15) first in time and then in space: it consists in the backward resolution of a discretized dynamic programming principle The reader may refer to [31] and [25] for the discretization theory. We point out that the semi-Lagrangian scheme does not require to generate scenari (as opposed to other mainstream approaches in stochastic programming), since the Brownian motion is discretized for each time step with deterministic variables (see [18]). For this kind of scheme, the computation burden is exponential in the state dimension (curse of dimensionality), but here this dimension is only two. On the other hand, the complexity increases only linearly with the number of time steps.

3.4 Parameters Estimation

We display in Table 3.1 the numerical values we have for the model. Most of them are those used in [49] and [75]. The data consist in about ten months ($N_{days} = 300$) of historic load and renewable production from Huatacondo. The renewable production data look both smooth and very similar days after days due to the climate in Huatacondo, so we use the average for the optimization and the simulation (see Figure 3.5). We denote by h the time step (15 minutes), $t_k = kh$, σ_k the volatility and $\hat{L}_k$ the historical load at time t_k. The data being discrete, Equation (3.8) becomes, for each day i

$$\hat{L}^i_{k+1} - \hat{L}^i_k = \Lambda_{k+1} - \Lambda_k + b(\Lambda_k - \hat{L}^i_k)h + \sigma_k \epsilon^i_k \sqrt{h}, \tag{3.19}$$

where ϵ^i_k is a standard centered Gaussian variable and $k \in \{1, \dots, 96\}$. Note that as discussed in §3.2.2, Λ_k is the historical average of the load at time t_k, i.e. $\Lambda_k = \sum_{i \in 1..N_{days}} \hat{L}^i_k / N_{days}$.

Table 3.1 – Numerical parameters

Notation	Value
Q_B	117 kWh
A	1320 kW
M	1000000 CLP
$2K$	1000 CLP
$\ell(d,s)$	$500d^{0.9} + 25000\|s\|$ CLP
ρ_O	0.95
ρ_I	0.95
I_d	[5,120] kW
I_{P_I}	[0,13.2] kW
I_{P_O}	[0,40] kW
I_c	[0.2,1] kW

Set for all $k \in \{1, \dots, 96\}$ and $i \in \{1, \dots, N_{days}\}$ $d_{k,i} = \hat{L}^i_k - \Lambda_k$, $b' = hb$ and $\sigma'_k = \sigma_k \sqrt{h}$. Then (3.19) is equivalent to $d_{k+1,i} - d_{k,i}(1-b') = \sigma'_k \epsilon_k$. We then use a mean square estimator. If we consider σ'_k fixed for all k, then b' should minimize $\sum_{k,i}(d^i_{k+1} - d^i_k(1-b'))^2/\sigma'^2_k$, so that $b' = \{\sum_{k,i} d^2_{k,i}/\sigma'^2_k - \sum d_{k,i}d_{k+1,i}/\sigma'^2_k\}/\{\sum d^2_{k,i}/\sigma'^2_k\}$. On the other hand, if we know b', σ'_k is the standard deviation of $d_{k+1,i} - d_{k,i}(1-b)$ computed over the same epoch of the day i on the data. So we start with $\sigma'_k = 1$ and iterate the two formulas until numerical convergence to get our estimators. We get $b' = 0.174$. We display σ in Figure 3.1. We display on Figure 3.2 some random samples of the data and some generated scenari. They qualitatively look alike.

3.5 Simulation

We compare the stochastic extension with the determinist rolling horizon algorithm proposed in [49] on a three day simulation. The simulation procedure is summarized in Figures 3.3 and 3.4.

The rolling horizon for the deterministic algorithm is set to 24 hours and for each horizon we impose the final state of charge to be at least equal to the initial state of charge at the beginning of the horizon. For every time step, we perform an optimization using an updated load forecast for the next 24 hours. We use as a forecast for the k^{th} step the expectation of the flow of the load process with initial condition L_k, where L_k is the corresponding historical Load.

For the stochastic simulation, we solve only once the Hamilton-Jacobi-Bellman equation, and then use the value function and the load historical realization to produce a trajectory. We impose the state of charge to be at least equal to what is obtained with the deterministic simulation at the end of the three day period.

We display on Figures 3.6, 3.8, 3.7 and 3.9 the results for the model with real data. On

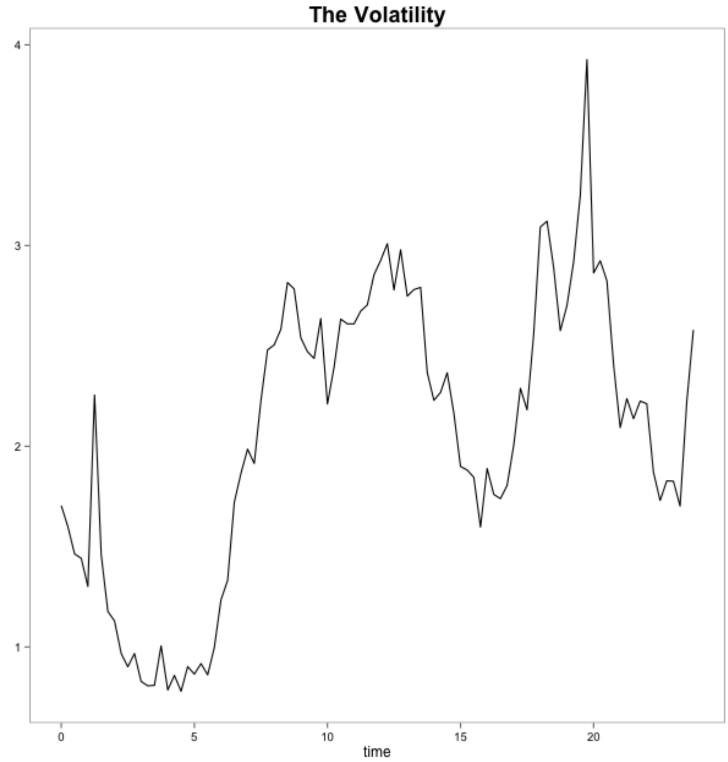


Figure 3.1 – The estimated volatility

our example the slack variable s is always zero so we do not plot it. The total costs for the deterministic and the stochastic simulation are respectively 66819 CLP and 62342 CLP.

3.6 Conclusion

We have extended the deterministic continuous time model for microgrid management to a stochastic setting and performed a numerical experiment on real data from the Huatacondo microgrid. The total cost of the solution proposed by the stochastic algorithm was lower than the one obtained with a deterministic rolling horizon formulation. Ongoing works on this topic include the study of the long-term aging of the battery as well as a jump model for the load process.

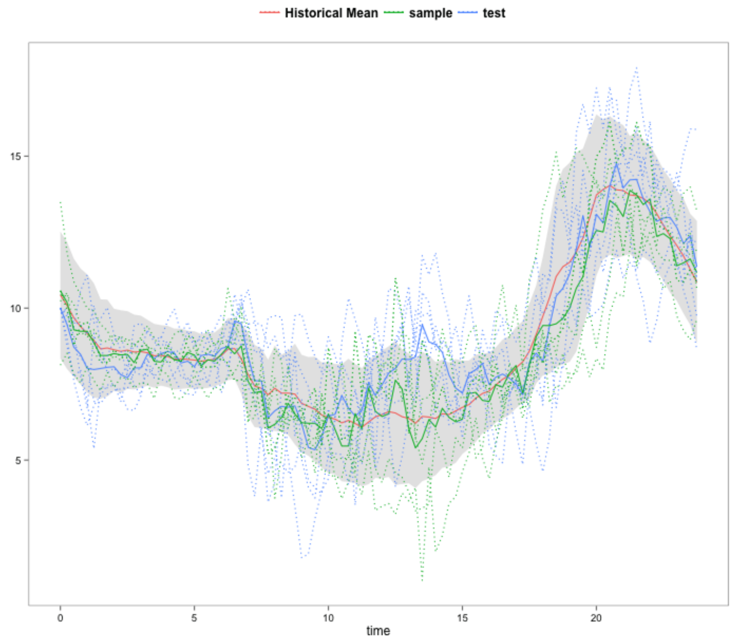


Figure 3.2 – We simulate some scenarios with the load model, and compare them with historical day taken at random. The grey area corresponds to one standard deviation. The unit is the kW.

APPENDIX

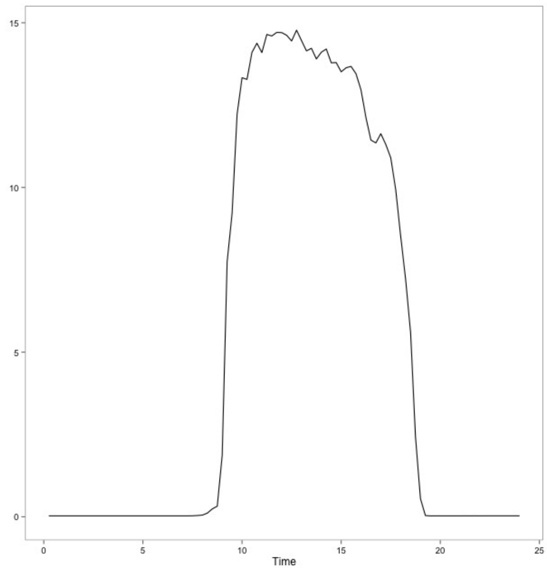


Figure 3.5 – Real model: Solar Production (in kW)

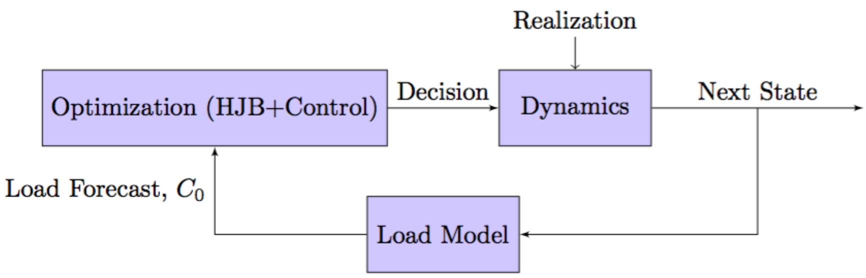


Figure 3.3 – Deterministic simulation algorithm

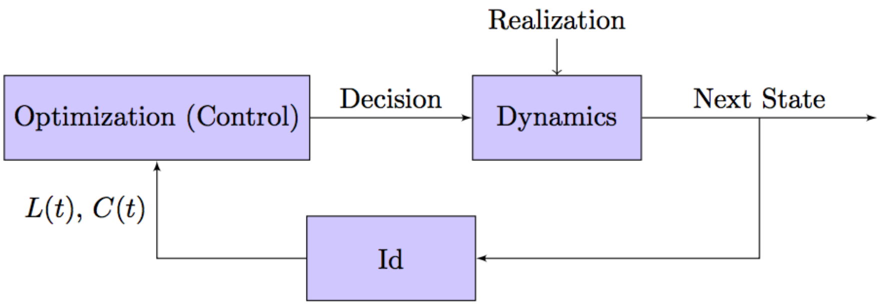


Figure 3.4 – Stochastic simulation algorithm

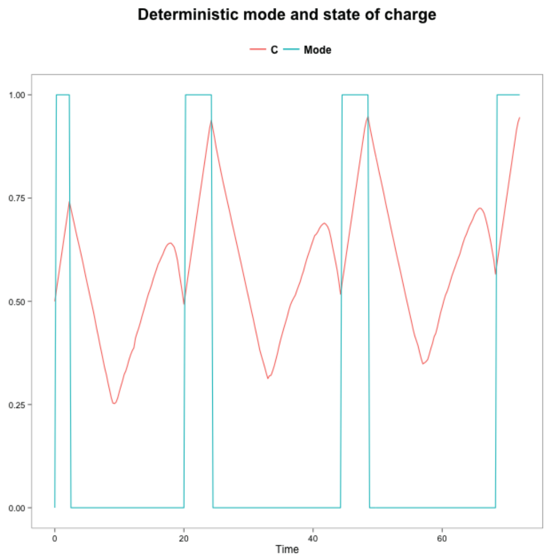


Figure 3.6 – C (in kW) and mode for the deterministic simulation

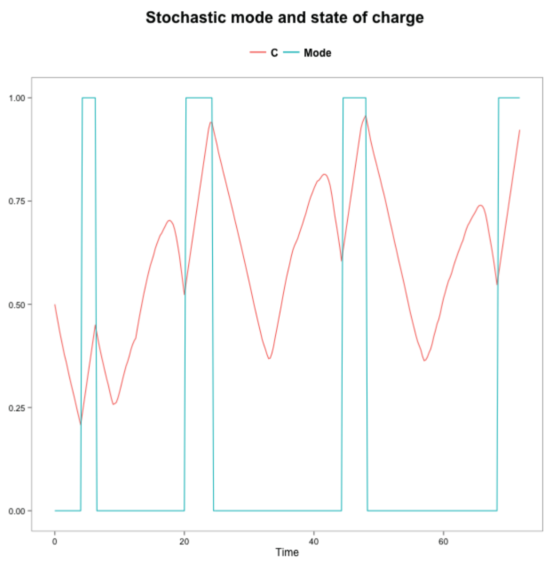


Figure 3.7 – C (in kW) and mode for the stochastic simulation

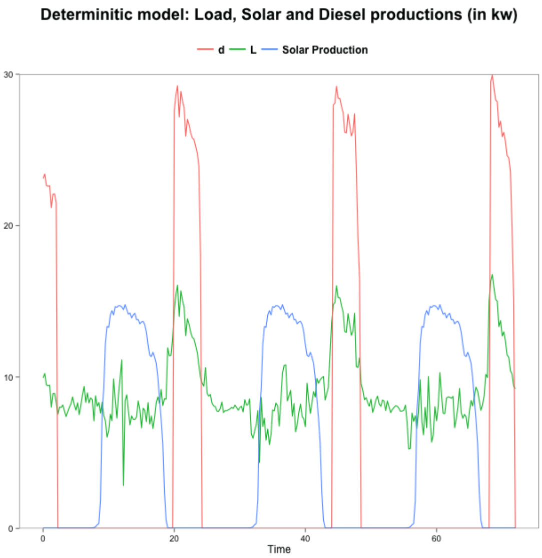


Figure 3.8 – Load and Production for the deterministic simulation (in kW)

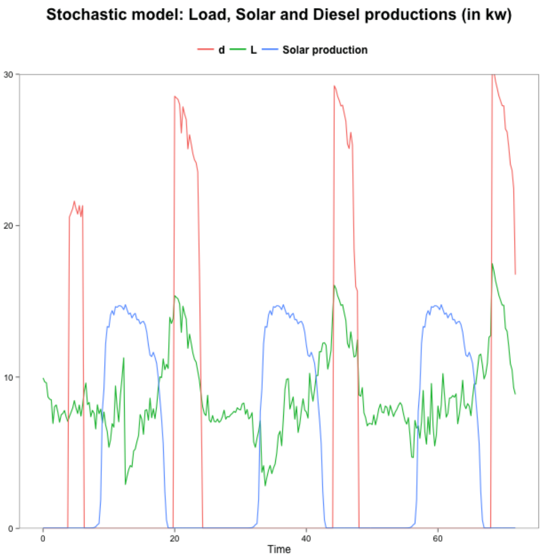


Figure 3.9 – Load and Production for the stochastic simulation (in kW)

Chapter 4

Long Term Aging : An Adaptative Weights Dynamic Programming Algorithm

This is a joint work with Frédéric Bonnans and Pierre Martinon.

Contents

We introduce a class of optimal control problems with periodic data. A state variable that we call the age of the system represents the negative impact of the operations on the system qualities over time: other things being equal, older systems have higher operating costs. Many industrial problems relate to this class. If we envision to perform an optimization over a large number of periods, there is a tradeoff between minimizing repeatedly the one-period criterion in a short sighted way and taking into account the impact of the decision on the aging speed (which modifies the minimal one period criterion). In general, because the aging process is slow, short term optimization strategies- such as one period sliding horizon strategies- either neglect it or use rule-of-thumb penalization terms in the criterion, which leads to suboptimal solutions. On the other hand, for most applications it is unrealistic to envision a brute-force numerical resolution by dynamic programming of the long term problem because of the computation burden. We introduce a two-scale method to reduce this computation burden. The method relies on Lagrangian duality and some monotony properties. We expose the theoretical foundations of the method and discuss some practical aspects: approximation errors, asymptotic estimation, computation burden, possible extensions, etc. Since our initial motivation was the difficulty to take long term battery aging in Energy Management Systems into account, we implement the method on a toy long term microgrid energy management problem.

4.1 Introduction

The aging of physical systems is almost never taken into account in decision making. This can be a cause of sub-optimality. For instance, for many controlled industrial systems, a decision is taken every day. The decision should minimize a tangible criterion, such as the operating cost and is subject to some operating and physical constraints. As time goes on, the system gets older and its physical qualities decrease. On the one hand, this slow aging process depends on how the system is operated. On the other hand the aging of the system is responsible for a loss in efficiency that increases the daily operating cost. Therefore, in an ideal world, the operator should take into account the impact of the daily decision on the long term aging of the system. Nonetheless, this is often technically challenging:

because of the time scales involved, the long term optimization problem requires a lot of time and memory to be solved (curse of dimensionality with respect to the state variables). In this work the system aging process is modeled with a one dimensional age variable a with values in a segment. The physical qualities of the system decrease as a increases. We point out that the choice of an increasing variable to measure the aging process is arbitrary, and we could have defined instead a decreasing health variable. Of course in this case the physical qualities of the system would have been increasing in the health process.

We introduce a class of optimal control problems with periodic data. If we envision to perform an optimization over a large number of periods, there is a trade-off between minimizing repeatedly the one-period criterion in a short sighted way and taking into account the impact of the decision on the aging speed (which will later on in turn have an impact on the minimal one-period criterion). Microgrid energy management relates to the framework proposed in this paper when considering the aging of the battery. We discuss some existing approaches from this literature in §4.6.2. As far as we know, it appears that when taken into account, aging is constrained or penalized but not directly subject to a long term optimization. An important characteristic of our problem is the existence of two time scales. Some other works on multi time scales problem consider averaging techniques (see [26]).

A first essential observation is that if we knew an optimal aging profile over the whole time horizon, then the long term problem could be decomposed into a sum of micro problems with smaller time horizon, say one period. We could then use myopic approaches to solve the resulting one period (or *micro*) problems. The reverse is also true: the long term (or *macro*) problem can be reformulated by means of the micro problems. So we could precompute the solutions of the micro problems off-line to solve the long term aging problem afterward with a *macro* dynamic programming optimization. This is a first step to reduce the problem numerical complexity. It will be detailed in §4.3.

Another essential observation is that aging is a slow process, so we could neglect the age variations within a period to solve simplified micro problems without truly impacting the performances. Indeed, this would decrease by one the number of state variables. Yet it is still necessary to know the age at the end of the period to solve the long term problem. Moreover, since we want to take the age into account in the decision, we should be able to control the total aging over the period. So we look for a method to *control* the total aging over a period without requiring the age to be a state variable in the micro problem *numerical* resolution.

Our approach consists in penalizing the aging over the period in the criterion of the micro problems. Then we map the penalization coefficients with the resulting agings. By doing so, we do not need to keep track of the aging within the micro problems (see §4.4). We then optimize for each period the penalization coefficient in the reformulation of the long term problem. Observe that even if the penalization parameter could be interpreted as a Lagrangian multiplier, we will be performing a *minimization* over this coefficient, which can be confusing. An alternative understanding is to see this penalization parameters as a change of variable: instead of optimizing over the one-period age increments, we optimize over the corresponding penalization coefficients. Figure 4.1 illustrates this idea, with α being the optimization parameter.

To sum up, we propose a decomposition/parametrization method to solve a long term optimal control problem incorporating an age variable. We use the fact that the age (as a state variable) is slow to neglect its variations within a single period to limit the computation burden. We solve a collection of one period, penalized, optimal control problems and associate the resulting total agings with the corresponding coefficients, which allow us to perform a dynamic optimization over the coefficients to solve the long term problem. Observe that we do not fix only one coefficient for the whole problem as in penalization approaches: we control the penalization coefficients of every single period. Then the aging-conscious on-line one-period decision can be computed for an optimized time varying penalization parameter that incorporates the long term aging.

The long term and short term optimal control problems are presented in the next subsection, followed by the technical assumptions we use in this work. We present the bi-level dynamic programming in §4.3 and the adaptive weights approach in §4.4. We discuss the theoretical results and some possible refinements and variations in §4.5. The last section is dedicated to an application of this approach to the microgrid energy management problem. Readers only interested in applications might want to start with this last section.

In the proofs we will use LHS and RHS as shorthands for left-hand side and right-hand side.

4.2 Setting

4.2.1 Problem Formulation

We consider a system with two state variables: the *age* a and the *fast state* c. The fast state should be interpreted as a form of wealth (available cash, energy, inventory). This is why we will refer to the fast state as the *charge*, in reference to this idea and to the toy microgrid model example proposed in the last section. The age a and the charge c follow a T-periodic dynamics controlled by a time dependent parameter u:

$$\begin{cases} \dot{a}(t) = F_a(a(t), c(t), u(t), t) \\ \dot{c}(t) = F_c(a(t), c(t), u(t), t), \end{cases} \tag{4.1}$$

or equivalently if we set $x = (a, c)$ and $F = (F_a, F_c)$:

$$\dot{x}(t) = F(x(t), u(t), t). \tag{4.2}$$

Think of $T > 0$ as the length of a day for instance. The whole horizon T_{tot} is a multiple of T, i.e. $T_{tot} = NT$ with N being a large integer (think of T_{tot} as the length of five years for example). The control u is restricted to be in $U = \{u \text{ s.t. } \forall t \in [0, T_{tot}], u(t) \in \bar{U}\}$, and the charge c and the age a should belong to respectively $\boldsymbol{C}$ and $\mathbb{A}$, where $\bar{U}$ and $\boldsymbol{C}$ are compact subsets of $\mathbb{R}^n$ and $\mathbb{R}$, and there exists (a_-, a_+) such that $\mathbb{A} = [a_-, a_+]$ and $a_- \leq a(0) \leq a_+$.

The age a of the system is non-decreasing in time, i.e. $F_a \geq 0$, and for all $(u, t, c) \in \bar{U} \times [0, T] \times \boldsymbol{C}$, $F_a(a_+, c, u, t) = 0$, which ensures that for all time t, $a(t) \in [a_-, a_+]$. One can interpret $a = a_+$ as the aging component of the system being definitely dead. Then $F_a(a_+, c, u, t) = 0$ means that once this component dead, it cannot get older anymore.

The behaviors of the system change as it get older. In fact, the older the system, the more effort should be needed to complete a given task. The aging process should decrease the *efficiency* of the system. In addition, as c is a form of wealth, the cost-to-go functions of our minimization problems should be decreasing in c. Those notions are expressed in the monotonicity assumption (see. §4.2.2).

While the relevant time scale for the charge c is the period T, the dynamics of the age a is so slow that one need to wait many periods to observe a non-microscopic change in a. Our objective is to minimize the sum of an integral criterion and a non-decreasing final cost $\phi(a_{T_{tot}})$ while verifying some constraints. Like the dynamics, the (bounded) running cost $\ell(u, t)$ is T-periodic. We end up with the following optimal control formulation:

$$P_1(a_0, c_0, t_0) \begin{cases} V_1(a_0, c_0, t_0) = \inf\limits_{u \in U} \displaystyle\int_{t_0}^{T_{tot}} \ell(u(t), t)\mathrm{d}t + \phi(a_{T_{tot}}) \\ (a(t_0), c(t_0)) = (a_0, c_0) \\ a(T_{tot}) \leq a_{max} \\ \forall t \in [t_0, T_{tot}], \varphi(c(t), u(t), t) \in A \\ (\dot{a}(t), \dot{c}(t)) = (F_a(a(t), c(t), u(t), t), F_c(a(t), c(t), u(t), t)), \end{cases} \tag{4.3}$$

where A is a closed set, φ a continuous T periodic-function, and a_{max} a final constraint. Problem P_1 is parametrized by the initial point (a_0, c_0, t_0). The value function V_1 associates any set of initial time and state variables with the corresponding minimal cost-to-go function. Under Lipschitz conditions on F_a and F_c, the ordinary differential equation has a unique solution.The integrand ℓ satisfies the following properties:

- ℓ is bounded
- for all $t \in [0, T]$, $u \to \ell(u, t)$ is continuous
- for all $u \in \bar{U}$, $t \to \ell(u, t)$ is measurable.

We use the standard flow notation i.e. for $\tau > 0$, $X_{x,t}^{u,t+\tau}$ is the value of the solution of the first order ordinary differential equation of the dynamics at time $t + \tau$ when the initial point is $x = (a, c)$ at time t and the control is u. We will use the notation $a_{x,t}^{u,t+\tau}$ (resp. $c_{x,t}^{u,t+\tau}$) to refer to the flow of the age (resp. the charge). When the context will be clear, we will sometimes simply write $X(t)$.

The dynamic programming principle applies and we can write the value function of problem P_1 as the solution of:

$$V_1(a_0, c_0, t_0) = \inf_{u \in U} \int_{t_0}^{t_0+T} \ell(u(t), t)\mathrm{d}t + V_1(X_{x_0,t_0}^{u,t_0+T}, t_0 + T), \tag{4.4}$$

where the infimum is taken over controls such that $\varphi(c_{x_0,0}^{u,t}, u(t), t) \in A$ for any $t \in [t_0, t_0 + T]$. Observe that problem P_1 is not always feasible, the optimization is by construction performed over the feasible $V_1(X_{x_0,t_0}^{u,t_0+T}, t_0 + T)$. Now set $t = t_k = kT$ for $k \in \mathbb{N}$, by T-periodicity of the data, we get the formulation we will use throughout this article

$$V_1(a_0, c_0, t_k) = \inf_{u \in U_{a_0,c_0}} \int_0^T \ell(u(t), t)\mathrm{d}t + V_1(X_{x_0,0}^{u,T}, t_k), \tag{4.5}$$

where U_{a_0,c_0} is the set of controls $u \in U_T = \{u \in L^\infty(0, T) \text{ s.t. } \forall t \in [0, T], u(t) \in \bar{U}\}$ such that $\varphi(c_{x_0,0}^{u,t}, u(t), t) \in A$ for any $t \in [0, T]$.

We propose a bi-level approach to solve problem P_1. First we introduce a collection of *micro* problems

$$P_1^\mu(a_0, \delta a, c_0, c_F) \begin{cases} V_1^\mu(a_0, \delta a, c_0, c_F) = \inf_{u \in U_T} \int_0^T \ell(u(t), t)\mathrm{d}t \\ (a(0), c(0)) = (a_0, c_0) \\ a(T) \leq a_0 + \delta a \\ c(T) \geq c_F \\ \forall t \in [0, T], \varphi(c(t), u(t), t) \in A. \\ (\dot{a}(t), \dot{c}(t)) = (F_a(a(t), c(t), u(t), t), F_c(a(t), c(t), u(t), t)). \end{cases} \tag{4.6}$$

The superscript μ stands for *micro.* The micro problem P_1^μ is in many ways similar to P_1: the dynamics, the mixed constraints and the integrand of the criterion are the same. The difference is that the time horizon is only one period. Moreover, the final state condition on the age a and the final cost $\phi(a_{T_{tot}})$ are replaced by two final state conditions on the age a and the state c. The problem is parametrized by δa, which represents the maximal amount of aging for the period. If $\phi = 0$ then for both P_1 and P_1^μ the goal is to minimize the integral of the running cost ℓ over a time horizon without increasing the age of the system a by more than a given quantity.

4.2.2 Assumptions

We make the following assumptions.

Assumption 1 (Slow aging). *There exists a constant $L > 0$ such that*

- F_c *is L-Lipschitz and uniformly bounded by L,*
- F_a *is L/N-Lipschitz and uniformly bounded by L/N.*

Assumption 1 expresses that the aging process is slow. The dependence of the F_a Lipschitz constant and bound in N (which is the number of periods in the macro problem P_1) is needed to perform an asymptotic estimation. (see §4.5.1). Assumption 1 is used in the proof of Lemma 4.15.

Assumption 2 (Monotonicity). *The value functions V_1^μ and V_1 are non decreasing in a_0 and non increasing in c_0.*

Assumption 2 corresponds to the fact that youth and wealth are always preferable for the systems envisionned.

Assumption 3 (Regularity of the aging process). *For any $\epsilon > 0$, $\Delta > 0$, there exists $\epsilon_1 > 0$ such that*
if $x_0 = (a_0, c_0) \in \mathbb{A} \times \boldsymbol{C}$, $u \in U_{x_0}$ *and* $\Delta \leq \int_0^T F_a(X(t), u(t), t)\mathrm{d}t \leq \Delta + \epsilon_1$,
then there exists $u' \in U_{x_0}$ *such that*

$$\int_0^T F_a(X(t), u'(t), t)\mathrm{d}t = \Delta \quad \textit{and} \quad |\int_0^T [\ell(u(t), t) - \ell(u'(t), t)]\mathrm{d}t| \leq \epsilon. \tag{4.7}$$

We can interpret Δ as a maximum aging level. Assumption 3 ensures that if we set a maximum aging level and a precision level, then we can modify any “almost admissible” control for the maximum aging level into an admissible one, and the change in integral cost will not be more than the precision level. This Assumption is used in the proof of Lemma 4.1.

4.3 Bilevel Dynamic Programming

4.3.1 Mathematical Justification

We start with the following (intuitive) result.

Lemma 4.1. *For all $(k, a_0, c_0) \in \mathbb{N} \times \mathbb{A} \times \boldsymbol{C}$,*

$$V_1(a_0, c_0, t_k) = \inf_{\delta a \in \mathbb{R}^+, c_F \in \boldsymbol{C}} V_1^\mu(a_0, \delta a, c_0, c_F) + V_1(a_0 + \delta a, c_F, t_{k+1}). \tag{4.8}$$

Proof. <u>$LHS \geq RHS$</u>: Set $x_0 = (c_0, a_0)$. First we establish that $LHS \geq RHS$. Take a control $u \in U_{x_0}$ and set $\delta a = a^{u,T}_{x_0,0} - a_0$ and $c_F = c^{u,T}_{x_0,0}$. By construction of δa and c_F, u is admissible for $P^\mu_1(a_0, \delta a, c_0, c_F)$ so by definition of V^μ_1, $\int_0^T \ell(u(t), t)\mathrm{d}t \geq V^\mu_1(c_0, c_F, a_0, \delta a)$. Moreover, trivially $V_1(X^{u,T}_{x_0,0}, t_{k+1}) = V_1(a_0 + \delta a, c_F, t_{k+1})$ therefore

$$\int_0^T \ell(u(t), t)\mathrm{d}t + V_1(X^{u,T}_{x_0,0}, t_{k+1}) \geq V^\mu_1(c_0, c_F, a_0, \delta a) + V_1(a_0 + \delta a, c_F, t_{k+1}) \geq RHS. \tag{4.9}$$

Since this is true for any $u \in U_{x_0}$, we can apply the dynamic programing principle (4.5) to get $LHS \geq RHS$. We point out that here we did not use the existence of minimizers.

<u>$LHS \leq RHS$</u> Take δa and c_F an ϵ-optimal decision for the RHS and $u \in U_{x_0}$ an ϵ-optimal control for $P^\mu_1(a_0, \delta a, c_0, c_F)$. By definition of $P^\mu_1(a_0, \delta a, c_0, c_F)$ and admissibility of u, $c^{u,T}_{x_0,0} \geq c_F$ and $a^{u,T}_{x_0,0} \leq a_0 + \delta a$. By ϵ-optimality,

$$RHS + 2\epsilon \geq \int_0^T \ell(u(t), t)\mathrm{d}t + V_1(a_0 + \delta a, c_F, t_{k+1}) \tag{4.10}$$

By monotonicity of V_1,

$$V_1(a + \delta a, c_F, t_{k+1}) \geq V_1(a + \delta a, c^{u,T}_{x_0,0}, t_{k+1}) \geq V_1(a^{u,T}_{x_0,0}, c^{u,T}_{x_0,0}, t_{k+1},). \tag{4.11}$$

Therefore,

$$RHS + 2\epsilon \geq \int_0^T \ell(u(t), t)\mathrm{d}t + V_1(X^{u,T}_{x_0,0}, t_{k+1}) \geq LHS, \tag{4.12}$$

where we used the dynamic programming principle (4.5) and the fact that $u \in U_{x_0}$ for the last inequality. We can conclude that $RHS = LHS$. □

4.3.2 Complexity Analysis

We proceed with a complexity analysis of the previous results. Assume we characterize the discretization of the space and time grid with the integer parameters N_a, N_c, N_u and N_t, which are the discretization levels of a, c, u and one unit of time. On the one hand, if we solve problem P_1 directly by dynamic programming (for a fixed initial state), the computation burden is proportional to $N_a N_c N_u N_t T N$. On the other hand if we use Lemma 4.1, we first solve P^μ_1 offline for each possible parameters. Problem P^μ_1 computation burden for one numerical resolution is proportional to $N_a N_c N_u N_t T$, but we need to solve it for each final parameters $(c_F, a_F = a_0 + \delta a)$, i.e. $N_a N_c$ times (indeed, observe that one resolution solves the problem for all possible initial states). Then we have to do the macro resolution ($O(N_a N_c N)$). So the total cost with the micro/macro formulation is $O(N_a^2 N_c^2 N_u N_t T) + O(N_a N_c N)$, which can be competitive against a brute-force dynamic programming if N is large compared to $N_a N_c$.

Consider a specific case where the charge c has to be the same at the beginning of each period. In this case we get:

- Direct dynamic programming resolution: $O(N_a N_c N_u N_t T N)$
- One micro problem resolution: $O(N_a N_c N_u N_t T)$
- Number of parameters for the micro problem: $O(N_a)$
- Resolution of the macro problem: $O(N_a^2 N)$
- Total computation burden for the bilevel approach: $O(N_a^2 N) + O(N_a^2 N_c N_u N_t T)$

Then this approach becomes competitive compared to the direct DP approach if N is larger than N_a. Note that when solving the micro problem, the complexity is proportional to N_a, but the grid does not need to contain the whole $\mathbb{A}$ segment for a given set of final ages. The pseudo code of such an algorithm is straightforward.

- Compute the value function of P_1^μ for all δa.
- Compute the value function of the macro problem using the previous results.

Now assume we do an online implementation (with periodic $c_F = \hat{c}_F$): at the beginning of each period, we compute an optimal control for the period, with a final constraint $c_F = \hat{c}_F$. With direct dynamic programming, either we recompute the value function every time (a), or we keep it in memory (which requires a lot of resources)(b). If we keep the whole value function in memory, it requires a memory space proportional to $N_a N_c N_t N T$, which is not realistic. If we only keep the value function at the end of each period, then we only need a memory space proportional to $N_a N$, but we need to recompute the intermediate value function for a computation burden of $N_a N_c N_u N_t T$.

With a bilevel approach, we can keep in memory either the whole value function of the macro problem (c), the value functions of the micro problems at $t = 0$ for different maximal aging (d), or the mapping of the optimal δa for each (a, t) (e). The first solution is similar to (a) and is not realistic. The second solution requires a memory space of $O(N_a^2)$, which is not proportional to N. Then we need to get the optimal aging by dynamic programming on the macro problem $O(N_a^2 N)$ and compute an optimal control $O(N_a N_c N_u N_t T)$. The last possibility requires a $N_a \times N$ table. The online computation of the control will then require two states, which represents a computation burden proportional to $N_a N_c N_u N_t T$. Note that this last possibility is equivalent to (b) for the online phase.

The complexity analysis is summarized in Tables 4.1 and 4.2.

We point out that we have not used much of the problem specificity, as only Assumptions 2 is needed in the proof of Lemma 4.1. Moreover, such online computation burden may be too big for some applications. This is what motivates the next section.

4.4 Adaptative Weights

4.4.1 Preliminary Results

Note that we could replace the final constraint and the final cost in P_1 by a penalization on the age variation. We would get the criterion

$$\int_{t_0}^{T_{tot}} \ell(u(t),t)\mathrm{d}t + \alpha[a(T_{tot}) - a_{max}] + \phi(a_{T_{tot}}). \tag{4.13}$$

where $\alpha \in \mathbb{R}_+$ is a penalization coefficient. Since a_{max} and $\hat{a}_0$ are constants over which we are not optimizing, we get an equivalent optimization problem by using instead the criterion

$$\int_{t_0}^{T_{tot}} \ell(u(t),t)dt + \alpha \int_{t_0}^{T_{tot}} F_a(X(t),u(t),t)\mathrm{d}t + \phi(a_{T_{tot}}). \tag{4.14}$$

We then get the penalized problem (with free final age)

$$P_2(a_0,c_0,\alpha,t_0) \begin{cases} V_2(a_0,c_0,\alpha,t_0) = \inf\limits_{u\in U} \displaystyle\int_{t_0}^{T_{tot}} [\ell(u(t),t) + \alpha F_a(X(t),u(t),t)]dt + \phi(a_{T_{tot}}) \\ \dot{X}(t) = F(X(t),u(t),t) \\ (a(t_0),c(t_0)) = (a_0,c_0) \\ \varphi(c(t),u(t),t) \in A \end{cases} \tag{4.15}$$

Let us introduce the corresponding micro-problem:

$$P_2^\mu(a_0,c_0,c_F,\alpha) \begin{cases} V_2^\mu(a_0,c_0,c_F,\alpha) = \inf\limits_{u\in U} \displaystyle\int_0^T [\ell(u(t),t) + \alpha F_a(X(t),u(t),t)]\mathrm{d}t \\ \dot{X}(t) = F(X(t),u(t),t) \\ (a(0),c(0)) = (a_0,c_0) \\ \varphi(c(t),u(t),t) \in A \\ c(T) \geq c_F. \end{cases} \tag{4.16}$$

As the notations implies, P_2^μ is to P_2 what P_1^μ is to P_1: a one day version. Just note that the final constraint in P_1 is considered as fixed and is a parameter in P_1^μ whereas the penalization coefficient is a parameter for both P_2^μ and P_2. In addition, note that we cannot write a dynamic programming principle directly with P_2^μ and P_2 as we did for P_1^μ and P_1 in Lemma 4.1.

If problem P_1 is strictly feasible, then for α big enough the final constraint is satisfied by the solutions of $P_2(a_0,c_0,\alpha,t_0)$ (see the proof of Lemma 4.6). A standard way to deal with aging is to replace $P_1(a_0,c_0,t_0)$ by an approximation of $P_2(a_0,c_0,c_F,\alpha)$ where the age a is fixed:

$$\tilde{P}_2(a_0,c_0,\alpha,t_0)\begin{cases} V_2(a_0,c_0,\alpha,t_0)=\inf_{u\in U}\int_{t_0}^{T_{tot}}[\ell(u(t),t)+\alpha F_a(a_0,c(t),u(t),t)]dt+\phi(a_{T_{tot}}) \\ \dot{c}(t)=F_c(a_0,c(t),u(t),t) \\ (a(t_0),c(t_0))=(a_0,c_0) \\ \varphi(c(t),u(t),t)\in A \end{cases} \tag{4.17}$$

To set α, practitioners would often compute a collection of solutions of $\tilde{P}_2$ for different values of α, and then take the best admissible solution in the sense of P_1. Nonetheless such an approach neglects the impact of the aging on the efficiency and more generally the fact that the dynamics are coupled.

In addition, there may be a duality gap. In this case any (theoretical) solution of P_2 admissible for P_1 is suboptimal for P_1. Ekeland and Aubin propose an estimate of this sub-optimality in a finite dimensional setting [6].

The following lemma is a relation between the value functions of the two micro problems P_2^μ and P_1^μ.

Lemma 4.2. *For any* $(a_0,c_0,c_F,\alpha)\in\mathbb{A}\times\boldsymbol{C}^2\times\mathbb{R}_+$,

$$V_2^\mu(a_0,c_0,c_F,\alpha)=\inf_{\delta a\in\mathbb{R}_+}V_1^\mu(a_0,\delta a,c_0,c_F)+\alpha\delta a \tag{4.18}$$

Proof. $\underline{LHS\geq RHS}$:
Take $\epsilon>0$, $u\in U_{x_0}$ an ϵ-optimal solution of the LHS. Set $\delta a=a_{x_0,0}^{u,T}-a_0$. By ϵ-optimality of u

$$LHS+\epsilon\geq\int_0^T\ell(u(t),t)\mathrm{d}t+\alpha\delta a. \tag{4.19}$$

In addition note that u is admissible for $P_1^\mu(a_0,\delta a,c_0,c_F)$ therefore by definition of V_1^μ, $\int_0^T\ell(u(t),t)\mathrm{d}t\geq V_1^\mu(a_0,\delta a,c_0,c_F)$. Then

$$LHS+\epsilon\geq V_1^\mu(a_0,\delta a,c_0,c_F)+\alpha\delta a\geq\inf_{\delta a\in\mathbb{R}_+}V_1^\mu(a_0,\delta a,c_0,c_F)+\alpha\delta a=RHS. \tag{4.20}$$

Therefore $LHS\geq RHS$.
$\underline{LHS\leq RHS}$:
Take $\epsilon>0$, $\delta a\in\mathbb{R}_+$ an ϵ-optimal solution of the RHS and $u\in U_{x_0}$ an ϵ-optimal solution of problem $P_1^\mu(a_0,\delta a,c_0,c_F)$. By ϵ-optimality

$$RHS+\epsilon\geq V_1^\mu(a_0,\delta a,c_0,c_F)+\alpha\delta a\quad\text{and}\quad V_1^\mu(a_0,\delta a,c_0,c_F)+\epsilon\geq\int_0^T\ell(u(t),t)\mathrm{d}t, \tag{4.21}$$

so

$$RHS \geq \int_0^T \ell(u(t),t)\mathrm{d}t + \alpha\delta a - 2\epsilon. \tag{4.22}$$

Moreover, since u is $P_1^\mu(a_0, \delta a, c_0, c_F)$-admissible, $\delta a \geq \int_0^T F_a(X(t), u(t), t)\mathrm{d}t$, so

$$RHS \geq \int_0^T [\ell(u(t),t) + \alpha F_a(X(t), u(t), t)]\mathrm{d}t - 2\epsilon \geq LHS - 2\epsilon. \tag{4.23}$$

Therefore $LHS \leq RHS$ and the proof is done. □

We point out that this result does not depend on ϕ and a_{max} which are macro problem specific parameters.

Corollary 4.3. *For any* $(a_0, c_0, c_F, \delta a, \alpha) \in \mathbb{A} \times \boldsymbol{C}^2 \times \mathbb{R}_+^2$,

$$V_1^\mu(a_0, \delta a, c_0, c_F) \geq V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha\delta a. \tag{4.24}$$

4.4.2 Nice Case: No Duality Jumps

To proceed we need some additional notations. First for any $x_0 \in \mathbb{A} \times \boldsymbol{C}$, and any control $u \in U_{x_0}$, we define

$$\Delta a(u) = \int_0^T F_a(X(t), u(t), t))\mathrm{d}t \quad \text{and} \quad \mathcal{L}(u) = \int_0^T \ell(u(t), t)\mathrm{d}t. \tag{4.25}$$

Note that for readability the initial conditions are kept implicit. For any $(a_0, c_0, c_F, \alpha) \in \mathbb{A} \times \boldsymbol{C}^2 \times \mathbb{R}_+$, let

$$\Gamma(a_0, c_0, c_F, \alpha) = \left\{ \begin{array}{l} \delta a = \lim_n \Delta a(u_n); \\ u_n \text{ minimizing sequence of } P_2^\mu(a_0, c_0, c_F, \alpha) \end{array} \right\} \tag{4.26}$$

Roughly speaking, given some $(a_0, c_0, c_F) \in \mathbb{A} \times \boldsymbol{C}^2$, $\alpha \to \Gamma(a_0, c_0, c_F, \alpha)$ associates the penalization coefficients with the set of optimal aging levels for P_2^μ. Lemma 4.4 is a key result for what follows.

Lemma 4.4. *Let* $(a_0, c_0, c_F, \alpha, \Delta a) \in \mathbb{A} \times \boldsymbol{C}^2 \times \mathbb{R}_+^2$ *such that* $\Delta a \in \Gamma(a_0, c_0, c_F, \alpha)$. *Then*

$$V_1^\mu(a_0, \Delta a, c_0, c_F) = V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha\Delta a. \tag{4.27}$$

Proof. We deduce $LHS \geq RHS$ from Corollary 4.3, so we only need to show that $LHS \leq RHS$. Let $\epsilon > 0$. By Assumption 3, there exists $\epsilon_1 \geq 0$ such that if $u_2 \in U_{x_0}$ and $\Delta a \leq \Delta a(u_2) \leq \Delta a + \epsilon_1$, then there exists $u' \in U_{x_0}$ such that

$$\Delta a(u') = \Delta a \quad \text{and} \quad |\mathcal{L}(u_2) - \mathcal{L}(u')| \leq \epsilon. \tag{4.28}$$

By hypothesis, $\Delta a \in \Gamma(a_0, c_0, c_F, \alpha)$ so by definition of Γ there exists u_2 ϵ-optimal for $P_2^\mu(a_0, c_0, c_F, \alpha)$ that satisfies $|\Delta a(u_2) - \Delta a| \leq \min(\epsilon_1, \frac{\epsilon}{\alpha})$.

If $\Delta a(u_2) \leq \Delta a$, then

$$V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha\Delta a \geq V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha\Delta a(u_2) - \epsilon \geq \tag{4.29}$$

$$-2\epsilon + \mathcal{L}(u_2) \geq -2\epsilon + V_1^\mu(c_0, c_F, a_0, \Delta a). \tag{4.30}$$

We used $\alpha|\Delta a(u_2) - \Delta a| \leq \epsilon$ for the first inequality, the ϵ-optimality of u_2 for the second inequality, and the admissibility of u_2 for $P_1^\mu(c_0, c_F, a_0, \Delta a)$ in the third.

Else we have the existence of a control u' satisfying (4.28). Then by ϵ-optimality of u_2 for P_2^μ, (4.28) and $\Delta a(u_2) > \Delta a$, and the fact that u' is admissible for P_1^μ and $\Delta a(u') = \Delta a$:

$$V_2^\mu(a_0, c_0, c_F, \alpha) \geq -\epsilon + \mathcal{L}(u_2) + \alpha\Delta a(u_2) \geq \tag{4.31}$$

$$-2\epsilon + \mathcal{L}(u') + \alpha\Delta(u') \geq \tag{4.32}$$

$$-2\epsilon + V_1^\mu(c_0, c_F, a_0, \Delta a) + \alpha\Delta a. \tag{4.33}$$

Then we can conclude. $\square$

From Lemma 4.4 it is trivial that

Corollary 4.5. *Let $(a_0, c_0, c_F, \alpha, \delta a, u) \in \mathbb{A} \times \boldsymbol{C}^2 \times \mathbb{R}_+^2 \times U_T$ and $\epsilon > 0$ such that u is an ϵ-optimal solution of $P_1^\mu(a_0, \delta a, c_0, c_F)$ and $\Delta a(u) \in \Gamma(a_0, c_0, c_F, \alpha)$ then*

$$V_1^\mu(a_0, \Delta a(u), c_0, c_F) = V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha\Delta a(u) \tag{4.34}$$

Now we have the tools to prove one of our main results.

Theorem 4.6. *Let $(a_0, c_0, \alpha_n, u) \in \mathbb{A} \times \boldsymbol{C} \times \mathbb{R}_+^n \times U$ and $\epsilon > 0$ such that u is an ϵ-optimal solution for $P_1(a_0, c_0, 0)$ and $(a_{k+1} - a_k) \in \Gamma(a_k, c_k, c_{k+1}, \alpha_n)$, where $a_k = a_{x_0,0}^{u,t_k}$ and $c_k = c_{x_0,0}^{u,t_k}$. Then for all $k = 0 \ldots N-1$*

$$|V_1(a_k, c_k, t_k) - \inf_{(\alpha, c_F, \delta a)} \{V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha\delta a + V_1(a_k + \delta a, c_F, t_{k+1})\}| \leq \epsilon, \tag{4.35}$$

where the optimization is performed over the $(\alpha, \delta a, c_F)$ such that $\alpha \in \mathbb{R}_+$, $c_F \in \boldsymbol{C}$ and $\delta a \in \Gamma(a_k, c_k, c_F, \alpha)$.

Proof. Without loss of generality, we deal with the case $k = 0$. First note that the restriction u_T of u to $[0, T]$ is an ϵ-optimal solution for $P_1^\mu(a_0, a_1 - a_0, c_0, c_1)$.

By ϵ-optimality, admissibility of u_T for $P_1^\mu(a_0, a_1 - a_0, c_0, c_1)$, Corollary 4.5 and the fact that $a_1 - a_0 \in \Gamma(a_0, c_0, c_1, \alpha_0)$,

$$V_1(a_0, c_0, 0) + \epsilon \geq \mathcal{L}(u_T) + V_1(a_1, c_1, T) \geq V_1^{\mu}(a_0, a_1 - a_0, c_0, c_1) + V_1(a_1, c_1, T) \tag{4.36}$$

$$\geq V_2^{\mu}(a_0, c_0, c_1, \alpha_0) - \alpha_0(a_1 - a_0) + V_1(a_0 + a_1 - a_0, c_1, T) \tag{4.37}$$

$$\geq \inf_{\alpha, \delta a, c_F} V_2^{\mu}(a_0, c_0, c_F, \alpha) - \alpha\delta a + V_1(a_0 + \delta a, c_F, T) \tag{4.38}$$

Therefore $RHS \leq LHS + \epsilon$.

For any $(\alpha, \delta a, c_F) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \boldsymbol{C}$ such that $\delta a \in \Gamma(a_0, c_0, c_F, \alpha)$, by Lemma 4.4 and Lemma 4.1:

$$V_2^{\mu}(a_0, c_0, c_F, \alpha) - \alpha\delta a + V_1(a_0 + \delta a, c_F, T) = \tag{4.39}$$

$$V_1^{\mu}(a_0, \delta a, c_0, c_F) + V_1(a_0 + \delta a, c_F, T) \geq V_1(a_0, c_0, 0). \tag{4.40}$$

Therefore $RHS \geq LHS$ and the conclusion follows. □

We point out that the result is still true if we want to fix the c_k as operational constraints. Under the hypothesis of the previous theorem, assume that Γ is a singleton, then V_1 can be computed by dynamic programming over α. We will get a similar result in the next section as a consequence of Theorem 4.13. Note that since in P_2^{μ} there is no final constraint on the age a we could approximate this problem by fixing the age in the dynamics. With Assumption 1 we should be able to get an error estimate. Last but not least, beware that the optimal α in Theorem 4.6 is not the α that would relate by duality two macro problems P_1 and P_2.

4.4.3 Some comments on Γ

We say that $P_1^{\mu}(a_0, \delta a, c_0, c_F)$ is strictly feasible if there exists $u \in U_{x_0}$ such that $\Delta a(u) < \delta a$. We start with the following classical result:

Lemma 4.7. *If problem $P_1^{\mu}(a_0, \delta a, c_0, c_F)$ is strictly feasible, then there is an α_0 such that for α bigger than α_0,*

$$\forall \Delta a \in \Gamma(a_0, c_0, c_F, \alpha), \Delta a \leq \delta a. \tag{4.41}$$

Proof. Since $P_1^{\mu}(a_0, \delta a, c_0, c_F)$ is strictly feasible, there exists $u \in U_{x_0}$ such that $\Delta a(u) < \delta a$. Assume that there exists α_n an increasing sequence such that

$$\alpha_n \to +\infty \text{ and } \forall n \in \mathbb{N}, \exists \Delta a_n \in \Gamma(a_0, c_0, c_F, \alpha_n), \Delta a_n > \delta a. \tag{4.42}$$

Then

$$n(\Delta a_n - \Delta a(u)) \geq n(\delta a - \delta a(u)) \tag{4.43}$$

Then for n big enough, since ℓ is bounded, we would have $\mathcal{L}(u)+n\Delta a(u) < V_2^\mu(a_0, c_0, c_F, n)$, which is absurd. □

The result of the next lemma is clear to the intuition: Γ should be decreasing in α.

Lemma 4.8. *Let $(a_0, c_0, c_F) \in \mathbb{A} \times \boldsymbol{C}^2$, then for any $0 \leq \alpha_1 < \alpha_2$,*

$$(\delta_1, \delta_2) \in \Gamma(a_0, c_0, c_F, \alpha_1) \times \Gamma(a_0, c_0, c_F, \alpha_2) \Rightarrow \delta_1 \geq \delta_2 \tag{4.44}$$

Proof. To simplify, we omit (a_0, c_0, c_F) because they do not intervene in the proof. Assume that for some non-negative $\alpha_1 < \alpha_2$, there exist some (δ_1, δ_2) such that $\delta_1 < \delta_2$. Then by corollary 4.3

$$V_2^\mu(\alpha_2) \leq V_1^\mu(\delta_1) + \alpha_2\delta_1. \tag{4.45}$$

This implies with Lemma 4.4 that

$$\alpha_2(\delta_2 - \delta_1) \leq V_1^\mu(\delta_1) - V_1^\mu(\delta_2) \tag{4.46}$$

Then since $\alpha_1 < \alpha_2$ and $\delta_2 - \delta_1 > 0$

$$\alpha_1(\delta_2 - \delta_1) < V_1^\mu(\delta_1) - V_1^\mu(\delta_2) \tag{4.47}$$

which in turn implies that

$$V_1^\mu(\delta_2) + \alpha_1\delta_2 < V_1^\mu(\delta_1) + \alpha_1\delta_1 = V_2^\mu(\alpha_1), \tag{4.48}$$

which is not coherent with the optimality of $V_2^\mu(\alpha_1)$. We conclude that $\delta_1 \geq \delta_2$. □

Since the data are bounded, Γ is included in a compact set. Since P_2^μ has a value, and $u \to \Delta(u)$ is valued in a compact set, Γ is not empty. We display in Figure 4.1 a sketch of Γ as a function of α (the other variables being fixed). By Lemma 4.8 Γ is non-increasing. There is no reason a priori why it could not be locally constant. Indeed, even the solution of P_2^μ could be locally constant with respect to α. Observe that Γ not necessarily a singleton: we can have some jumps. In addition it is not necessarily convex valued.

Lemma 4.9. *Let $(a_0, c_0, c_F, \alpha, \Delta a_1, \Delta a_2) \in \mathbb{A} \times \boldsymbol{C}^2 \times \mathbb{R}_+^2$ such that $\Delta a_i \in \Gamma(a_0, c_0, c_F, \alpha)$ for $i = 1, 2$ then*

$$V_1^\mu(a_0, \Delta a_1, c_0, c_F) - V_1^\mu(a_0, \Delta a_2, c_0, c_F) = \alpha(\Delta a_2 - \Delta a_1) \tag{4.49}$$

Proof. This is a direct consequence of Lemma 4.4. □

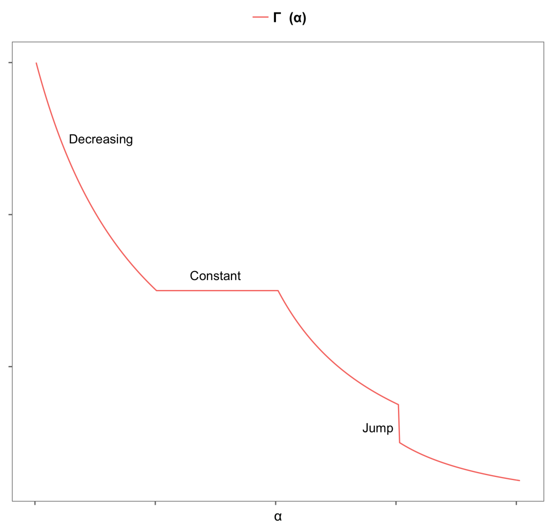


Figure 4.1 – AN EXAMPLE OF $\Gamma(\alpha)$ PROFILE: This drawing summarize the possible behaviors of $\Delta a(\alpha)$, i.e the influence of the penalization parameter over the aging in the micro problem. It can be either continuous and strictly decreasing or constant, or it may "jump" for a given value of α. If there is a jump at $\alpha = \alpha_0$ then we cannot a priori say that $\Gamma(\alpha_0)$ is a singleton. In addition, it may happen that Γ 'misses' some values at α_0.

4.4.4 Generic Case

In general we do not have any guarantee of the existence of the α's as in Theorem 4.4. In relation with problem P_2^{μ}, we introduce for any $(x_0, c_F, \alpha) \in (\mathbb{A} \times \boldsymbol{C}) \times \boldsymbol{C} \times \mathbb{R}_+$

$$\Delta^-(x_0, c_F, \alpha) = \inf_{u \in S(x_0, c_F, \alpha)} \liminf_{n \to +\infty} \Delta a(u_n), \tag{4.50}$$

and

$$\Delta^+(x_0, c_F, \alpha) = \sup_{u \in S(x_0, c_F, \alpha)} \limsup_{n \to +\infty} \Delta a(u_n), \tag{4.51}$$

where

$$S(x_0, c_F, \alpha) = \{u \in (U_T)^{\mathbb{N}}; (u_n)_{n \in \mathbb{N}} \text{ minimizing sequence of } P_2^{\mu}(x_0, c_F, \alpha)\}. \tag{4.52}$$

In addition, we denote by $\hat{\Gamma}$ the set

$$\hat{\Gamma}(x_0, c_F, \alpha) = \{\Delta^-(x_0, c_F, \alpha), \Delta^+(x_0, c_F, \alpha)\} \tag{4.53}$$

which is either a 2-uplet if there is a jump, and a singleton otherwise. The set $\hat{\Gamma}$ corresponds to the minimal and maximal optimal age increments for a given α. Last, we denote by γ

the quantity:

$$\gamma = \sup_{(x_0,c_F,\alpha)\in(\mathbb{A}\times\boldsymbol{C})\times\boldsymbol{C}\times\mathbb{R}_+} \{\lim_{\alpha^-<\alpha} \Delta^+(x_0,c_F,\alpha^-) - \lim_{\alpha^+>\alpha} \Delta^-(x_0,c_F,\alpha^+)\}, \tag{4.54}$$

which corresponds to the size of the biggest possible duality jump. By monotonicity of Γ with respect to α, γ is non negative, and by Assumption 1, γ is finite. We denote by $P_3(x_k,t_k)$ for $k = 0 \dots N-1$ the problem

$$V_3(x_k,t_k) = \inf_{(x_i,\alpha_i)} \sum_{i=k}^{N-1} \{V_2^\mu(x_i,c_{i+1},\alpha_i) - \alpha_i(a_{i+1}-a_i)\} + \tilde{\phi}(a_N), \tag{4.55}$$

where the optimization is performed over the $(x_i,\alpha_i)_{i=k+1,\dots,N} \in (\mathbb{A}\times\boldsymbol{C}\times\mathbb{R}_+)^{N-k}$ such that for any $i = k \dots N-1$,

$$(a_{i+1}-a_i) \in \hat{\Gamma}(x_i,c_{i+1},\alpha_i) \tag{4.56}$$

and $\tilde{\phi}(a) = \phi(a)$ if $a < a_{max}$, $+\infty$ else.

Lemma 4.10. *Let* $(a_0,c_0,c_F,\alpha,\delta a) \in \mathbb{A}\times\boldsymbol{C}^2\times R_+^2$, *such that* $\delta a \in \hat{\Gamma}(a_0,c_0,c_F,\alpha)$, *then*

$$V_1^\mu(a_0,c_0,c_F,\delta a) = V_2^\mu(a_0,c_0,c_F,\alpha) - \alpha\delta a \tag{4.57}$$

Proof. We just observe that $\hat{\Gamma}(x_0,c_F,\alpha) \in \Gamma(x_0,c_F,\alpha)$ and apply Lemma 4.4. □

Lemma 4.11. *Let* $(x,k) \in \mathbb{A}\times\boldsymbol{C}\times[0\dots N-1]$, *then* $V_3(x,t_k) \geq V_1(x,t_k)$.

Proof. By Lemma 4.1, we have

$$V_1(x,t_k) = \inf_{x_i} \sum_{i=k}^{N-1} V_1^\mu(x_i,c_{i+1},a_{i+1}-a_i) + \tilde{\phi}(a_N). \tag{4.58}$$

and by Lemma 4.10 and the definition of V_3,

$$V_3(x,t_k) = \inf_{x_i} \sum_{i=k}^{N-1} V_1^\mu(x_i,c_{i+1},a_{i+1}-a_i) + \tilde{\phi}(a_N), \tag{4.59}$$

where the optimization is performed over the x_i such that there exist $(\alpha_i)_{i=k+1,\dots,N}$ to satisfy (4.56). QED. □

Remember the definition of γ in (4.54).

Lemma 4.12. *For any* $(x_0,c_F,u) \in (\mathbb{A}\times\boldsymbol{C})\times\boldsymbol{C}\times U_T$ *such that* u *is admissible for problem* $P_1^\mu(a_0,\Delta a(u),c_0,c_F)$, *there exists* α *such that*

$$dist(\Delta a(u),\hat{\Gamma}(x_0,c_F,\alpha)) \leq \gamma. \tag{4.60}$$

Proof. We know that for $\alpha \in \mathbb{R}_+$ big enough, $\delta \in \hat{\Gamma}(x_0, c_F, \alpha)$ implies $\delta \leq \Delta a(u)$. Take α_0 the infimum of those α. Then if there exists $\delta \in \hat{\Gamma}(x_0, c_F, \alpha_0)$ such that $\delta = \Delta a(u)$ we just take $\alpha = \alpha_0$ and we are done. Else by definition of α_0, for all $\alpha^- < \alpha$, $\Delta^+(x_0, c_F, \alpha^-) > \Delta a(u)$ and for all $\alpha^+ > \alpha$, $\Delta^-(x_0, c_F, \alpha^+) \leq \Delta a(u)$. This rewrites, for all $\delta \in \hat{\Gamma}(x_0, c_F, \alpha)$

$$\Delta^-(x_0, c_F, \alpha^+) - \delta \leq \Delta a(u) - \delta \leq \Delta^+(x_0, c_F, \alpha^-) - \delta \tag{4.61}$$

Taking the limit in α^- and α^+, we get

$$|\Delta a(u) - \delta| \leq \gamma \tag{4.62}$$

i.e.

$$dist(\Delta a(u), \hat{\Gamma}(x_0, c_F, \alpha)) \leq \gamma. \tag{4.63}$$

□

The next result contains one of the main ideas of the paper: the set of the x_i satisfying (4.56) is rich enough to approximate a trajectory with a precision of γ.

Theorem 4.13. *Let $x_0 \in \mathbb{A} \times \boldsymbol{C}$ and $u \in U$ admissible for $P_1(x_0, 0)$. Then there exist $(\alpha, \delta a) \in R_+^{2N}$ such that for all $k = 0, \ldots, N$:*

$$\delta a_k \in \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.64}$$

$$\sum_{i=0}^{k-1} \delta a_i \leq a(t_k) - a_0 \tag{4.65}$$

$$\sum_{i=0}^{k-1} \delta a_i \geq a(t_k) - a_0 - \gamma, \tag{4.66}$$

where $a_k = a(t_k)$ and $c_k = c(t_k)$.

Proof. First note that (4.65) and (4.66) are satisfied for $k = 0$. Now consider $k = 0, \ldots, N$ such that the three properties are satisfied until $(k-1)$. If there exist α such that

$$a(t_{k+1}) - a(t_k) \in \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.67}$$

then set $\delta a_k = a(t_{k+1}) - a(t_k)$ and $\alpha_k = \alpha$. and then the three properties are still trivially satisfied for k. Else, we apply Lemma 4.12 to justify the existence of an α_k such that

$$dist(a(t_{k+1}) - a(t_k), \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \leq \gamma. \tag{4.68}$$

then if

$$\sum_{j=0}^{k-1} \delta a_j + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \leq a(t_{k+1}) - a_0 \tag{4.69}$$

then we set $\delta a_k = \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$ Then constraints (4.64) and (4.65) are satisfied. We need to check that (4.66) is also satisfied. To see this, observe that

$$\sum_{j=0}^{k-1} \delta a_j + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.70}$$

$$\geq a(t_k) - a_0 - \gamma + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.71}$$

$$\geq a(t_{k+1}) - a_0 + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - (a(t_{k+1}) - a(t_k)) - \gamma \tag{4.72}$$

$$\geq a(t_{k+1}) - \gamma, \tag{4.73}$$

where we applied, the induction and the fact that $\Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \geq a(t_{k+1}) - a(t_k)$. Else we set $\delta a_k = \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$. We have

$$\sum_{j=0}^{k-1} \delta a_j + \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.74}$$

$$> a(t_{k+1}) - a_0 - \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.75}$$

$$> a(t_{k+1}) - a_0 - \gamma \tag{4.76}$$

and

$$\sum_{j=0}^{k-1} \delta a_j + \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.77}$$

$$\leq a(t_k) - a_0 + \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \tag{4.78}$$

$$\leq a(t_{k+1}) - a_0 \tag{4.79}$$

which concludes the induction. □

We point out that the proof of this result is constructive. We propose in the next result an a posteriori error estimate based on this construction. Since $a(t_k)$ is the age of the system piloted by the ϵ-optimal solution u and $a_0 + \sum_{i=1}^{k} \delta a_i$ is the age of the system piloted by the solution we are building with adaptative weights, relation (4.64) means that the system is always in better shape (i.e. younger) in the solution we are building, and at the same time, it cannot be more than γ better (reminder: γ is the maximal diameter of Γ).

Theorem 4.14 (Error Estimate)**.** *For any ϵ-optimal solution of $P_1(a_0, c_0, t_0)$, the construction of the previous theorem gives the estimate:*

$$V_3(a_0, c_0, t_0) - V_1(a_0, c_0, t_0) \leq \sum_{k=0}^{N-1} \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k) + \epsilon \tag{4.80}$$

Note that $(a(t_{k+1}) - a(t_k) - \delta a_k) = 0$ when there is no jump at α_k, which can be numerically checked. Thus the theorem give an a posteriori estimate of the error when optimizing with an adaptative weights approach. For instance, if the structure of the problem allows us to claim that there are no jumps, then we do not make any approximation error.

Proof. Take $\epsilon > 0$ and $u \in U$ an ϵ-optimal control for problem $P_1(c_0, a_0, t_0)$. We apply Theorem 4.13 to u.

We have

$$V_3(c_0, a_0) \leq \sum_{k=1}^{N} \{V_2^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - \alpha_k \delta a_k\} + \tilde{\phi}(a + \sum_{k=0}^{N-1} \delta a_k) \tag{4.81}$$

We deal with the first term of the RHS. According to Lemma 4.10, for all $k = 0 \dots N-1$

$$V_2^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - \alpha_k \delta a_k = V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, \delta a_k, c_k, c_{k+1}) \tag{4.82}$$

Therefore

$$V_2^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_{k-1}, c_k, \alpha_k) - \alpha_k \delta a_k - V_1^\mu(a(t_k), a(t_{k+1}) - a(t_k), c_k, c_{k+1}) \tag{4.83}$$

$$= V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, \delta a_k, c_k, c_{k+1}) - V_1^\mu(a(t_k), a(t_{k+1}) - a(t_k), c_k, c_{k+1}) \tag{4.84}$$

$$= V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, \delta a_k, c_k, c_{k+1}) - V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, a(t_{k+1}) - a(t_k), c_k, c_{k+1}) + \tag{4.85}$$

$$V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, a(t_{k+1}) - a(t_k), c_k, c_{k+1}) - V_1^\mu(a(t_k), a(t_{k+1}) - a(t_k), c_k, c_{k+1}) \tag{4.86}$$

The second difference is negative because V_1^μ is decreasing in its first variable and $a_0 + \sum_{j=0}^{k-1} \delta a_j \leq a(t_k)$ by construction. We thus concentrate on the first difference. If there is no jump, this quantity is zero. Otherwise, we need to compare (simplifying the notations)

$V_1^\mu(\delta a_k)$ and $V_1^\mu(a(t_{k+1}) - a(t_k)$. By definition of P_2^μ, we have

$$V_1^\mu(a(t_{k+1}) - a(t_k)) + \alpha_k(a(t_{k+1}) - a(t_k)) \geq V_2^\mu(\alpha_k). \tag{4.87}$$

Remember that by definition the pair $(\alpha_k, \delta a_k)$ satisfies $\delta a_k \in \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$. We can apply Lemma 4.10 to get

$$V_2^\mu(\alpha_k) = V_1^\mu(\delta a_k) + \alpha_k \delta a_k. \tag{4.88}$$

Therefore, combining relations (4.87) and (4.88) we get

$$V_1^\mu(\delta a_k) - V_1^\mu(a(t_{k+1}) - a(t_k)) \leq \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k) \tag{4.89}$$

Therefore if we denote by K_{jumps} the $k \in [0..N-1]$ such there is a jump in the construction:

$$V_3(a_0, c_0, t_0) - V_1(a_0, c_0, t_0) \leq \tag{4.90}$$

$$\sum_{k \in K_{jumps}} \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k) + \tilde{\phi}(a_0 + \sum_{k=0}^{N-1} \delta a_k) - \phi(a(t_N)) + \epsilon \tag{4.91}$$

$$\leq \sum_{k \in K_{jumps}} \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k) + \epsilon. \tag{4.92}$$

Since $a(t_N) \geq a_0 + \sum_{k=0}^{N-1} \delta a_k$ by construction, ϕ is monotone decreasing and u is admissible for $P_1(x_0, t_0)$. □

4.4.5 Complexity Analysis

We denote by N_α the number of elements in the discretization of α. We get the following offline computation burden for the adaptative weights algorithm if we neglect the age variations in the micro problem.

- Micro problem: $O(N_c N_u N_t T)$
- Parameters for micro: $O(N_\alpha N_a N_c)$
- Macro: $O(N_a N_c N_\alpha N)$
- Total bilevel: $O(N_a N_\alpha N) + O(N_\alpha N_a N_c^2 N_u N_t T)$

We proceed with a complexity analysis for the case where we have the constraint $c_{t_k} = c_0$ and we neglect the age variations in the micro problem:

- Micro problem: $O(N_c N_u N_t T)$
- Parameters for micro: $O(N_\alpha N_a)$

	Offline Computation burden
Approach	offline
BF	$N_a N_c N_u N_t T N$
Bilevel	$N_a^2 N_c^2 N + N_a^2 N_c^2 N_u N_t T$
AWA	$N_\alpha N_a N_c^2 N + N_\alpha N_a N_c^2 N_u N_t T$

Table 4.1 – The offline computation burdens for the general case

		Computation burden		Memory requirement
Approach		offline	online	
BF	(b)	$N_a N_c N_u N_t T N$	$N_a N_c N_u N_t T$	$N_a N$
Bilevel	(d)	$N_a^2 N + N_a^2 N_c N_u N_t T$	$N_a^2 N + N_a N_c N_u N_t T$	N_a^2
	(e)	"	$N_a N_c N_u N_t T$	$N_a N$
AWA		$N_a N_\alpha N + N_\alpha N_a N_c N_u N_t T$	$N_a N_\alpha N + N_c N_u N_t T$	$N_a N_\alpha$

Table 4.2 – The computation burdens for the case where c has to be the same at the beginning of each period

- Macro: $O(N_a N_\alpha N)$
- Total bilevel: $O(N_a N_\alpha N) + O(N_\alpha N_a N_c N_u N_t T)$

If N_a and N_α are of the same order, then the offline computation burdens for the adaptive weights and bilevel dynamic programming algorithms are of the same order. We then store an $N_a \times N_\alpha$ matrix. The online optimization complexity is proportional to $N_a N_\alpha N + N_c N_u N_t T$.

The complexity analysis is summarized in Tables 4.1 and 4.2.

4.5 Discussion

4.5.1 Asymptotic Analysis

We propose in this subsection an asymptotic error estimate. The derivation of the estimate relies mostly on Assumption 1, which by the way we did not use in the previous sections. We now express formally the approximation of V_2^μ envisioned in the complexity analysis:

$$\tilde{P}_2^\mu(a_0, c_0, c_F, \alpha) \begin{cases} \tilde{V}_2^\mu(a_0, c_0, c_F, \alpha) = \inf_{u \in U} \int_0^T [\ell(u(t), t) + \alpha F_a(a_0, c(t), u(t), t)] \mathrm{d}t \\ \dot{c}(t) = F_c(a_0, c(t), u(t), t) \\ (a(0), c(0)) = (a_0, c_0) \\ \varphi(c(t), u(t), t) \in A \\ c(T) \geq c_F. \end{cases} \tag{4.93}$$

This approximation consists in neglecting the evolution of a in the micro optimal control problem. We then set $\tilde{\Delta}a(u) = \int_0^T F_a(a_0, c(t), u(t), t)\mathrm{d}t$ and define $\tilde{\Gamma}$ for $\tilde{P}_2^\mu$ the same way we defined $\hat{\Gamma}$ for P_2^μ. Last we define $\tilde{V}_3$ by replacing $\hat{\Gamma}$ and V_2^μ by $\tilde{\Gamma}$ and $\tilde{V}_2^\mu$ in the definition of V_3 and $\tilde{U}_x$ as the set of controls $u \in U_T = \{u \in L^\infty(0,T) \text{ s.t. } \forall t \in [0,T], u(t) \in \bar{U}\}$ such that $\varphi(\tilde{c}_{x_0,0}^{u,t}, u(t), t) \in A$ for any $t \in [0,T]$, where $\tilde{c}$ is the flow corresponding to the dynamics (4.93).

First we estimate with Gronwall's lemma the error made on the trajectories.

Lemma 4.15. *There exists a constant K such that for any $(a_0, c_0, u, t) \in \mathbb{A} \times \boldsymbol{C} \times U_T \times [0,T]$,*

$$|a_0 - a_{x_0,0}^{u,t}| \leq K/N \text{ and } |c_{x_0,0}^{u,t} - \tilde{c}_{x_0,0}^{u,t}| \leq K/N, \tag{4.94}$$

and

$$|\Delta a(u) - \tilde{\Delta}a(u)| \leq K/N^2. \tag{4.95}$$

Proof. We get the first inequality using $F_a \leq L/N$, the second inequality by combining the L-Lipschitzianity of F_c, the first inequality and Gronwall lemma. We get (4.95) combining the L/N-Lipschitziannity of F_a with (4.94). □

We continue with the error estimate.

Theorem 4.16. *Let $x_0 = (a_0, c_0) \in \mathbb{A} \times \boldsymbol{C}$. Let (α_i, a_i, c_i) be some minimizers of $V_3(x_0, 0)$. Let $u_i \in U_{a_i,c_i}$ be a minimizer of $P_2^\mu(a_i, c_i, c_{i+1}, \alpha_i)$ such that $u_i \in \tilde{U}_{a_i,c_i}$. Let $\psi \in \mathbb{R} \to \mathbb{R}$ and $C \geq 0$ be such that*

- $a_{i+1} - a_i \in \tilde{\Gamma}(x_i, c_{i+1}, \psi(\alpha_i))$
- $\mid \psi(\alpha_i) - \alpha_i \mid \leq C$
- $\alpha_i \leq NC$

Then

$$\frac{\tilde{V}_3(x_0,0) - V_3(x_0,0)}{N} \leq \frac{3CK}{N} \tag{4.96}$$

Proof. Set $\delta a_i = a_{i+1} - a_i$. By definition of $\tilde{V}_3$, we have

$$\tilde{V}_3(x_0, 0) - V_3(x_0, 0) \leq \tag{4.97}$$

$$\sum_{i=0}^{N-1} \tilde{V}_2^{\mu}(x_i, c_{i+1}, \psi(\alpha_i)) - V_2^{\mu}(x_i, c_{i+1}, \alpha_i) + (\alpha_i - \psi(\alpha_i))\delta a_i + \phi(a_n) - \phi(a_n) \tag{4.98}$$

$$\leq \sum_{i=0}^{N-1} \inf_{u \in \tilde{U}_{x_i}} [\mathcal{L}(u) + \psi(\alpha_i)\tilde{\Delta} a(u)] - \inf_{u \in U_{x_i}} [\mathcal{L}(u) + \alpha_i \Delta a(u)] + (\alpha_i - \psi(\alpha_i))\delta a_i \tag{4.99}$$

$$\leq \sum_{i=0}^{N-1} \inf_{u \in \tilde{U}_{x_i}} [\mathcal{L}(u) + \psi(\alpha_i)\tilde{\Delta} a(u)] - \inf_{u \in \tilde{U}_{x_i}} [\mathcal{L}(u) + \alpha_i \Delta a(u)] + CK/N \tag{4.100}$$

$$\leq \sum_{i=0}^{N-1} \sup_{u \in \tilde{U}_{x_i}} [\psi(\alpha_i)\tilde{\Delta} a(u) - \alpha_i \Delta a(u)] + CK/N \tag{4.101}$$

$$\leq \sum_{i=0}^{N-1} \frac{CK}{N} + \frac{\alpha_i K}{N^2} + \frac{CK}{N} \leq 3CK \tag{4.102}$$

□

The reverse result can be proved similarly.

To give an intuition of the estimate of α, observe that $\Gamma(x_0, c_F, N\alpha)$ is equal to a constant divided by N for any α.

4.5.2 Extensions

In this subsection we propose some possible extensions for the adaptative weights algorithm.

Obstacle

In many applications, it is possible to buy off the shelf spare parts to replace worn components. Hence we may want to introduce the possibility to buy a replacement in the optimization problem. This could be done with an impulse control: for a fixed price p, we should be able to reset the age a to zero. We would get the following dynamic programming principle:

$$V_3(a, c, t_k) = \min\{\inf_{c_f, \alpha, \delta a} V_2^{\mu}(a, c, c_f, \alpha) - \alpha\delta a + V_3(a + \delta a, c_f, t_{k+1}), V_3(0 + \delta a, c_f, t_{k+1}) + p\} \tag{4.103}$$

Periodicity

We can include seasonality by having different kinds of periods. For instance, we could model winter and summer days. In this case, one need to perform an offline pre-processing for each kind of day.

Short Term and Long Term Randomness

With Markovian dynamics and final constraints on the average age increment, the same arguments should apply. Moreover, we could add an integer state with Markovian dynamics to model the type of 'days'.

Infinite Horizon

If we add a discount rate, the arguments should apply for infinite horizon. Then one needs to replace the macro dynamic programing algorithm by either a policy iteration algorithm or a value iteration algorithm.

4.5.3 Algorithm

We propose a bi-level approach that consists in an offline and an online part. For readability we assume that c should have the same value at the end of each period $\tilde{c}$. We already defined $\Delta a(u)$ and $\mathcal{L}(u)$. The inputs of the algorithm are the discretization grids of the age and the parameters, namely I_a and I_α. We denote by k_0 the current period number and by a_0 the current age. We use the notation $\tilde{\phi}(a) = \phi(a)$ if $a < a_{max}$, $+\infty$ else. For a table T indexed by I_a $\mathcal{F}T$ is an interpolation of T over the grid. The output of the offline algorithm is the pair of tables $(\Delta_{\alpha,a_0}, \mathcal{L}_{\alpha,a_0})$. The output of the online algorithm is a control $u^* \in U_T$.

Algorithm 1 OFFLINE Algorithm

Data: I_a, I_α
Result: Δ_{α,a_0} $\mathcal{L}_{\alpha,a_0}$
for $\alpha \in I_\alpha$ **do**
  for $a_0 \in I_a$ **do**
    Solve $P_2^\mu(a_0, \tilde{c}, \tilde{c}, \alpha)$ Compute an optimal control u for $P_2^\mu(a_0, \tilde{c}, \tilde{c}, \alpha)$ $\Delta_{\alpha,a_0} \leftarrow \Delta a(u)$ and $\mathcal{L}_{\alpha,a_0} \leftarrow \mathcal{L}(u)$
  end
end

Algorithm 2 ONLINE Algorithm

Data: k_0, a_0, I_a, I_α, Δ_{α,a_0}, $\mathcal{L}_{\alpha,a_0}$

Result: u^*

Initialize $\tilde{V}_{:,:} \in I_a \times [k_0 + 1 \ldots N]$ $\tilde{V}_{:,:} \leftarrow +\infty$ $\tilde{V}_{:,N} \leftarrow \tilde{\phi}(:)$

for $k \leftarrow N-1$ **to** k_0+1 **do**

  for $a \in I_a$ **do**

    for $\alpha \in I_\alpha$ **do**

      $\tilde{V}_{a,k} \leftarrow \min\{\mathcal{L}_{\alpha,a} + \mathcal{F}\tilde{V}(a + \Delta_{\alpha,a}, k+1); \tilde{V}_{a,k}\}$

    end

  end

end

$\alpha^* \leftarrow \mathrm{argmin}_{I_\alpha}\{\mathcal{L}_{\alpha,a} - \alpha\Delta_{\alpha,a_0} + \tilde{V}(a_0 + \Delta_{\alpha,a_0}, k_0+1)\}$ Compute an optimal control u^* for $P_2^\mu(a_0, \tilde{c}, \tilde{c}, \alpha^*)$ Return u^*

4.6 Simulation and Implementation on a Microgrid Model

4.6.1 Problem Presentation

A microgrid is an electric system that includes electricity generation units (dispatchable and non dispatchable) and a battery to store energy for later use. Although the battery price is a non negligible part of the total infrastructure cost, the battery aging is rarely taken into account in the control of the grid: this is the source of sub-optimality we propose to deal with. The profusion and complexity of battery aging models is a reason why, to the extent of our knowledge, no generic optimization tools have been proposed yet. It is nonetheless natural to model the aging through a quantity representing the age of the battery that would increase as the battery is used. The scope of our framework is the models for which the battery age dynamics is a controlled first order ordinary differential equation. We use a severity factor model for the aging and solve the simplified optimal control formulation (introduced in [49] and [48] by dynamic programming.

Here is a brief description of the microgrid we consider. The electricity is produced by some non dispatchable units (solar panels) and a dispatchable unit (a diesel generator or the network for instance). At each instant, there is an instantaneous demand (non controllable) for electricity. If there is a production surplus, one can store this surplus in the battery. If there is not enough electricity produced at this instant, and there is some energy left in the battery, one can use the battery to fill the gap or increase the production from the dispatchable unit. Note that the battery is not a perfect storage: if one unit of energy is stored in an empty battery, the total amount of energy we can get from the battery is strictly lower than 1. Of course, there is a cost associated with the production of electricity from the dispatchable unit.

A more detailed and technical description of the system, with the underlying equations, is proposed in §4.6.3. As already discussed, the battery state can be described by two variables: the state of charge c and the age a. The state of charge c is a normalized quantity that is the ratio of energy stored in the battery over the maximum quantity the battery can store. It is zero when the battery is empty, and 1 when the battery is full.

The age a is also a normalized quantity in $[0, 1]$. We set $a = 0$ for a brand new battery and $a = 1$ for a dead one. Obviously, we need to precise the dynamics of a and c so that the model makes sense from a physics perspective. Such dynamics can be found in the literature (see §4.6.2). Very often in the literature instead of the notion of age we find the concept of state of health $h = 1 - a$. We prefer the notion of age in this work, to stay coherent with the previous sections.

In order to make the document self contained and the results reproducible we use analytical inputs for the solar power production and for the power consumption (load)

4.6.2 Battery Aging Model

As noted by Koller et al. in [60], battery based solutions for energy storage present the advantage of being deployable without any consideration of the geographic factors and within short schedule thanks to their modularity. In [83] the authors propose three ways to model battery aging: a physico-chemical model, a weighted Amp-hour (Ah) throughput model (or charge counting model) and an event oriented model. In [22] Borhan et al. propose a model predictive control approach where the aging is penalized in the criterion. They implement a weighted Ah throughput model. In [60] Koller et al. propose a discrete time, model predictive control where the aging factor is the Depth of Discharge (DoD), which is modeled with piece-wise affine dynamics.

In [46] Haessig et al. propose a simulation that includes an aging model in order to perform a cost analysis, yet the optimization of the operations is not in the scope of this work. In his PhD thesis [44] Haessig describes a battery aging model (among others) based on the total amount of energy exchanged during the lifetime of the battery. In [81] Riffonneau et al. use a a discrete time dynamic programming approach to solve an optimal power flow problem. The battery aging is proportional to the discharge of the battery. It linearly decreases the capacity of the battery.

In [75], Palma et al. integrate the battery aging in a rolling horizon strategy model. The aging is taken into account in the model using a working zones approach as proposed in [43] and the penalty parameter is the investment cost of the battery.

As argued in the final comments of [44] instead of using a penalization approach, one could impose a maximal aging constraint. The supporting argument is that the appropriateness of an aging profile depends on the time horizon over which the battery is supposed to be in operation. Then in [45] Haessig et al. propose to implement an aging constraint by introducing the notion of exchangeable energy for exchanged energy counting aging model.

To our knowledge there are basically three approaches in the literature to take battery aging into account in an optimization model:

- Some constraints on the control and state variables (for instance to avoid extreme State of Charge values). This requires deciding which constraints to implement. The aging is not directly taken into account in the optimization process. The constraints can be too or not enough conservative.

- A penalization of the aging. This requires choosing an aging model and a penalization parameter.

- An aging constraint. This requires choosing an aging model and the aging level. In addition, one may need a heuristic to implement this constraint if a direct numerical optimization is too burdensome.

Observe that ideally, we should perform an optimization over the whole remaining existence of the microgrid and take into account the impact of the aging on the battery performance. If the battery scheduled lifetime is shorter than the microgrid one, then the optimal aging profile should take into account the possibilities and the conditions (price, etc.) to buy a new battery. Then it appears that in this very idealistic viewpoint, the optimal control of the microgrid would take into account the aging without requiring the implementation of any penalization or constraint. The aging penalization and the aging constraint are in fact rule of thumbs to incorporate those long term considerations.

Nonetheless, it is hard to conciliate long term optimization and the modeling of the aging related performances variations. Moreover, the numerical resolution of the optimal control formulation problem needs to be fast enough if one envisions a real *online* implementation.

Note that the adaptative weights approach presented in this paper allows for a long term *offline* optimization. The output of the offline optimization is a closed loop optimal penalization parameter which can be then used as input for the *online* (and short term) optimal control problem. Since the age variation within a single day has a negligible impact on the performance, we can neglect those variations for the numerical resolution of the *online* optimal control problem. By doing so, the *online* problem approximation is one dimensional an can be solved efficiently with dynamic programming.

Note that this approach should work as long as we have a continuous time model for the aging (i.e. an ordinary differential equation). In the following we will apply our framework to a severity factor model based on the state of charge of the battery. The whole quantitative formulation is presented in §4.6.3.

4.6.3 The Optimal Control Formulation

We implement the adaptative weights framework on a simplified version of the continuous time optimal control formulation we introduced in [49] and extended to a stochastic setting in [48]. The unit of time is the hour and $T = 24$ corresponds to a day. The long term optimal control problem writes

$$P_1(a_0, c_0, t_0) \begin{cases} V_1(a_0, c_0, t_0) = \inf_{u \in U} \int_{t_0}^{T_{tot}} \ell(u(t))\mathrm{d}t \\ (a(t_0), c(t_0)) = (a_0, c_0) \\ a(T_{tot}) \leq a_{max} \\ c(t) \in [0.1, 1] \\ (\dot{a}(t), \dot{c}(t)) = (F_a(a(t), c(t), u(t), t), F_c(a(t), c(t), u(t), t)), \end{cases} \tag{4.104}$$

The control u corresponds to the power produced by the dispatchable unit (a diesel generator). If the battery is full and renewable production is greater than demand, we can disconnect the battery. To simplify the model, we implement this by allowing u to be negative. Let $\beta > 0$. The integral cost

$$\ell(u) = \beta(u^+)^2, \tag{4.105}$$

is a quadratic (and so convex) function associated with the generator consumption of fuel. The value of β as well as the other model parameters are detailed in Table 4.3. Observe that only the product of U_{bat} and Ah_{bat} matters, so to decrease the number of parameters, we only indicate their product. If we denote by $P_s(t)$ the power produced by the solar panels and by $P_L(t)$ the load then the state of charge dynamics is

$$F_c(c, u, t) = \frac{\rho_i(a)P_i(a, c, u, t) - P_o(a, c, u, t)/\rho_o}{C} \tag{4.106}$$

with $P_i(a, c, u, t) = (-u - P_s(t) + P_l(t))^+$ being the power that gets into the battery and $P_o(a, c, u, t) = (-u - P_s(t) + P_l(t))^-$ being the power that gets out of the battery. We make the choice for simplicity purpose to model the aging impact on the performances by decreasing the efficiency ratio ρ_i:

$$\rho_i(a) = (1 - a)\rho \tag{4.107}$$

Where ρ is the initial coefficient for $a = 0$. We denote by C the capacity of the battery. In order to make our numerical experiment reproducible, we take T periodic functions with analytic expression for the data input. The functions were chosen to be realistic enough. For $t \in [0, 24]$

$$P_S(t) = \max(0, 13 - 0.3(4t - 48)^2) \tag{4.108}$$

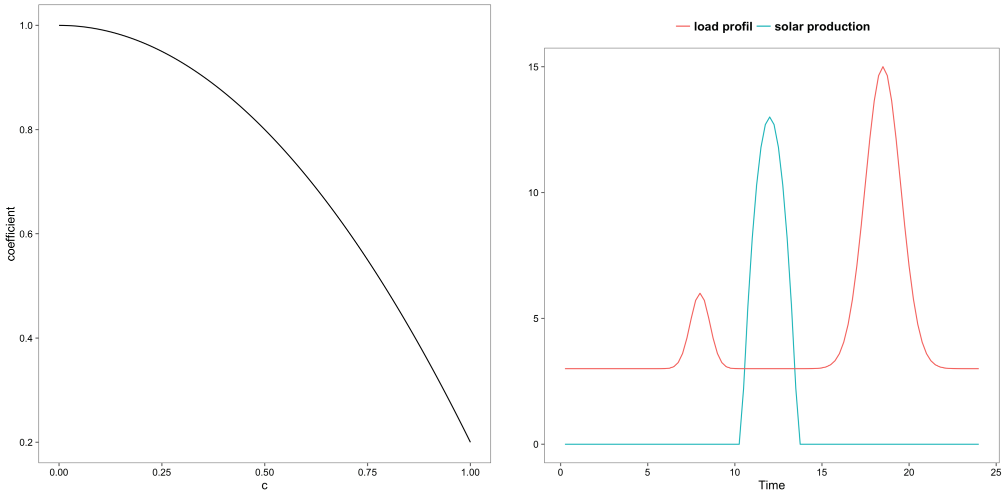


Figure 4.2 – ON THE LEFT: The aging severity factor as a function of the charge. The severity factor is greater when the battery is depleted. This function was originally obtained by interpolation of a piece-wise constant severity factors model. Note that in the model the battery aging happens only when energy is taken *from* the battery.
ON THE RIGHT: solar production and load during the day. Observe that the solar production only happens during mid-day, whereas the load profile has a constant component and two peaks, one in the morning and one in the evening.
For both plots, we made the choice to use analytic expressions to produce synthetic data.

Constant	Interpretation	Value
β	running cost coef.	0.5
ρ_i	efficiency factor (in)	0.95
ρ_o	efficiency factor (out))	0.95
$U_{bat}Ah_{bat}$	*	12.5

Table 4.3 – Model Constants

and

$$P_L(t) = 3 + 3e^{-0.1(4t-32)^2} + 12e^{-0.03(4t-74)^2}. \tag{4.109}$$

The aging dynamics corresponds to a severity factor model

$$F_a(a(t), c(t), u(t), t) = \eta(c)\frac{P_o(a, c, u, t)}{U_{bat}Ah_{bat}} \tag{4.110}$$

where

$$\eta(c) = \frac{(-4c^2 + 5)}{5} \tag{4.111}$$

is the severity factor (see Figure 4.2) and U_{bat}, Ah_{bat} are parameters that depend on the battery (see4.3). We set $a_- = 0$ and $a_+ = 1000$. For the numerical experiment, we will set $a_{max} = 500$ and $N = 600$.

Parameter	N_a	N_c	N_u	N_t
Value	100	100	100	4

Table 4.4 – Discretization parameters

algorithm	micro (total)	macro	total AWA	bruteforce
Comp. time	12.8 min	0.2 sec	12.8 min	> 10 hours

Table 4.5 – The computing times

4.6.4 Implementation

Periodicity

We impose that at the end of each day, the charge c should be equal to $\bar{c}$, which is a parameter decided upfront. The origin of this additional constraint is operational. Indeed, in a real setting, the optimization is performed regularly on a 24-hours sliding horizon window. Without any final constraint, the optimization program will tend to deplete the battery at the end of its horizon (*end of the world* effect). What is often done to deal with the undesirable effect is to impose that the battery at the end of the time horizon should be as charged as at the beginning. This is more or less what we implement here. In dynamic programming, we cannot in general impose hard equality constraints such as $c_T = \bar{c}$ when doing numerics. So we implement the periodicity of the daily final state of charge using a penalty function (in the running cost for the long term bruteforce problem, in the final constraints for the micro problem):

$$\Psi(c) = \begin{cases} (c - \bar{c})M_1 & \text{if } c \geq \bar{c} \\ M_2 & \text{else} \end{cases}. \tag{4.112}$$

Where M_1 and M_2 are two penalty parameters.

Discretization and Numerical Resolution

We display in Figure 4.4 the discretization parameters. We take $N = 600$ and $T = 24$. We take α between 0 and 500 with a discretization of 10 points, to which we add a $\alpha = +\infty$ point. We use a discretization of 20 points for the adaptive weights algorithm.

We use the optimal control toolbox BocopHJB (see [18] and [19]) to solve the optimal control problems. The macro algorithm is coded in the R scripting language. We performed the computation on laptop running OSX 10 with 1.3 GHz and 4 logical cores. The computing times are displayed in Table 4.5.

4.6.5 Results for the Micro Problem

Before commenting the results of the numerical experiment for the adaptive weights dynamic programming algorithm (that we will refer to as AWA), it is worth having a closer look at the micro problem (one day time horizon). We display in Figure 4.3 three simulations for (α, a) equal to $(0,0)$ (solid line), $(0,400)$ (dot and dash) and $(250,0)$ (dot). We see that the increase in age or α is associated with a decrease in the total aging within the day. This is done by diminishing the use of the battery: the maximal value of c is greater for the red curve. The explanations are different for the case $a = 400$ and $\alpha = 250$. For the first one, because the battery efficiency is poor, the quantity that gets effectively stored in the battery is low, so that there is not much to take from the battery during the peaks. The diesel needs to compensate the battery age. For the second one, the battery is efficient, but its use is penalized, so the diesel generator is used during the load peak to decrease the quantity of energy taken from the battery. Observe that, unlike what is seen in the two other cases, the control is flat for a new battery with no penalty. The aging occurs during the two load peaks for all profiles, when the battery is discharging.

We display $\Delta a(\alpha)$ for $\alpha = 0$ in Figure 4.4 (solid curve). Observe that this picture is qualitatively similar to the sketch in Figure 4.1. Yet it is likely that some *jumps* are the result of the discretization.

We display in Figure 4.4 and 4.5 $\Delta a(\alpha)$ and $\mathcal{L}(\alpha)$ with respect to α for two values of a and make the following observations:

- monotonicity of $\Delta a(\alpha)$: we observe that $\Delta a(\alpha)$ is monotone in a.
- regularity of $\Delta a(\alpha)$: There seem to be smooth and non smooth ranges for α. It is probable that some of the jumps are due to the discretization choice.
- as expected $\mathcal{L}(\alpha)$ and $\Delta a(\alpha)$ are respectively non-decreasing and non-increasing with respect to α for a fixed.

Remark on the periodicity of c We chose to impose a periodicity condition on the charge c for simplicity (in particular, the results are easier to represent) and because it makes sense from an operational perspective (see for instance [75]).

We compare in Figure 4.6 the optimal trajectories (computed by dynamic programming) with and without this periodicity condition.

We now proceed with the analysis of the macro part of the adaptative weights algorithm.

4.6.6 Results for the Macro Dynamic Programming Phase of AWA (Adaptative Weights Algorithm)

We display in Figure 4.7 two trajectories corresponding to two different initial ages. We observe that as long as the age is far from a_{max}, the lines *look* smooth. We display in

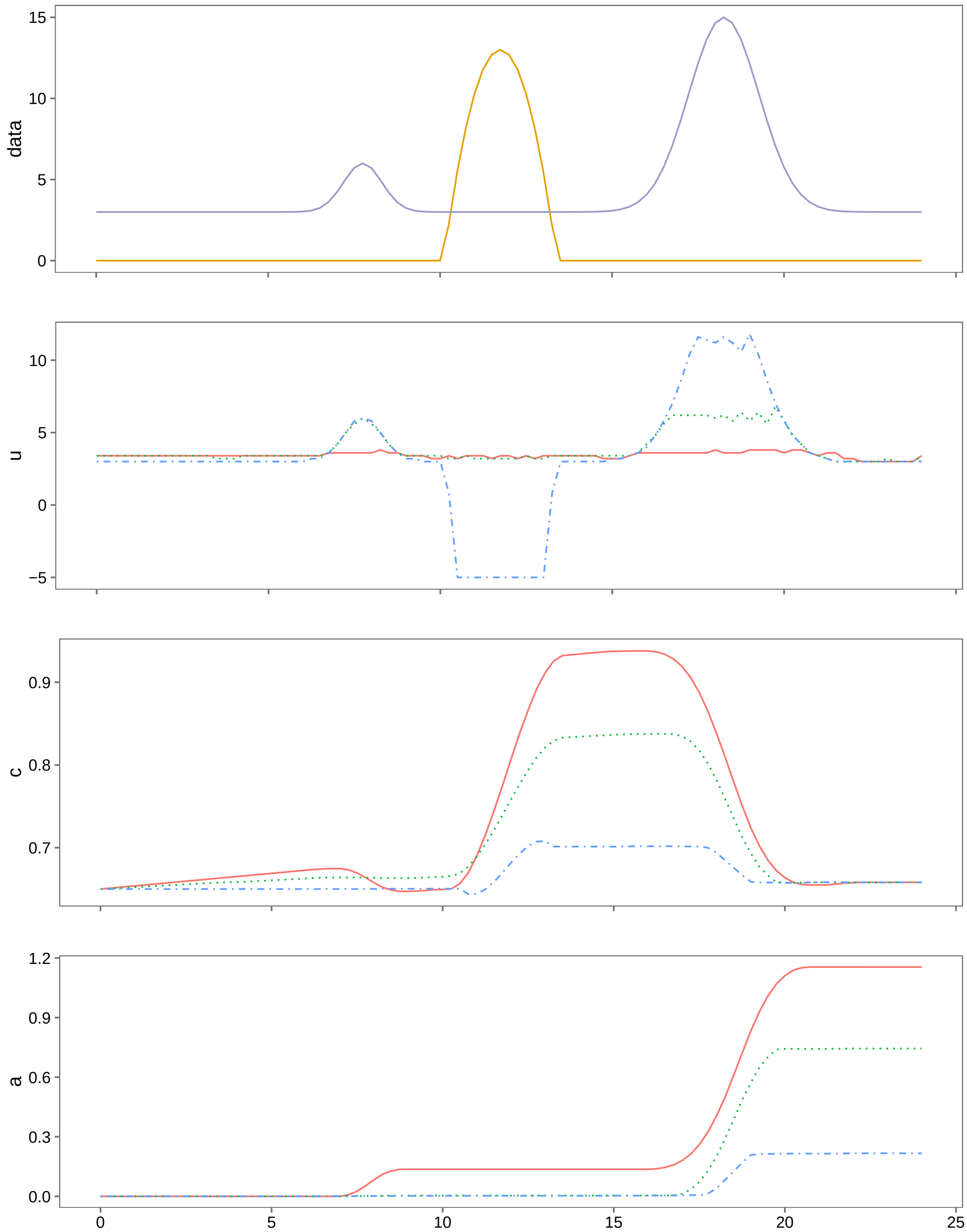


Figure 4.3 – Three simulations for (α, a) equal to $(0, 0)$ (solid), $(0, 400)$ (dot and dash) and $(250, 0)$ (dot). The bell shaped curve correspond to the solar production P_s and the other one to the load P_L.

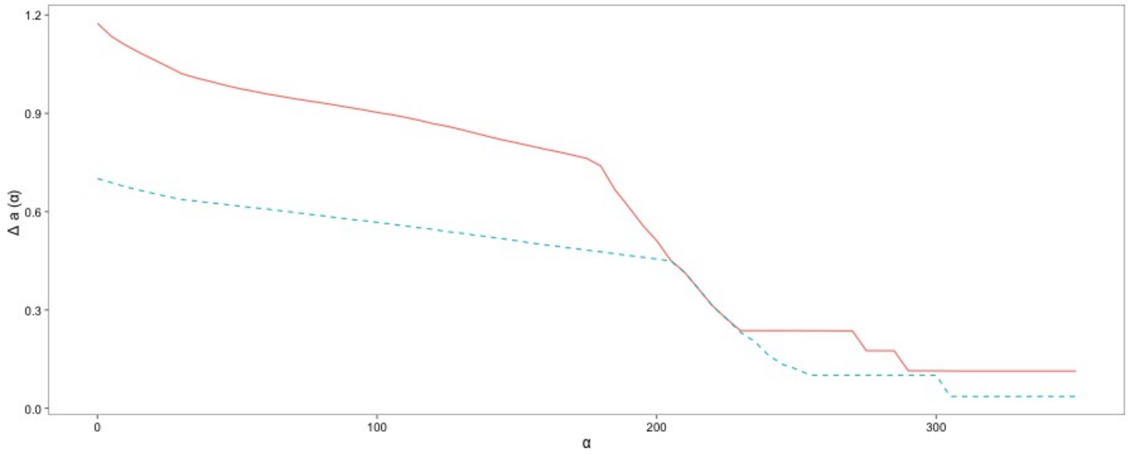


Figure 4.4 – $\Delta a(\alpha)$ for $a = 0$ (solid line) and $a = 450$ (dashed line)

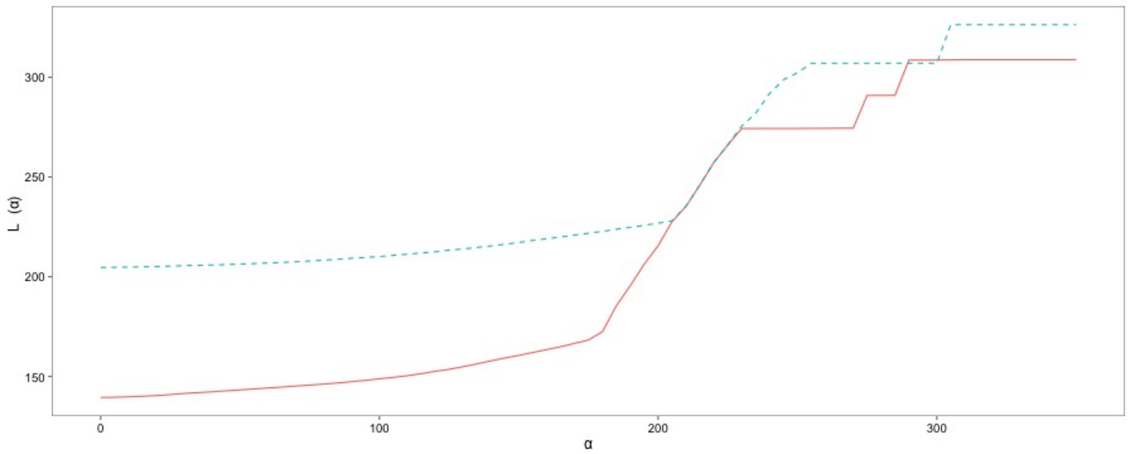


Figure 4.5 – $\mathcal{L}(\alpha)$ for $a = 0$ (solid line) and $a = 450$ (dashed line)

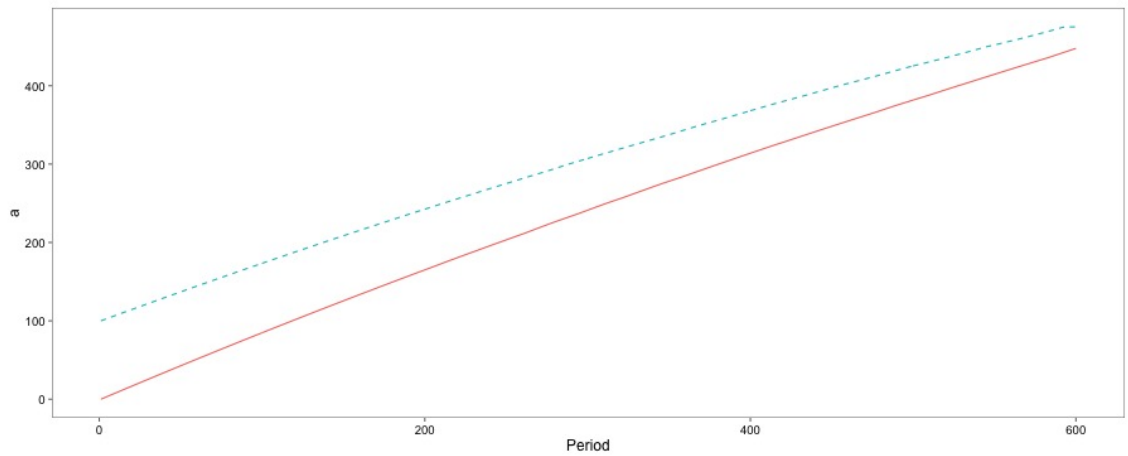


Figure 4.6 – The age a trajectories for $a(0) = 0$ and $a(0) = 100$. The solid lines correspond to the *periodic case* and the dotted line to the unconstrained case. The periodicity constraint increases the overall aging.

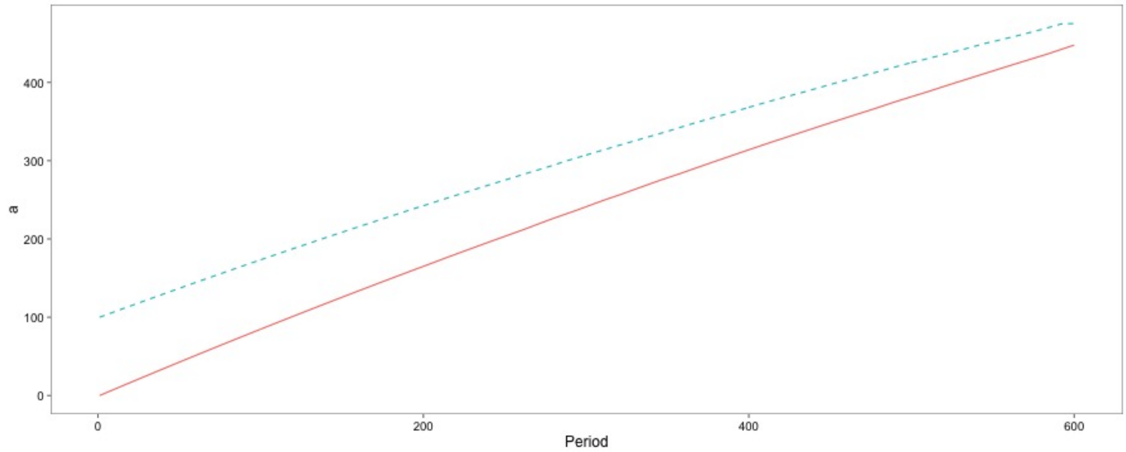


Figure 4.7 – Two trajectories computed with AWA for $a(0) \in \{0, 100\}$

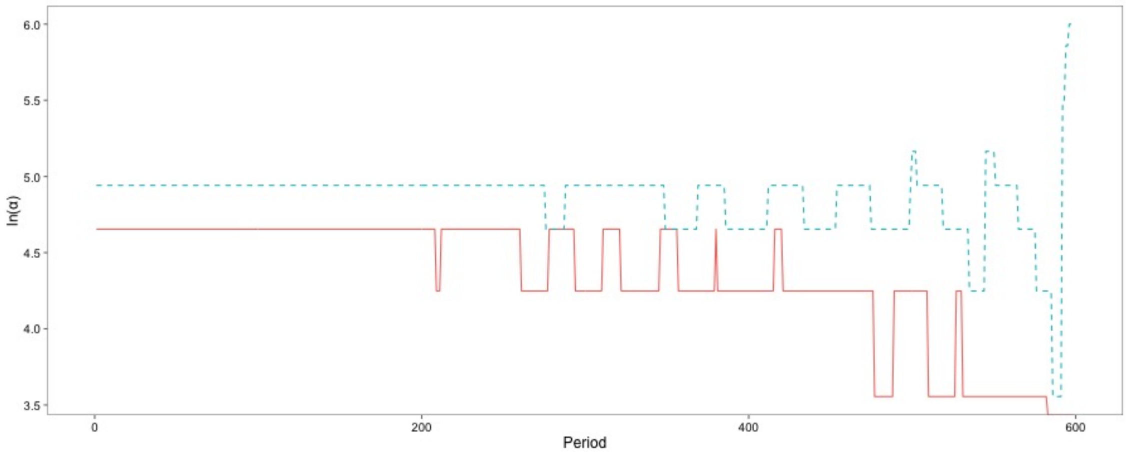


Figure 4.8 – The daily weights along the trajectories for $a(0) \in \{0, 100\}$

Figures 4.8 and 4.9 the daily weights and the age increments along those same trajectories. The oscillations of the weights have two possible explanations: first the discretization of the set to which α belongs, second, a jump in Δa, which the oscillations smooth out on average. As explained in the discretization section, on Figure 4.8 the maximal value of the dotted curve ($a(0) = 100$) corresponds to $\alpha = +\infty$ as we have added such point in the discretization of α to freeze the aging.

4.6.7 Comparison with the Bruteforce Results

We display in Figures 4.10 and 4.11 the envelopes of c and u for $a(0) = 0$ obtained with the bruteforce dynamic programming algorithm. We see that the battery utilization rate decreases as time goes on, while the diesel generator's increases. We observe that the state of charge c is always slightly above its expected terminal value during the first month, which is possible since the constraint is implemented through a piecewise linear penalt, for

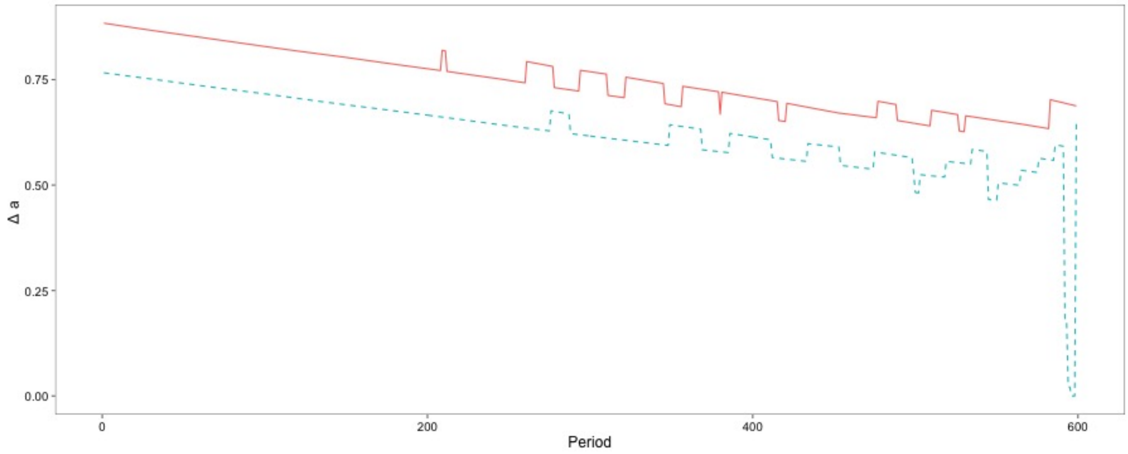


Figure 4.9 – The daily age increments along the trajectories for $a(0) \in \{0, 100\}$

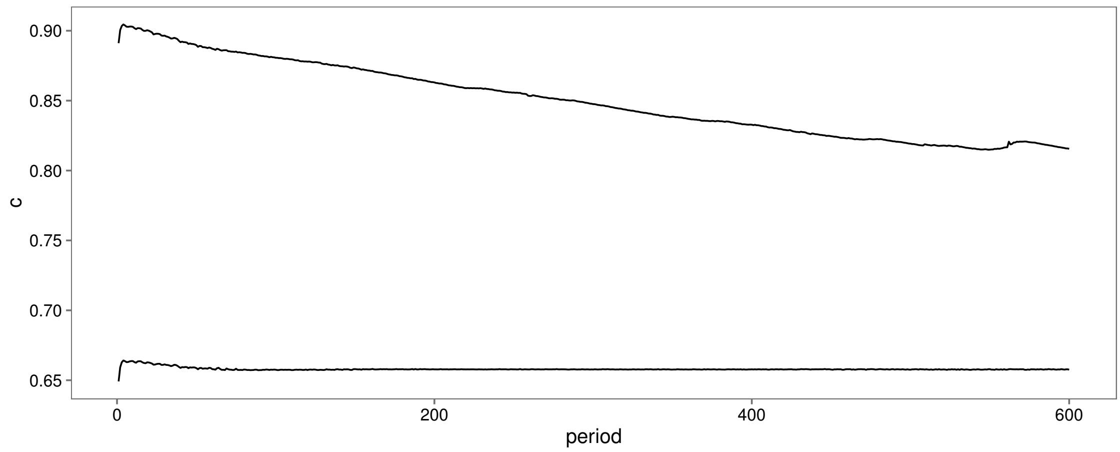


Figure 4.10 – Envelope of c obtained by dynamic programming for the long term problem

numerical reasons.

Last but not least we display on the same plot in Figure 4.12, two pairs of trajectories computed with AWA and a bruteforce dynamic programming approach for two initial ages. The corresponding values are displayed in Table 4.6.

start. age/algo.	AWA	BF	(AWA-BF)/BF
$a(0) = 0$	106544	105213	0.012
$a(0) = 100$	116901	111279	0.050

Table 4.6 – Estimates of the value function

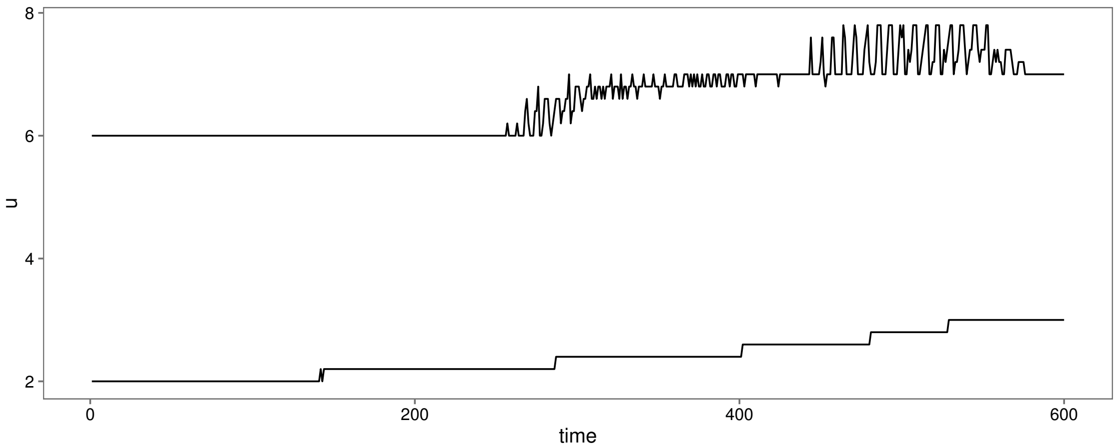


Figure 4.11 – Envelope of u obtained by dynamic programming for the long term problem

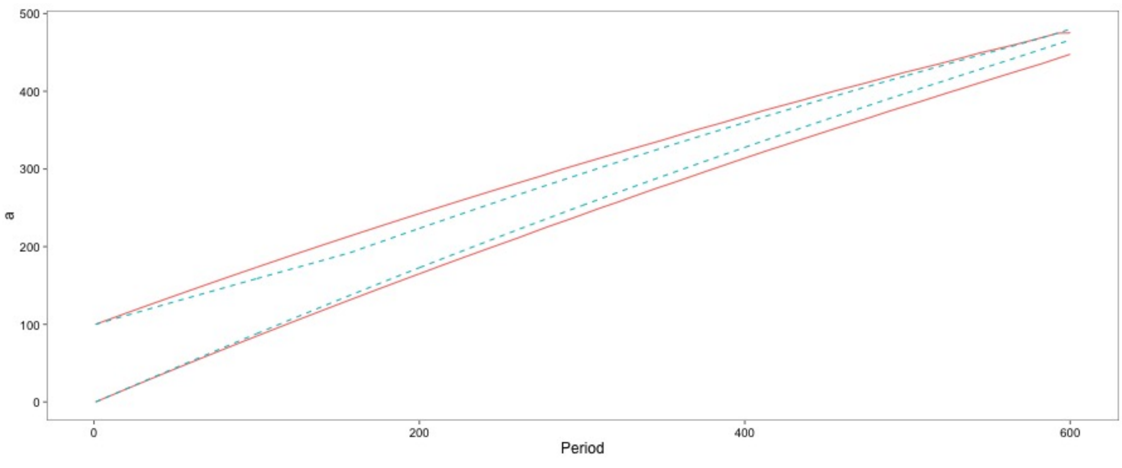


Figure 4.12 – The age profile computed with AWA (solid line) and bruteforce dynamic programming (dotted line) for two initial age values

4.7 Conclusion

We have introduced the adaptive weight dynamic programming algorithm (AWA), which is a decomposition technique for problems with periodic data. We tested this algorithm on a toy micro grid problem to integrate the battery aging within the decision process. The trajectories and the value functions obtained with AWA are close to those obtained with a bruteforce approach, and the computing times are way smaller.

Part II

Electricity markets

Chapter 5

Mechanism Design and Auctions for Electricity Networks

This joint work with Alejandro Jofré was accepted for publication in the Contributed Volume *Generalized Nash Equilibrium Problems, Bilevel Programming and MPEC* (Springer).

Contents

In this chapter, we present some key aspects of wholesale electricity markets modeling and more specifically focus our attention on auctions and mechanism design. Some of the results stemming from these models are the computation of an optimal allocation for the Independent System Operator, the study of equilibria (existence and uniqueness in particular) and the design of mechanisms to increase the social surplus. More generally, this field of research provides clues to discuss how wholesale electricity market should be regulated. We begin with a general introduction and then present some results we obtained recently. We also briefly discuss some ongoing related research. As an illustrative example, a section is devoted to the computation of the Independent System Operator response function for a symmetric binodal setting with piece-wise linear production cost functions.

5.1 Introduction

Economists, engineers and mathematicians have given a lot of attention to electricity markets since the beginning of the liberalization era in the 1980s. We present recent results and ongoing research about wholesale electricity markets and, in particular, the optimal design of such market. The field of market design studies the effects of market rules on economic functioning such as oligopoly behavior, vertical integration, market power, pricing, externalities and so on. The number of recent Nobel Prize laureates with contributions in this field demonstrates its impact on economic thinking. In this chapter we focus on recent works [36], [35], [34] and [38] as well as on some ongoing research by the authors. In the following sections, we assume that we are in a mandatory pool market, i.e. the agents will satisfy their engagements.

Diversity in electrical markets models can be explained by market specificity. This specificity stems from its economics, industrial and geographical setting, its dependency on the regulatory environment, time scales, and the complex physical properties of electrical networks (Kirchhoff laws, for instance) as well as the entities that compose it - producers, consumers, Independent System Operator (ISO) and networks.

Key modeling decisions concern the agents preferences, the uncertainties on the energy sources and demands, information representation, production capacities, and the physics of the system. In particular one has to specify the structure used to represent the bidding strategies of the producers. Since the physics of an electrical network are a difficult problem too, it is usually simplified.

There are (also) classic questions which accompany any modeling attempt, such as the mathematical well posedness of the problem, and the existence, uniqueness and tractability of the equilibria as wel as their properties. One might also ask about the existence of efficient algorithms for calculating those equilibria. We point out that models for wholesale electricity markets are often general enough to be relevant for other economic settings.

In our setting, the production allocation plan is the result of an auction. The producers communicate their selling prices to a central agent, and then the central agent minimizes the total cost while satisfying the demand. In our model we most of the time take into account the geography of the network (i.e. production and consumption are not co-localized at one point) as well as the losses due to the electricity transportation. Figure 5.1 presents a simple example of a network with four nodes.

We first present the general setting of the model in Section 2. Section 3 is a short review of some recent relevant work in the field. We give a quantitative formulation of the problem in Section 4. We will discuss the main results in Section 5. In Section 6 we develop the example of a two producers setting with piece-wise linear cost functions. We conclude in Section 7.

5.2 Setting

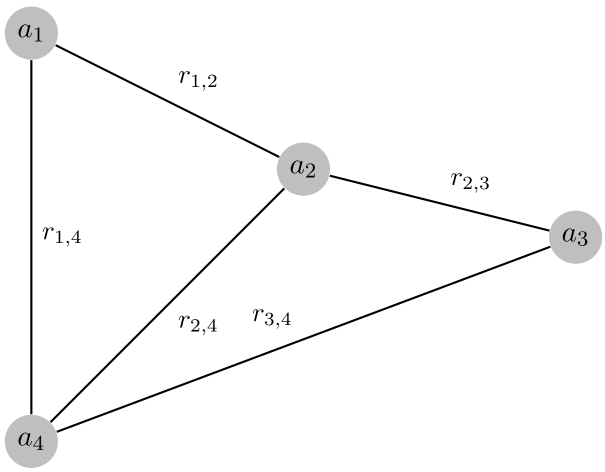
This section is a qualitative description of the market settings encountered in literature. We try to be as general as possible, whereas in Section 3 we focus on the frameworks Escobar, Figueroa, Heymann and Jofré study in [36], [35], [34] and [38]. The main market model components are the agents, the demand, the network, the regulation and the structure of information (since some uncertainty is usually part of the model). Different types of equilibria could be considered. An example of a network is proposed in Figure 5.1.

The agents are divided among those who produce electricity (producers) and those who consume it (usually aggregated into an ISO). In our setting, both producers and consumers are macroscopic, but for the sake of completeness, we note that some models use a continuum of microscopic producers. Such models correspond to situations where no producer can have any unilateral impact on the market.

Producers incur production costs when they supply electricity to the market. These costs depend on the quantity of electricity they each produce individually. The relation between the quantity a producer supplies and its production cost is encoded in his production cost function. To sell electricity, producers quote a price to the market. We consider two structures to model the way a producer specifies a selling price, which we discuss later in the chapter: the bid function and the supply function. A bid function maps any quantity of electricity to the price a producer asks to supply such a quantity. A supply function maps any price to the quantity a producer is ready to supply at such price. The objective of each producer is to maximize his individual profit (or his average profit if the model contains any form of randomness). We consider only non cooperative settings: producers are competing against each other.

We assume dispatch decisions to be centralized: a unique agent, the Independent System Operator (ISO), aggregates the demand side. We justify this aggregation by the

Figure 5.1 – An example of a wholesale electricity market network. On the demand side, at each node a_i there is a demand (load) d_i that has to be fulfilled. On the supply side, electricity can be produced at some nodes by some independent agents. The production cost function associates the quantity produced by an agent and the corresponding economic cost (it is specific to the agent). For modeling but also technical and practical reasons, those production cost functions are often approximated with functions in simple functional sets (linear, quadratic, piecewise linear...). Electricity can be sent from one node to another through the edges of the network, but there is a price for that (for instance, a loss proportional to the square of the quantity sent, due to resistivity). The Independent System Operator (ISO) is as its name indicates a central operator that has to allocate the production so that supply meet demand while minimizing a criteria (usually the total price). To produce this allocation, the ISO needs to know the price he will have to pay for each allocation, so the producers specify their bid functions that are usually in the same simple aforementioned functional set. Since the ISO has no way to know the real production cost function of the producers, it is in their interest to game the system. We point out that stochasticity may occurs on the demand. Moreover, the network aspect of the market is of primary importance, as it is responsible for the market power of the agent.



regulatory environment and the market organization. The ISO receives bids from producers and has to supply the local electricity demand where it is needed by buying the electricity at the quoted prices (pay as bid market). Generally, the demand is inelastic (in the literature we are considering), but it could be either deterministic or stochastic.

Usually electricity is seen as a divisible commodity. Nonetheless, in our model we can also see it as a geographically differentiated product. Unless production facilities and consumers are colocalized, productions and demands are dispatched on a network. The network contains nodes and edges. Each node is a place where electricity is produced or consumed (or both) and as such is characterized by its local demand and its local producers. Each edge is a place through which electricity can be sent. The ISO has the possibility to send any nodal electricity surplus where it is needed through the network cables but those cables are subject to physical limitations such as capacity constraints and online losses (due to Joule effects). The geographical differentiation comes from the fact that it the ISO which incurs those losses.

In our model, we envision different kinds of optimization problems: the standard ISO problem, the agent's profit maximization problem and the mechanism design problem. The first one consists simply in finding the minimal cost production plan. For this problem, the ISO (or the principal, if we wish to use mechanism design terminology) receives bids from the different producers and knows the demand (deterministic or stochastic) at each node. He then has to supply this demand at each node for the cheapest total cost. This optimization is subject to the network's physical limitations and is parametrized by the demand at each node and the producers' bids. We call the function that maps these parameters with the solution of the corresponding problem *principal response.*

The second problem stems from the agents' perspectives. Knowing the principal response function, their own production cost functions and possessing some common knowledge on their fellow producers, they optimize their bids to maximize their individual profits. This problem raises questions about best response strategies and Nash equilibria.

As alluded previously, in mechanism design, there is an agent and a principal. In our model, we assign the role of the principal to the ISO. The mechanism design problem reverts the role of the principal and the agents: the principal builds his response function knowing that the agents will then maximize their profits. By offering the right incentives, he leads the agents. The mechanism design problem can be formulated by considering that the principal gives a new response function that is not the optimal solution of the first problem. Indeed, instead of waiting for the bids to order the production, the ISO defines in a contract a response function that he will respect in the future. This contract depends on the (future) bids of each producer and the demand at each node. This occurs because otherwise there would not exist incentives to convince the agents to tell the truth about their production costs. They would optimize their own benefits based on the information they have. We have shown that in some very general setting, it is possible for the principal to formulate the response function in the contract so that the producers are incentivized to reveal their true types (i.e. real production costs). Put differently, the principal can design a contract so that for each producer, it is optimal to reveal his true production costs function. To do so, the principal has to pay (virtually, through the payment function defined in the contract) an information rent to the producers, but his total cost is smaller than in the previous setting.

In general, the principal does not know the real production cost of the producers. This iis the reason why producers can bid higher than their production cost. The information the principal has about producers' costs is modeled by a probability distribution. The less the producer knows about production costs, the higher the information rent.

5.3 Literature

Several approaches have been proposed to answer the questions raised in the previous section. In [59], Klemperer and Meyer show that uncertainty reduces the quantity of supply function Nash equilibria. The firms bid their supply functions before demand is revealed. The existence of a Nash equilibria is shown for a symmetric oligopoly. In [4], Anderson and Philpott show how to construct optimal time dependent supply functions in electricity market settings where demand and competing generators' behaviors are unknown by introducing a market distribution function. The gaming aspect of the situation is reduced by arguing that competitors do not react to the producer bids. The problem is formulated as an optimal control problem. In [3] the authors study asymmetric competition and propose a numerical solver based on GAMS to compute the optimal strategies. They compare the algorithm with the ODE method. In [1], Anderson gives a proof of existence of a pure Nash Equilibria under some technical assumptions when the network is reduced to a single point. He also gives sufficient conditions for uniqueness. Optimal auction design was introduced by Myerson in his 1981 seminal article [70]. Laffont and Martimort wrote in [61] an introduction to mechanism design in a general setting. These authors expose important concepts such as the revelation principle, adverse selection, participation constraints and information rent. The book does not consider interactions on a network -which is the specificity of the wholesale electricity market. Bi-level approaches with quadratic production cost functions are proposed in [8] and [52] to study the ISO response functions and the Nash equilibria. For a reference on complementarity modeling, the reader can consult [?]. Escobar and Jofré show in [34] that in a random environment a Walrasian and a non-cooperative equilibria exist (for the non-cooperative equilibrium the distribution need to be atom-less) in this setting, the demand is elastic and the ISO maximizes the sum of the utility functions. Utility functions and cost functions are general.

5.4 Quantitative formulations

We briefly present in this section some questions of interest concerning models that fit into the general setting described in Section 2. Those questions were partially addressed in recent works by the authors.

5.4.1 Generality

We will generically use the notations i to refer to a node or its corresponding producer and q_i to refer to the quantity this agent produces. The nodes are connected by edges and we denote by $h_{i,i'}$ the quantity of electricity that is sent from node i to node i'. The market network is not necessary complete. We call d_i the demand at node i. Each producer quotes a bid denoted by b_i to the principal. This bid is a function of the quantity q_i. Each producer also informs the principal of the maximum quantity $\bar{q}_i$ he can produce. In general, the allocation problem is subject to network constraints, i.e., the vector h of components $h_{i,i'}$ has to be in a set H. For example the set H could be made of all vectors h such that $h_{i,i'} \leq h_{i,i'}^{max}$, which means that one cannot send an arbitrary big amount of electricity through the network.

5.4.2 The standard allocation problem

The principal receives bids from the agents and allocates the production so that:

- the allocation minimizes the total cost;
- the allocation respects the network and capacity constraints;
- supply is greater than demand at any node.

The last point corresponds to the nodal constraints. The supply at a given node i is the sum of the local production q_i and the importations from neighboring nodes $\sum_{i'} h_{i',i}$. To this we need to subtract the exportations to neighboring nodes $\sum_{i'} h_{i,i'}$ and the line losses. If we send a quantity h through an edge $\{i, i'\}$, we denote by $L_{i,i'}(h)$ the corresponding loss. We will count half of this loss at node i and the other half at node i'. We end up with the following nodal constraint:

$$q_i + \sum_{i'} [h_{i',i} - h_{i,i'}] - \sum_{i'} \frac{L_{i',i}(h_{i',i}) + L_{i,i'}(h_{i,i'})}{2} \geq d_i, \tag{5.1}$$

where the summations are performed over the nodes adjacent to i. All this being said, the generic allocation problem writes

$$\begin{aligned}
\underset{q,h}{\text{minimize}} \quad & \sum_i b_i(q_i) \\
\text{subject to} \quad & q_i + \sum_{i'}[h_{i',i} - h_{i,i'}] - \sum_i \frac{L_{i,i'}(h_{i',i}) + L_{i,i'}(h_{i,i'})}{2} \geq d_i \\
& q_i \leq \bar{q}_i \\
& q_i \geq 0 \\
& h \in H \\
& h_{i,j} \geq 0.
\end{aligned} \tag{5.2}$$

We point out that if the bidding and the loss functions are convex functions and H is a convex set, then the problem is convex. For instance, one can take the bid functions linear, the loss functions quadratic and H equal to $\mathbb{R}^+$. Observe that for a convex problem, if L is strictly convex then at optimality $h_{i',i}h_{i,i'} = 0$. Note that the bid functions b_i and the demand vector d can be seen as parameters of the optimization problem. We could make the solution of this problem stochastic by adding a dependency of d to a random variable ω. This would not change the solution of the problem from the operator perspective, but it would change the market setting for the agents. **What is the solution of this deterministic allocation problem? What are the analytical properties of this solution? How can we compute it?**

5.4.3 The agent problem

The objective of each producer is to maximize his profit. Note that by solving the principal allocation problem, we have the response function of the ISO to the agents bids. It is stochastic if the demand is stochastic. We can map each bidding profile of the agents with the expected profit of each agent. By competing against each other, the agents are playing a game. In addition, producer i does not know the production cost functions of his fellow agents. As a result, we are in an imperfect information setting. We assume that for each agent i there is a probability distribution f_i over a set of potential production cost functions. The probability distribution f_i represents the (common) information the other agents have about agent i. We assume those probability distributions to be independent. The profit of an agent of type (i.e. production cost function) c_i that bids b_i (that associates any production level with the price he asks or such production level) is given by

$$\pi_i(c_i, b_i) = \int_{C_{-i}} [b_i(q_i(b_i, b_{-i}(c_{-i})) - c_i(q_i(b_i, b_{-i}(c_{-i})))]f_{-i}(c_{-i})dc_{-i}, \tag{5.3}$$

where the integral is performed over the types of the other agents. In this expression, $q_i(b_i, b_{-i})$ corresponds to the production level of producer i in the ISO allocation plan when the bids are (b_i, b_{-i}). The production cost function c_i associates the production level q_i and the corresponding true cost for producer i: $c_i(q_i)$. The function $b_{-i}(c_{-i})$ is the vector of bidding functions of the other producers when their types is the vector c_{-i}. Then the maximized profit is

$$\overline{\pi}_i(c) = \max_{b_i} \int\limits_{C_{-i}} [(b_i(q_i(b_i, b_{-i}(c_{-i})) - c(q_i(b_i, b_{-i}(c_{-i}))] f_{-i}(c_{-i}) dc_{-i}. \tag{5.4}$$

So for each agent the best response strategy to the other agents is the solution of an optimization problem on the set of maps from the types (i.e. production cost functions) c to the bids b. Usually, the production cost functions will be characterized by a vector of $\mathbb{R}^n$. In this case this is an optimization over the functions from $\mathbb{R}^n$ to $\mathbb{R}^n$. Observe that this setting corresponds to a (Bayesian) Bertrand game. Of course, it is natural to ask about the Nash Equilibria of the game. We point out that when $L = 0$, and there are no network and capacity constraints (and of course, the network is connected), the problem corresponds to the classic setting of first best auction theory (see Figure 5.3). **What can we say about this game? Is there an equilibrium ? Is it unique ? Can the agents 'game' the system** ?

5.4.4 The optimal mechanism design problem

In this section we assume that every participant knows the demand. As in 5.4.3, only producer i knows his true type c_i. The other agent and the ISO only know f_i. In order to decrease the market power of the agents and increase social welfare, we reverse the role of the principal (the ISO) and the agents, i.e. the principal "bids" a contract to the agent. The contract should associate each bid profile $(b_i)_i$ with two vectors q and x, where q_i is the quantity of electricity agent i has to produce and x_i is amount of money he will receive. This contract is communicated to the producers before the bidding phase. Of course, this contract has to be incentive compatible, i.e. the payments described by the principal need to be high enough to make the agent willing to stay in the market. In this situation, the problem we are solving is the design of the optimal contract:

$$\begin{aligned}
\underset{q_j, h_{i,k}, x_j}{\text{minimize}} \quad & \sum_j \mathbb{E} x_j(c) \\
\text{subject to} \quad & q_j(c) + \sum_i h_{i,j} - h_{j,i} - \sum_i \frac{L_{i,j}(h_{i,j}) + L_{j,i}(h_{j,i})}{2} \geq d_j \\
& \mathbb{E} x_j(c) - c_j(q_j(c)) \geq \mathbb{E} x_j(b) - c_j(q_j(b)) \\
& \mathbb{E} x_j(c) - c_j(q_j(c)) \geq 0 \\
& h_{i,j}, x_j \geq 0,
\end{aligned} \tag{5.5}$$

where $\mathbb{E}$ denotes the mean operator with respect to the f_i's, c denote the vector of production cost functions and the constraints should be verified for all c. We refer to [38] for a justification of the formulation. We point out that this is an optimization problem over a functional set (so infinite dimensional) with an infinite number of constraints. The solution of this problem is an optimal mechanism, i.e. based on the information to the ISO, it provides the allocation and payment rules (q, x) that minimize the expected payments to the producers. We display some results in Figures 5.3 and 5.4. **How do we build such problem? How can we solve it? How much better is the social surplus with an optimal design?**

5.4.5 A differential equation

As noted in 5.4.3, the agents are playing a Bayesian Bertrand-like game in the standard setting. In this section we propose a technique to compute a Nash equilibrium of this game. It is based on a fictitious play like dynamics. For instance, consider the simplified binodal, symmetric setting:

- 2 agents;
- $L_{i,i'}(h_{i,i'}) = r h_{i,i'}$;
- $H = \mathbb{R}^2_+$;
- $\bar{q}_i = +\infty$;
- the cost functions and the bid functions are linear;
- $d_1 = d_2$: the demand is equal at each node.
- $f_1 = f_2 = f$

We look for a symmetric equilibrium. If the agents iteratively change their bid functions proportionally to the corresponding increase in profit this will produce, the bid functions dynamics should be described by this formal differential equation.

$$\partial_t b(c,t) = \partial_b \pi_b(c, b(c,t))) \tag{5.6}$$

with

$$\pi_b(c,s) = \int_{C_{-i}} (s-c)(q_i(s, b(c_{-i}))f(c_{-i})dc_{-i}. \tag{5.7}$$

Is this dynamical system well posed? What conclusions can we draw from its study? Can we build such dynamics for more general settings?

5.5 Important results

In this section we sum up some results concerning the setting introduced previously. Most of the results focus on quadratic externalities (i.e. $L_{i',i}(h_{i',i}) = rh^2_{i',i}$) which is a realistic and simple assumption. Escobar and Jofré demonstrate in [36] the existence of non-cooperative and Walrasian equilibrium when the ISO solves the standard ISO problem and demand is uncertain. The paper finishes with a welfare theorem for wholesale electricity auction. Escobar and Jofré give in [35] a lower bound on the market power exercised by each producer. The existence of a mixed strategy Nash equilibrium is given.The authors also give some regularity property on the ISO response function (condition to be a singleton, continuity and Lipschitzianity). The cost functions are general. Figueroa, Jofré and Heymann study in [38], a bi-nodal symmetric market with linear production cost functions and quadratic losses (i.e. $L_{i,i'}(h_{i,i'}) = rh^2_{i,i'}$, see Figure 5.3). The principal minimal cost production plan problem was already solved in [35] and an explicit solution given. If we define

$$F(x,y) = d + \frac{1}{2r}\left(\frac{x-y}{x+y}\right)^2 - \frac{1}{r}\left(\frac{x-y}{x+y}\right) \quad \text{and} \quad \overline{q} = 2\left[\frac{1-\sqrt{1-2dr}}{r}\right], \tag{5.8}$$

then the solution of the standard allocation problem is

$$q_i(c_i, c_{-i}) = \begin{cases} F(c_i, c_{-i}) & \text{if } F(c_i, c_{-i}) \geq 0 \text{ and } F(c_{-i}, c_i) \geq 0 \\ \overline{q} & \text{if } F(c_{-i}, c_i) < 0 \text{ and } F(c_i, c_{-i}) \geq 0 \\ 0 & \text{if } F(c_i, c_{-i}) < 0 \text{ and } F(c_{-i}, c_i) \geq 0 \end{cases} \tag{5.9}$$

This solution is used to compute an explicit solution of the mechanism design problem. The mechanism design solution is then compared to the standard setting for which numerical simulations are performed. The authors assume that the function $J_i : c_i \longrightarrow c_i + \frac{F_i(c_i)}{f_i(c_i)}$ is increasing in c_i, where f_i is the distribution of the marginal cost of producer i and F_i is its integral. Then the main result is

Proposition 5.1. *Define*

$$\tilde{q} = 2\left[\frac{1-\sqrt{1-2dr}}{r}\right]$$

Then an optimal mechanism is given by

$$\begin{aligned}
\hat{q}_i(c) &= \begin{cases} F(J_i(c_i), J_{-i}(c_{-i})) & \text{if } F(J_i(c_i), J_{-i}(c_{-i})) \geq 0 \\ & \text{and } F(J_{-i}(c_{-i}), J_i(c_i)) \geq 0 \\ 0 & \text{if } F(J_i(c_i), J_{-i}(c_{-i})) \leq 0 \\ \tilde{q} & \text{if } F(J_{-i}(c_{-i}), J_i(c_i)) \leq 0 \end{cases} \\
\hat{h}_i(c) &= \begin{cases} \frac{1}{r}\left[\frac{J_{-i}(c_{-i})-J_i(c_i)}{J_{-i}(c_{-i})+J_i(c_i)}\right] & \text{if } J_i(c_i) \leq J_{-i}(c_{-i}) \text{ and } F(J_{-i}(c_{-i}), J_i(c_i)) \geq 0 \\ \tilde{q}-1 & \text{if } J_i(c_i) \leq J_{-i}(c_{-i}) \text{ and } F(J_{-i}(c_{-i}), J_i(c_i)) \leq 0 \\ 0 & \text{if not} \end{cases} \\
\hat{x}_i(c) &= c_i\hat{q}_i(c) + \int_{c_i}^{\bar{c}_i} q_i(s, c_{-i})ds
\end{aligned}$$

We point out that the mechanism is built with the standard ISO response function, where we just replace c_i by $J_i(c_i)$. The mechanism is incentive compatible, i.e. for any agent of any type, bidding the true type ensures a better profit for the agent than any other bidding strategy. Also the mechanism should satisfy a participation constraint, so that any agent can make a nonnegative profit. The optimal mechanism minimizes the total expected payment from the ISO to the agents while satisfying the incentive compatibility constraints, the participation constraints and the nodal constraints.

5.6 The ISO response for a binodal setting with piecewise linear cost

5.6.1 Introduction

In this section we derive an explicit expression for a specific example of ISO allocation problem as defined in 5.4.2 We study the bi-nodal market with quadratic externalities displayed in Figure 5.3. The production cost functions of both agents are made of two linear pieces, with a slope change when the production level is equal to $\bar{q}$. We denote by c_1 (resp. c_2) producer 1 (resp. 2) marginal cost when his production level is below $\bar{q}$, and by $\bar{c}_1$ (resp. $\bar{c}_2$) when it is above. The production cost functions are convex i.e. $c_i < \bar{c}_i$ and the demand d is the same at both nodes. We end-up with the following formulation for the ISO allocation problem:

$$\begin{aligned}
\underset{q_i,\bar{q}_i,h}{\text{minimize}} \quad & c_1 q_1 + \bar{c}_1 \bar{q}_1 + c_2 q_2 + \bar{c}_2 \bar{q}_2 \\
\text{subject to} \quad & q_i + \bar{q}_i + (-1)^i h \geq \frac{r}{2}(h^2) + d \qquad (\lambda_i) \quad \text{for i} = 1,2 \\
& q_i, \bar{q}_i \geq 0 \qquad (\mu_i) \quad \text{for i} = 1,2 \\
& q_i \leq \bar{q} \qquad (\gamma_i) \quad \text{for i} = 1,2.
\end{aligned} \tag{5.10}$$

In this formulation, q_i is the quantity produced by agent i at marginal cost c_i, and $\bar{q}_i$ is the quantity produced by i at marginal cost $\bar{c}_i$. These quantities are subject to positivity constraints with multipliers μ_i and $\bar{\mu}_i$. We also introduce λ_i the multipliers of the nodal constraints, and γ_i the multipliers of the constraints $q_i \leq \bar{q}$. We denote

$$F(\lambda_1, \lambda_2) = d + \frac{1}{r}\frac{\lambda_2 - \lambda_1}{\lambda_1 + \lambda_2} + \frac{1}{2r}\left(\frac{\lambda_2 - \lambda_1}{\lambda_1 + \lambda_2}\right)^2, \tag{5.11}$$

$$P(h) = h + \frac{rh^2}{2} + d, \tag{5.12}$$

$$k(\lambda_1, \lambda_2) = P\left(\frac{\lambda_2 - \lambda_1}{r(\lambda_1 + \lambda_2)}\right), \tag{5.13}$$

and

$$q_i^{tot} = q_i + \bar{q}_i. \tag{5.14}$$

We assume without loss of generality that $\bar{q} < 2d$ and $c_1 < c_2$. It is clear that if $q_1 < \bar{q}$, then $\bar{q}_1 = 0$. To solve this problem, we check whether $d < \bar{q}$ or $d \geq \bar{q}$.

5.6.2 If $d < \bar{q}$

By hypothesis $c_1 < c_2$. This implies that $q_1 \geq q_2$. So $\bar{q}_2 > 0$ implies that $q_1 = q_2 = \bar{q} > d$, which is not optimal. Therefore we can set $\bar{q}_2 = 0$. We can also relax the constraint $q_2 < \bar{q}$ because it won't be binding for the optimal solution. So we rewrite the problem

$$\begin{aligned}
&\underset{q_i,\bar{q}_1,h}{\text{minimize}} \quad c_1 q_1 + \bar{c}_1 \bar{q}_1 + c_2 q_2 \\
&\text{subject to} \\
&q_1 + \bar{q}_1 - h \geq \frac{r}{2}(h^2) + d \qquad (\lambda_1) \\
&q_2 + h \geq \frac{r}{2}(h^2) + d \qquad (\lambda_2) \\
&q_1, q_2, \bar{q}_1 \geq 0 \qquad (\mu_i) \\
&q_1 \leq \tilde{q} \qquad (\gamma_1)
\end{aligned}$$

The first order conditions give

$$c_1 - \lambda_1 - \mu_1 + \gamma_1 = 0 \tag{5.15}$$
$$c_2 - \lambda_2 - \mu_2 = 0 \tag{5.16}$$
$$\bar{c}_1 - \lambda_1 - \bar{\mu}_1 = 0 \tag{5.17}$$
$$h = \frac{\lambda_2 - \lambda_1}{r(\lambda_1 + \lambda_2)} \tag{5.18}$$

There are four possible cases.

***Case 1*: $P(\frac{c_2-c_1}{r(c_1+c_2)}) \leq \bar{q}$**

We consider a relaxation of the problem by removing the constraint $q_1 \leq \bar{q}$. In this relaxed problem, any optimal solution should verify $\bar{q}_1 = 0$ so the relaxed problem is equivalent to the linear cost functions allocation problem with costs c_i, for which we have an explicit formula of the solution. **We then notice that the optimal solution of the relaxed problem is admissible, so it is also the solution of** (5.10)**.**

***Case 2*: $P(\frac{c_2-c_1}{r(c_1+c_2)}) > \bar{q}$ and $P(\frac{c_2-\bar{c}_1}{r(\bar{c}_1+c_2)}) \leq \bar{q}$**

We show that $\bar{q}_1 = 0$ and $q_1 = \bar{q}$.
<u>If $\bar{q}_1 > 0$</u>, then by complementarity of the multiplier $\bar{\mu}_1 = 0$, so with (5.17) $\lambda_1 = \bar{c}_1$. So by (5.18) we have $h = \frac{\lambda_2-\bar{c}_1}{r(\bar{c}_1+\lambda_2)}$. Then by hypothesis and the fact that P is increasing and $\lambda_2 \leq c_2$ we have that $P(h) \leq P(\frac{c_2-\bar{c}_1}{r(\bar{c}_1+c_2)}) \leq \bar{q}$. So $q_1 + \bar{q}_1 = q^{tot} \leq \bar{q}$. Then using the fact that $q_1 < \bar{q} \Rightarrow \bar{q}_1 = 0$, we deduce that $\bar{q}_1$ is null, which is not the hypothesis. We conclude that <u>$\bar{q}_1 = 0$</u>.
By hypothesis, $\bar{q} < 2d$ and less than $\bar{q}$ is produced at node 1. Summing the two nodal constraints we see that $q_2 > 0$, so that <u>$\lambda_2 = c_2$</u>.
Now <u>if $q_1 < \bar{q}$</u>, then by complementarity of the multiplier $\gamma_1 = 0$, then with (5.15) $c_1 = \lambda_1$ and with (5.18), $h = \frac{c_2-c_1}{r(c_1+c_2)}$. Therefore we get $q_1 + \bar{q}_1 = q_1^{tot} \geq P(\frac{c_2-c_1}{r(c_1+c_2)})$ (by the first nodal constraint), so by hypothesis $q_1^{tot} \geq \bar{q}$ and so $\bar{q}_1$ which we know as false. <u>So $q_1 = \bar{q}$</u>.

***Case 3*: $P(\frac{c_2-c_1}{r(c_1+c_2)}) > \bar{q}$ and $P(\frac{c_2-\bar{c}_1}{r(\bar{c}_1+c_2)}) > \bar{q}$ and $P(\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)}) > 0$**

We show that $q_2 > 0$, $q_1 = \bar{q}$, $\bar{q}_1 > 0$, and, $q_2 = P(\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)})$.
<u>First we show that $q_2 > 0$.</u> If $q_2 = 0$, then by the second nodal constraint $h \geq \frac{rh^2}{2} + d$, which means that $P(-h) \leq 0$. Moreover, $\bar{q}_1 > 0$ because $2d > \bar{q}$. So by (5.17) $\lambda_1 = \bar{c}_1$. With (5.16) and (5.18) we have $P(\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)}) \leq P(\frac{\bar{c}_1-\lambda_2}{r(\bar{c}_1+\lambda_2)}) = P(-h) \leq 0$, which is false by hypothesis. <u>So $q_2 > 0$</u>. We deduce from this and (5.16) that $\lambda_2 = c_2$.
<u>If $q_1 < \bar{q}$</u> then by complementarity of the multiplier $\gamma_1 = 0$, so by (5.15), $\lambda_1 = c_1$ and by (5.18), $h = \frac{c_2-c_1}{r(c_1+c_2)}$. Then we get $q_1^{tot} \geq P(\frac{c_2-c_1}{r(c_1+c_2)})$ (by the first nodal constraint), so by

hypothesis $q_1^{tot} \geq \bar{q}$, which implies $q_1 = \bar{q}$, which is absurd since we assumed $q_1 < \bar{q}$. So $q_1 = \bar{q}$.

If $\bar{q}_1 = 0$ then with (5.17), $\lambda_1 \leq \bar{c}_1$ and so with (5.18), $h \geq \frac{c_2-\bar{c}_1}{r(\bar{c}_1+c_2)}$. We then deduce by nodal constraint 1 and the hypothesis that $q_1^{tot} \geq P(h) \geq \bar{q}$, which implies that $\bar{q}_1 > 0$, which is absurd. So $\bar{q} > 0$.

We know that $\bar{q}_2 = 0$. Using the second nodal constraint, we get $q_2 = P(\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)})$. With the first nodal constraint, we have $\bar{q}_1 = P(\frac{c_2-\bar{c}_1}{r(\bar{c}_1+c_2)}) - \bar{q}$.

***Case 4*: $P(\frac{c_2-c_1}{r(c_1+c_2)}) > \bar{q}$ and $P(\frac{c_2-\bar{c}_1}{r(\bar{c}_1+c_2)}) > \bar{q}$ and $P(\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)}) \leq 0$**

We show that $q_2 = 0$.

Indeed, if $q_2 > 0$, then $\lambda_2 = c_2$. Using the same reasoning as the one used in the third case, we would show that $q_1 = \bar{q}$ and $\bar{q}_1 > 0$. So that $h = -\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)}$. So using $P(\frac{\bar{c}_1-c_2}{r(\bar{c}_1+c_2)}) \leq 0$, we see that nodal constraint 2 is satisfied with $q_2 = 0$, so the solution is not optimal, which is absurd. So $q_2 = 0$.

We conclude

Theorem 5.2. *Assuming $d < \bar{q} < 2d$, then:*

$$\begin{aligned}
q_1^{tot} = k(c_1,c_2) \text{ and } q_2^{tot} = k(c_2,c_1) \text{ if } k(c_1,c_2) \leq \bar{q} \\
q_1^{tot} = \bar{q} \text{ and } q_2^{tot} = \bar{q} - 2\frac{-1+\sqrt{1+2r(\bar{q}-d)}}{r} \text{ if } k(c_1,c_2) > \bar{q} \text{ and } k(\bar{c}_1,c_2) \leq \bar{q} \\
q_1^{tot} = k(\bar{c}_1,c_2) \text{ and } q_2^{tot} = k(c_2,\bar{c}_1) \text{ if } k(c_1,c_2) > \bar{q}, k(\bar{c}_1,c_2) > \bar{q} \text{ and } k(c_2,\bar{c}_1) > 0 \\
q_1^{tot} = 2\frac{1-\sqrt{1-2dr}}{r} \text{ and } q_2^{tot} = 0 \text{ if } k(c_1,c_2) > \bar{q},\ k(\bar{c}_1,c_2) > \bar{q} \text{ and } k(c_2,\bar{c}_1) \leq 0
\end{aligned}$$

5.6.3 Case $d \geq \bar{q}$

Since we consider that c_1 is smaller than c_2, there are two possibilities. Either the $\bar{c}_i$ are all bigger than the c_i, or $\bar{c}_1$ is smaller than c_2.

If the $\bar{c}_i$ are all bigger than the c_i

In this case, we first show that $q_1 = q_2 = \bar{q}$. The problem then writes

$$\begin{aligned}
\underset{q_i,\bar{q}_i,h}{\text{minimize}} \quad & \bar{c}_1\bar{q}_1 + \bar{c}_2\bar{q}_2 \\
\text{subject to} \quad & \bar{q}_1 + -h \geq \frac{r}{2}(h^2) + d - \bar{q} && (\lambda_1) \\
& \bar{q}_2 + h \geq \frac{r}{2}(h^2) + d - \bar{q} && (\lambda_2) \\
& \bar{q}_i \geq 0 && (\mu_i) \quad \text{for i} = 1,2
\end{aligned}$$

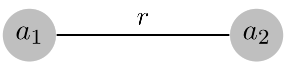


Figure 5.2 – In [38] and [35], the authors consider a binodal market with quadratic line losses. The demand is the same at both nodes. The production cost functions are linear. There are no network and capacity constraints. A very intuitive justification of the market power induced by the line losses is given in [35]. Indeed, in a symmetric perfect information setting with linear production cost function of slope c, the equilibrium strategy of both producers is to bid $\frac{c}{1-2dr} > c$.

which corresponds to the linear case problem with a demand of $d - \bar{q}$ and costs of $\bar{c}_i$.

If $\bar{c}_1$ is smaller than c_2

We point out that replacing c_1 by $\bar{c}_1$ does not change the solution.
If $F(c_2, \bar{c}_1) \leq \bar{q}$, we show that $\bar{q}_2 = 0$ the problem can be reduced to the linear production cost problem with demand d and marginal costs $\bar{c}_1$ and c_2.
If $F(c_2, \bar{c}_1) > \bar{q}$, we show that we can reduce the linear production cost problem with demand $d - q$ and marginal costs $\bar{c}_1$ and $\bar{c}_2$.

Theorem 5.3. *If $d \geq \bar{q}$, then*

- *If $c_i \leq \bar{c}_j$ for all i, j, then we get the result by solving the linear problem with demand $d - \bar{q}$ and costs $\bar{c}_i$ and adding $\bar{q}$ to the quantity we get.*
- *If $0 \leq F(c_2, \bar{c}_1) \leq \bar{q}$ we reduce to the linear allocation problem with demand d and marginal cost $\bar{c}_1$ and c_2.*
- *If $F(c_2, \bar{c}_1) > \bar{q}$, we reduce the problem to the linear allocation problem with demand $d - q$ and marginal costs $\bar{c}_1$ and $\bar{c}_2$ and add $\bar{q}$ to the q_is we get.*

5.7 Ongoing work

We are currently working on several questions raised in this chapter. In particular, we have shown that for a market with n-pieces piecewise linear production cost functions and any number of producers, there is a mechanism design with an explicit formulation.

5.8 Acknowledgments

The authors would like to thank Idalia Gonzalez for her English editing.

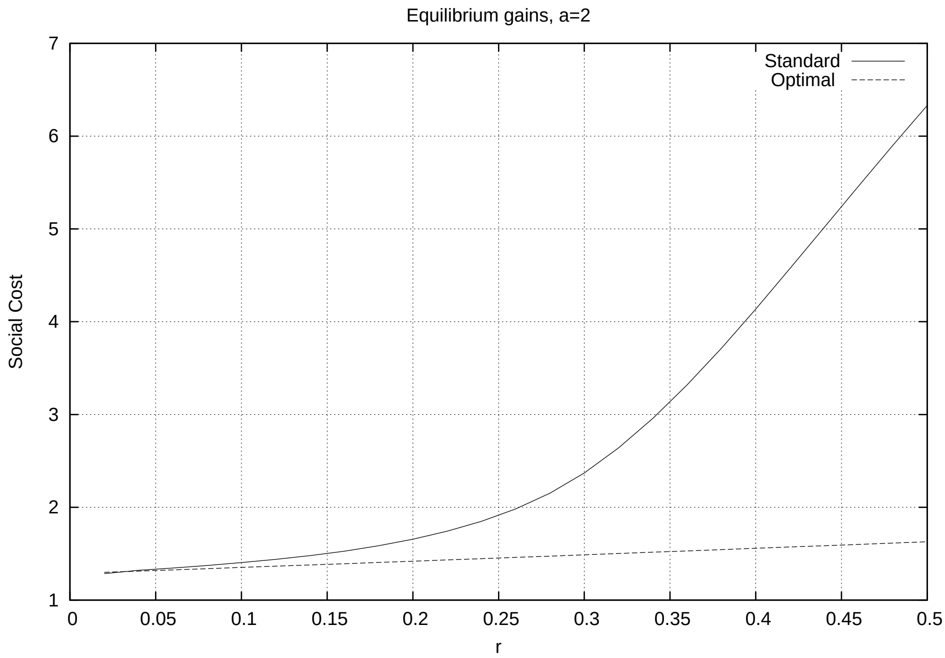


Figure 5.3 – The average total cost for the ISO (in the market described in Figure 5.3) as a function of the loss coefficient r for the standard mechanism and the optimal mechanism. We take $f_1(c) = f_2(c) = 2c + (1 - \frac{2}{4})1_{c \leq \frac{1}{2}} + -2c + (1 + \frac{3}{2})1_{c \geq \frac{1}{2}}$. Note how r influences the social cost in the standard mechanism. The agents market power increase with r. When r goes to zero, the two mechanisms lead to the same social cost. When $r = 0$ we recover a classic result on first and second best auctions.

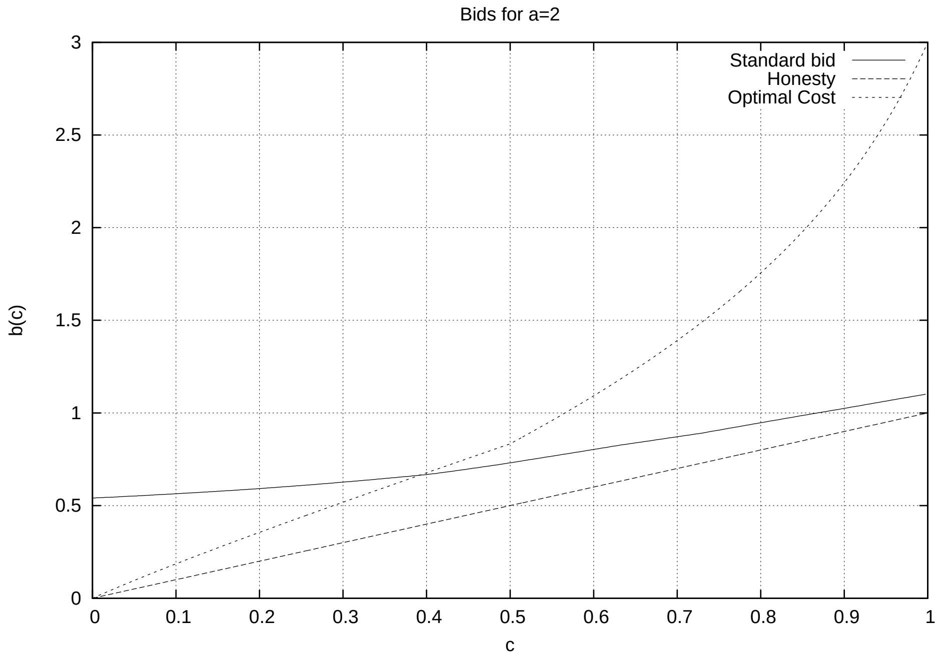


Figure 5.4 – Comparison of bidding strategy and information rent for the market described in Figure 5.3. The standard bid strategy corresponds to the equilibrium strategy of the Bayesian Game. The Honesty strategy correspond to a producer telling the truth. The optimal cost correspond to the sum of the truth-telling strategy and the information rent.

Chapter 6

Cost-Minimizing regulations for a wholesale electricity market

This is a joint work with Alejandro Jofré and Nicolas Figueroa.

Contents

We consider a wholesale electricity market model with general networks, transmission losses and strategic producers. Previous work by Escobar and Jofré [35] shows how regulation mechanisms, for instance when prices correspond to the Lagrange multipliers of a centralized cost minimization program, allow the producers to charge significantly more than marginal price. In this paper we consider an incomplete information setting where the cost structure of a producer is unknown to both its competitor and the regulator. We derive an optimal regulation mechanism, and compare its performance to the "price equal to Lagrange multiplier" mechanism in an incomplete information setting (that we solve numerically).

6.1 Introduction

A common design in many recently liberalized electricity markets (e.g. US, UK, Spain, etc) involves wholesale trading through an integrated market. Frequent market clearing processes take place, through rules managed by a central authority, commonly referenced as an independent system operator (ISO). In the context of this short-term operations, demand is inelastic and must be satisfied through a grid that imposes significant constraints on the allocations that can be selected. A particularly important feature of the problem is that generators are connected through a grid to the places where demand is concentrated, and there are important losses through the transmission lines. This makes electric power, which is a priori an homogeneous good, a differentiated product, giving generators some degree of market power. This has been remarked, among others, in the work by Borenstein et al. [21], which emphasizes that "transmission costs will be at the heart of market power issues in a restructured electricity market".

This situation originates a very important regulatory problem. Given a set of generators and a vector of demands in a network, with production costs that are private information, what pricing scheme should be used in order to minimize the total cost paid and such that the vector of demands is satisfied? An extensively used method is the following: ask each generator its cost function, solve the problem of minimizing cost subject to the fulfillment of the demand requirements, then pay, at each node, a price equal to the Lagrange multiplier associated to that constraint. In such a game, generators have incentives to "shade their bids", and bid costs above their true levels. This is true even if there is complete information among generators but transmission costs are not zero, as it was shown in Escobar and Jofré [35] in a simple network with two nodes and a fixed symmetric cost structure.

With this in mind, we solve the problem of designing an optimal (cost minimizing) mechanism in the presence of incomplete information and transmission losses in a simple network. By using mechanism design tools, we completely characterize the optimal mechanism, derive explicit formulas and compute the total cost for the regulator. Moreover, in order to compare the performance of the optimal mechanism with the standard one, we numerically solve the Bayesian equilibrium of the game induced by the standard mechanism.

Some important features appear. In the absence of transmission costs, both the standard and the optimal mechanism yield the same expected costs. The intuition comes from standard auction theory. In the absence of transmission costs there is a single object being bought (the procurement of all nodes in the network), the standard mechanism is equivalent to a first-price procurement auction and the optimal one is a second-price auction. Since both assign to the lowest cost generator, the revenue equivalence principle implies that both yield the same expected costs.

However, if transmission costs are non-zero, the standard and the optimal mechanism allocate production differently, and have therefore different expected costs. Basically, the distortions induced by (individually-optimal) bid-shading in the standard mechanism are different than those induced by the cost-minimizing mechanism. While both induce inefficient allocations, the ones in the mechanism we characterize are designed so that expected total cost is minimized. It is important to remark that the allocative first best, the assignment that would actually result if there was complete information, can be achieved through a VCG mechanism. However, such a mechanism would not minimize expected total cost. From a cost-minimizer perspective, it is *optimal* to induce inefficiencies in order to minimize the producer's informational rents.
We show that the implementation of the cost-minimizing mechanism is essentially of the same complexity as the implementation of the standard one. To compute the optimal allocation, it requires solving the same problem the ISO operator solves in the standard one, but with modified parameters. To compute the optimal transfers, which are an essential part of the rules of the game, a parameterized optimization problem must be solved.

6.2 The Problem

It is well known that price competition with differentiated products does not necessarily lead to zero profits for the firms. In the context of electricity markets, the presence of transmission losses makes the energy produced by different generators differentiated goods, thus creating power and allowing generators to charge prices above marginal cost. This effect was shown in Escobar and Jofré [35], in the context of symmetric producers and complete information. There, the mechanism "price equal to Lagrange multiplier" induces insufficient competition, leading to a Nash Equilibrium where both firms charge prices above marginal cost.

To keep things simple, we consider an electric network with two nodes, each of them with a fixed demand d. At each node, there exists a generator, and it is possible to transmit any amount of electricity h between the nodes, but in that case an amount rh^2 is lost. So the network would look like the one in Figure 6.1.

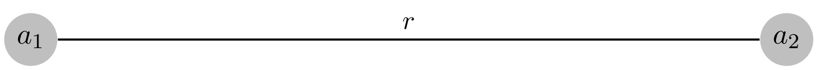


Figure 6.1 – The binodal market

The fact that firms exercise their market power under the classical regulatory system gives rise to some important questions. We know that with the current system firms charge prices above marginal cost but this does not seem completely avoidable. Is the current mechanism optimal? If not, how costlier than an optimal mechanism is it? Maybe more importantly, what is the best mechanism to regulate generators in the presence of transmission losses?

To study these natural regulatory questions, a degree of incomplete information, which is at the core of regulatory issues, is needed. First of all, it is realistic, since generators rarely know exactly the cost structure of competitors, and the ISO is not perfectly informed about generators either. Moreover, without private information the regulatory problem is trivial: just ask each generator to produce at marginal cost, minimizing cost and guaranteeing participation.

We introduce incomplete information in the marginal cost of a generator, c_i. We assume that c_i is drawn from a distribution with density f_i which has support $C_i = [\underline{c}_i, \overline{c}_i]$ and is positive in the interval. The parameter c_i is known by firm i, but the regulator and its competitor only know the distribution f_i (of cumulated distribution function F_i). We will use the notation $C = C_1 \times C_2$ and $f(c) = f_1(c_1)f_2(c_2)$.

Given the rules designed by the regulator, the generators maximize their expected profits. Knowing this behavior, the regulator aims to design a mechanism that satisfies demand at each node and minimizes expected cost. We address this problem in the next section, using the techniques of mechanism design.

6.3 A Mechanism Design Approach

We now study the problem of the optimal regulatory mechanism in the presence of transmission losses and private information. We assume that the regulator sets up some "rules of the game", which are then followed by the generators based on their private information.

Due to the presence of private information, these rules must allow the generators to reveal their preferences, which in this case are given by their cost structure. By the revelation principle we know that without loss of generality the regulator can restrict attention to incentive compatible direct revelation mechanisms, where the agents's message space is their type space C_i, and to incentive compatible mechanisms, where generators have incentives to tell the truth.

Definition 6.1. *A* direct revelation mechanism $M = (q, h, x)$ *consists of an* assignment rule $(q_1, q_2, h_1, h_2) : C \longrightarrow \mathbb{R}^4$ *and a* payment rule $x : C \longrightarrow \mathbb{R}^2$*.*

The assignment rule specifies the quantity that each generator must produce and the flows for each vector of reports. We denote by $q_i(c)$ the quantity that generator i must produce when the vector of reports is c and by $h_i(c)$ the surplus transmitted from generator i to generator j. The payment rule x specifies, for each vector of reports c, a vector of payments, one for each generator.

The ex-ante expected utility of a buyer of type c_i when he participates and declares c_i' is

$$U_i(c_i, c_i'; (q, h, x)) = E_{c_{-i}}[x_i(c_i', c_{-i}) - c_i q_i(c_i', c_{-i})]$$

Since we can consider only incentive compatible mechanisms, a natural notion of a feasible mechanism is the following:

Definition 6.2. *(Feasible Mechanisms) We say that a mechanism (q, h, x) is feasible iff*

$$\begin{aligned} U_i(c_i, c_i; (q, h, x)) &\geq U_i(c_i, c_i'; (q, h, x)) \text{ for all } c_i, c_i' \in C_i \text{ and } i = 1, 2 \\ U_i(c_i, c_i; (q, h, x)) &\geq 0 \text{ for all } c_i \in C_i \text{ and } i = 1, 2 \\ q_i(c) - h_i(c) + h_{-i}(c) &\geq \frac{r}{2}[h_1^2(c) + h_2^2(c)] + d \text{ for all } c \in C \text{ and } i = 1, 2 \\ q_i(c), h_i(c) &\geq 0 \text{ for all } c \in C \text{ and } i = 1, 2 \end{aligned}$$

The first set of constraints are the incentive compatibility constraints *IC*, the second one are the voluntary participation constraints, *PC*, and the last two ones impose the requirements that the quantities generated and the flows satisfy the demand and are positive quantities, in other words these are the resource constraints, *RES*.

At an incentive compatible direct revelation mechanism $M = (q, h, x)$, the regulator's expected cost is given by

$$\int_C \sum_{i=1,2} x_i(c) f(c) dc$$

Therefore, the problem of the planner can be written as

$$\min \int_C \sum_{i=1}^{I} x_i(c) f(c) dc \tag{6.1}$$

subject to (q, h, x) being "feasible".

6.3.1 Some Basic Results

We now investigate properties of feasible mechanisms. This will in turn allow us to write the regulator's problem in a simpler way. We use $V_i(c_i)$ to denote i's maximized surplus, that is

$$V_i(c_i) = \max_{c_i'} E_{c_{-i}}[x_i(c_i', c_{-i}) - c_i q_i(c_i', c_{-i})] \tag{6.2}$$

We will also define the expected quantity to be produced by a generator that declares a type c_i' (resp. expected payment) as

$$Q_i(c_i') \equiv E_{c_{-i}}[q_i(c_i', c_{-i})] \quad \text{(resp. } X_i(c_i') \equiv E_{c_{-i}}[x_i(c_i', c_{-i})])$$

Lemma 6.3. *A mechanism (q, h, x) is feasible iff*

$$\begin{aligned} Q_i(c_i') &\leq Q_i(c_i) \textit{ for all } c_i' > c_i & (6.3)\\ V_i(c_i) &= V_i(\bar{c}_i) + \int_{c_i}^{\bar{c}_i} Q_i(s)ds \textit{ for all } c_i \in C_i,\ i = 1,2 & (6.4)\\ q_i(c) - h_i(c) + h_{-i}(c) &\geq \frac{r}{2}[h_1^2(c) + h_2^2(c)] + d \textit{ for all } c \in C, i = 1,2 & (6.5)\\ q_i(c), h_i(c) &\geq 0 \textit{ for all } c \in C \textit{ and } i = 1,2 & (6.6) \end{aligned}$$

Proof. See Appendix □

With this, we can write the expected payment to a seller as a function of the assignment rule. This is expressed in the next lemma.

Lemma 6.4. *The expected payment of buyer i can be written as*

$$\int_C x_i(c)f(c)dc = \int_C q_i(c)[c_i + \frac{F_i(c_i)}{f_i(c_i)}]f(c)dc \tag{6.7}$$

Proof. See Appendix. □

With these two lemmas, we can characterize the seller's problem in the next proposition:

Proposition 6.5. *If in a mechanism $(\hat{q}, \hat{h}, \hat{x})$ the assignment function $(\hat{q}, \hat{h})$ solves*

$$\min_{q,h} \int_C \sum_{i=1,2} q_i(c)[c_i + \frac{F_i(c_i)}{f_i(c_i)}]f(c)dc \tag{6.8}$$

subject to the constraints (6.3), (6.5) and (6.6), and the payment function $\hat{x}$ satisfies

$$\hat{x}_i(c) = \hat{q}_i(c)c_i + \int_{c_i}^{\bar{c}_i} q_i(s, c_{-i})ds \tag{6.9}$$

then $(\hat{q}, \hat{h}, \hat{x})$ is an optimal mechanism.

Proof. Direct from lemma (6.3) and (6.4). □

Now, with a simple assumption on the distribution f, we can find an explicit solution of the problem.

Assumption 4. *The function* $J_i : c_i \longrightarrow c_i + \frac{F_i(c_i)}{f_i(c_i)}$ *is increasing in* c_i.

This assumption will guarantee that pointwise minimization of (6.8), subject to the feasibility constraints (6.5) and (6.6), leads to an allocation rule $q(c)$ that is incentive compatible. With this, the solution to the seller's problem is simple: for each realization of costs c, solve a cost minimization problem, but where the agents' costs are $J_i(c_i)$ instead of c_i.

Proposition 6.6. *Suppose that assumption 4 is satisfied.*

Define

$$F(x,y) = d + \frac{1}{2r}\left(\frac{x-y}{x+y}\right)^2 - \frac{1}{r}\left(\frac{x-y}{x+y}\right)$$

and

$$\overline{q} = 2\left[\frac{1-\sqrt{1-2dr}}{r}\right]$$

Then an optimal mechanism is given by

$$\hat{q}_i(c) = \begin{cases} d + \frac{1}{2r}\left[\frac{J_{-i}(c_{-i})-J_i(c_i)}{J_{-i}(c_{-i})+J_i(c_i)}\right]^2 + \frac{1}{r}\left[\frac{J_{-i}(c_{-i})-J_i(c_i)}{J_{-i}(c_{-i})+J_i(c_i)}\right] & \text{if } \begin{cases} F(J_i(c_i), J_{-i}(c_{-i})) \geq 0 \\ \text{and } F(J_{-i}(c_{-i}), J_i(c_i)) \geq 0 \end{cases} \\ 0 & \text{if } F(J_i(c_i), J_{-i}(c_{-i})) \leq 0 \\ \overline{q} & \text{if } F(J_{-i}(c_{-i}), J_i(c_i)) \leq 0 \end{cases}$$

$$\hat{h}_i(c) = \begin{cases} \frac{1}{r}\left[\frac{J_{-i}(c_{-i})-J_i(c_i)}{J_{-i}(c_{-i})+J_i(c_i)}\right] & \text{if } J_i(c_i) \leq J_{-i}(c_{-i}) \text{ and } F(J_{-i}(c_{-i}), J_i(c_i)) \geq 0 \\ \overline{q} - 1 & \text{if } J_i(c_i) \leq J_{-i}(c_{-i}) \text{ and } F(J_{-i}(c_{-i}), J_i(c_i)) \leq 0 \\ 0 & \text{if not} \end{cases}$$

$$\hat{x}_i(c) = c_i \hat{q}_i(c) + \int_{c_i}^{\overline{c}_i} q_i(s, c_{-i}) ds$$

Proof. Notice that $(\hat{q}, \hat{h})$ solve the problem of proposition 6.5 pointwise if the constraint (6.3) is relaxed (see appendix) so it is enough to show that (6.3) is satisfied. A sufficient condition is that the function $\hat{q}_i : (c_i, c_{-i}) \longrightarrow \hat{q}_i(c_i, c_{-i})$ is non-increasing in c_i for a fixed c_{-i}. A straightforward calculation[1] shows that $\frac{\partial \hat{q}_i(c_i, c_{-i})}{\partial c_i} = -\frac{4J_{-i}^2(c_{-i})J_i'(c_i)}{[J_i(c_i)+J_{-i}(c_{-i})]^2} \leq 0$ in the region where $\hat{q}_i(c_i, c_{-i})$ is not constant, so the result follows. □

6.4 The Standard Mechanism

A standard regulatory mechanism is the following: the regulator asks each generator his marginal cost of production and then, using these revelations as the true costs, solves the

[1] We assume differentiability of J_i just for simplicity, the proof can be done without it.

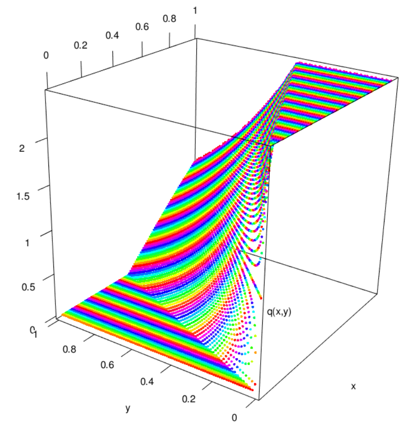


Figure 6.2 – The function q

problem of minimizing the total cost of production subject to the feasibility constraints. In our case this corresponds to solve, considering c_1 and c_2 as given:

$$\begin{array}{ll} \min\limits_{q,h} & \sum\limits_{i=1}^{2} c_i q_i \\ s.t. & q_i - h_i + h_{-i} \geq \frac{r}{2}[h_1^2 + h_2^2] + d \text{ for } i = 1,2 \\ & q_i, h_i \geq 0 \text{ for } i = 1,2 \end{array}$$

The regulator will then ask generator i to produce a quantity q_i and will pay him a unit price λ_i, where λ_i is the Lagrange multiplier associated with the feasibility constraint at node i.

If we define

$$F(x,y) = d + \frac{1}{2r}\left(\frac{x-y}{x+y}\right)^2 - \frac{1}{r}\left(\frac{x-y}{x+y}\right)$$

and

$$\overline{q} = 2\left[\frac{1-\sqrt{1-2dr}}{r}\right]$$

the solution to this problem can be written as (see appendix)

$$q_i(c_i, c_{-i}) = \begin{cases} F(c_i, c_{-i}) & \text{if } F(c_i, c_{-i}) \geq 0 \text{ and } F(c_{-i}, c_i) \geq 0 \\ \overline{q} & \text{if } F(c_{-i}, c_i) < 0 \text{ and } F(c_i, c_{-i}) \geq 0 \\ 0 & \text{if } F(c_i, c_{-i}) < 0 \text{ and } F(c_{-i}, c_i) \geq 0 \end{cases}$$

$$\lambda_i(c_i, c_{-i}) = \begin{cases} c_i & \text{if } F(c_i, c_{-i}) \geq 0 \\ \left[\frac{2-\sqrt{1-2dr}}{\sqrt{1-2dr}}\right] c_{-i} & \text{otherwise} \end{cases}$$

We display q in Figure 6.2.

6.4.1 The Bayesian Game

Consider now the situation we analyzed in the previous section: each generator independently draws a marginal cost $c_i \in C_i$. They then play a Bayesian game where the quantity asked from a generator that bids a cost x and confronts a generator who bids a cost y is given by $q_i(x, y)$, and the unit price paid to him is $p_i(x, y) = \lambda_i(x, y)$. Generators are profit maximizers.
The equilibrium of this game would be the benchmark against which we compare the optimal mechanism. A symmetric equilibrium corresponds to a strategy $b : [0,1] \longrightarrow \mathbb{R}$ that is played by both generators, where $b(c)$ is the bid of a generator of type c.
The profit of a generator of type c that bids x is given by

$$\pi(c, x) = \int\limits_{C_{-i}} [p_i(x, b(c_{-i})) - c] q_i(x, b(c_{-i})) f_{-i}(c_{-i}) dc_{-i} \tag{6.10}$$

The maximized profit is

$$\overline{\pi}(c) = \max_x \int\limits_{C_{-i}} [p_i(x, b(c_{-i})) - c] q_i(x, b(c_{-i})) f_{-i}(c_{-i}) dc_{-i} \tag{6.11}$$

and an optimal strategy $\bar{b}(c)$ must satisfy:

$$\bar{b}(c) \in \text{argmax}_x \int\limits_{C_{-i}} [p_i(x, b(c_{-i})) - c] q_i(x, b(c_{-i})) f_{-i}(c_{-i}) dc_{-i} \tag{6.12}$$

6.4.2 Approximation

We now propose an approximation scheme to compute $\bar{b}(c)$. Without loss of generality, let's assume $C_i = [0, 1]$ (this is just a rescaling).

Doing a direct resolution by iterating on the strategies leads to numerical instability for small r. We introduce formally the equation $\partial_t b_t(c) = \partial_b \pi_{b_t}(c, b_t(c))$. The economical interpretation is clear: each agent changes his strategy proportionally to the gain he would get from small local changes. While a mathematical analysis of this equation and of the Nash equilibrium will soon be presented in another paper, we will assume here that the derivative could be understood as small variations for a given discretization when the derivative does not exist. Starting with $b(c) = c$, we observe numerically that the limit when t goes to infinity of b is a symmetric Nash equilibrium strategy. Some results are displayed in Figures 5.4, 6.5 and 6.5. An interesting comment is that the ISO expected cost in the optimal mechanism seems to depend linearly on r. We show in Section 6.8 that this is not the case from a mathematical standpoint, but still, this approximation is very close to the true function.

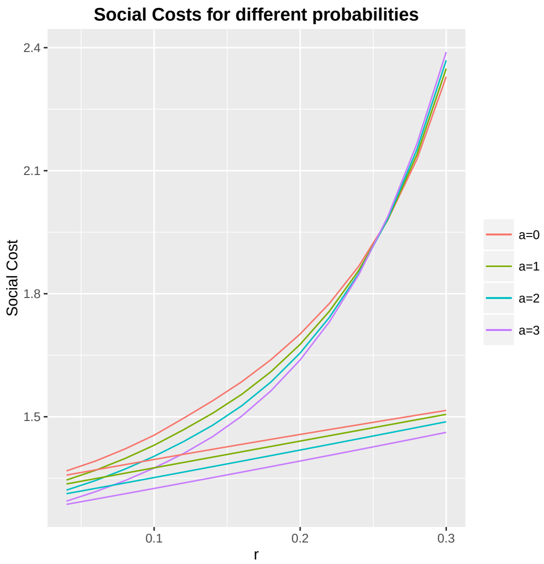


Figure 6.3 – The social costs for the standard mechanism and the optimal mechanism

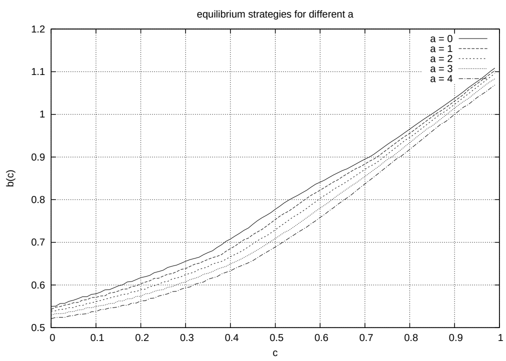


Figure 6.4 – The equilibrium strategies for different a

6.5 Some Comparisons

In order to compare both mechanisms in the presence of transmission costs we consider the following class of distributions in $[0, 1]$:

$$g_a(x) = \begin{cases} ax + (1 - \frac{a}{4}) & \text{if } x \leq \frac{1}{2} \\ -ax + (1 + \frac{3a}{4}) & \text{if } x \geq \frac{1}{2} \end{cases}$$

It is easy to verify that these distributions satisfy Assumption 4. The case of maximum variance corresponds to $a = 0$ (the uniform distribution) and $a = 4$ corresponds to the case of minimal variance. The results are display in Figures and .

As we can see, the optimal mechanism performs significantly better.

6.6 Practical Implementation

Now that we have seen that the optimal mechanism design performs significantly better, we turn to the problem of implementation. How difficult is it to implement it in practice? Is it significantly more difficult to implement than the standard mechanism?

Let's start with the simple case analyzed in this paper, that is a network with only two nodes. As long as the distribution of types satisfies Assumption 4, all that is required from the regulator is to solve the optimization problem (6.8) pointwise, for all possible realizations of (c_1, c_2). With this, he would state the rules of the game $(\hat{q}, \hat{x}, \hat{h})$. Once this is done, the agents will declare marginal costs c_i, and the regulator will evaluate the mechanism at the revealed costs, instructing each generator to produce $\hat{q}_i(c)$ and paying them $\hat{x}_i(c)$. In comparison, in the standard case this optimization is solved once for each time the agents declare some costs. Therefore, if the mechanism is to be used repeatedly over time, the optimal mechanism has a computational cost similar to the standard mechanism which is frequently used.

If we apply this technique to a more complicated network, like the ones that appear in practice, a result analogous to the one in Proposition 6.5 can be proved. Moreover, under Assumption 6, the pointwise solution of the problem would continue to be incentive compatible, since the quantity bought from a generator is decreasing in its own cost (keeping the other generators' costs fixed), which is sufficient for incentive compatibility (for details about this result in a general setting, see Correa and Figueroa [29]). Then, computing the optimal mechanism would require solving the ISO problem for modified costs (instead of actual ones), a problem of the same computational complexity as the one actually solved by the ISO.

6.7 Sensitivity Analysis

There is a natural concern. To compute the optimal mechanism, the regulator does not need to know the generators' costs, but he needs to have a prior about the *distribution* of these costs. This weaker requirement can still be difficult to satisfy in practice, and the regulator can make mistakes when assessing these prior distributions. This, coupled with the well documented fact that generators know each other very well, and much better than the regulator knows them, could give rise to unexpected and undesirable results. In what follows we consider a scenario where the regulator designs a mechanism with wrong information, and generators that know the true distributions of their competitors, plus the rules of the game.

We conduct the following sensitivity analysis. Let's suppose that the true distributions are f_1, f_2, but that the regulator mistakenly designs the optimal mechanism $(\hat{q}^{\bar{f}}, \hat{h}^{\bar{f}}, \hat{x}^{\bar{f}})$ for distribution $\bar{f}_1$, $\bar{f}_2$. We compare the expected payment that the seller was hoping to pay (for distributions f_1, f_2) with the expected payment that the seller actually pays (for distributions $\bar{f}_1, \bar{f}_2$).

First, we note that the mechanism $(\hat{q}^{\bar{f}}, \hat{h}^{\bar{f}}, \hat{x}^{\bar{f}})$ (designed with $\bar{f}_1$, $\bar{f}_2$ in mind) is incentive compatible in *dominant strategies* since the payment defined in (6.9) satisfies (6.4) pointwise. Therefore the agent has incentives to tell the truth regardless of his beliefs about the other agent's distribution. This is particularly important since the "wrong" mechanism does not induce, as a second-order effect, players to lie about their costs, no matter how mistaken the regulator was in his assessment of the prior distribution f. The cost paid by a regulator who thinks the distributions are $\bar{f}_1, \bar{f}_2$ but is really facing distributions f_1, f_2 is given by

$$C^{\bar{f},f} = \int_C \sum_{i=1,2} \hat{q}_i^{\bar{f}}(c)[c_i + \frac{F_i(c_i)}{f_i(c_i)}]f(c)dc$$

6.8 The social cost is not affine

We derive an explicite expression of the expected social cost for the optimal mechanism. We consider the symmetric binodal setting with the common knowledge density f being the uniform distribution on $[0,1]$. For r in $[0,1/2d]$, we denote by $G(r)$ the social cost associated with the optimal mechanism. By definition of f, the virtual cost writes (by symmetry, the expression is the same for producer 1 and producer 2): $J(c) = 2c$. We will use the shorthand $\tilde{F}(c_1, c_2) = F(J(c_1), J(c_2)) = F(2c_1, 2c_2)$. Observe that, $\tilde{F}(c_1, c_2) = F(c_1, c_2)$. The expected social cost (i.e. what is paid by the ISO) $G(r)$ can be expressed as an integral over $(c_1, c_2) \in [0,1]^2$:

$$G(r) = \mathbb{E}(x_1 + x_2) = \int_{c_1=0}^{1} \int_{c_2=0}^{1} \{J(c_1)\tilde{q}(c_1, c_2) + J(c_2)\tilde{q}(c_2, c_1)\} \mathrm{d}c_2 \mathrm{d}c_1 \tag{6.13}$$

where $\tilde{q}$ is the optimal mechanism allocation rule (which depends on r). Note that by symmetry, the participation of the part such that $c_1 < c_2$ is equal to the participation of the part such that $c_2 \leq c_1$ and those two parts form a partition of $(c_1, c_2) \in [0,1]^2$. Therefore,

$$G(r) = 2 \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} \{J(c_1)\tilde{q}(c_1, c_2) + J(c_2)\tilde{q}(c_2, c_1)\} \mathrm{d}c_2 \mathrm{d}c_1, \tag{6.14}$$

then we decompose the integral in two terms: the integral over the locus such that $\tilde{F}(c_1, c_2) \geq 0$ and the integral over the locus such that $\tilde{F}(c_1, c_2) < 0$. Observe that since $c_1 > c_2$, $\tilde{F}(c_2, c_1) \geq 0$.

$$\begin{aligned} G(r) &= 2 \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} \{J(c_1)\tilde{F}(c_1, c_2) + J(c_2)\tilde{F}(c_2, c_1)\} 1_{\tilde{F}(c_1,c_2)\geq 0} \mathrm{d}c_2 \mathrm{d}c_1 && (6.15) \\ &\quad +2 \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} \bar{q} J(c_2) 1_{\tilde{F}(c_1,c_2)<0} \mathrm{d}c_2 \mathrm{d}c_1 && (6.16) \\ &= 4 \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} \{c_1 F(c_1, c_2) + c_2 F(c_2, c_1)\} 1_{F(c_1,c_2)\geq 0} \mathrm{d}c_2 \mathrm{d}c_1 && (6.17) \\ &\quad +4 \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} \bar{q} c_2 1_{F(c_1,c_2)<0} \mathrm{d}c_2 \mathrm{d}c_1. && (6.18) \end{aligned}$$

Therefore

$$G(r) = 4A(r) + 4\bar{q}B(r) \tag{6.19}$$

with

$$A(r) = \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} \{c_1 F(c_1, c_2) + c_2 F(c_2, c_1)\} 1_{F(c_1,c_2)\geq 0} \mathrm{d}c_2 \mathrm{d}c_1 \tag{6.20}$$

and

$$B(r) = \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} c_2 1_{F(c_1,c_2)<0} \mathrm{d}c_2 \mathrm{d}c_1. \tag{6.21}$$

Note that the dependence in r is implicite in F and $\bar{q}$. We denote by Y the function $Y(c_1, c_2) = \frac{c_1 - c_2}{c_1 + c_2}$ and by P the polynomial $P(Y) = d + \frac{1}{2r}Y^2 - \frac{1}{r}Y$. We introduce

$y_0(r) = 1 - \sqrt{1-2rd}$ the unique solution of

$$P(y) = 0; \quad 0 \le y \le 1. \tag{6.22}$$

Now we observe that for any $(c_1, c_2) \in [0,1]^2$ such that $c_1 \ge c_2$:

$$F(c_1,c_2) \ge 0 \iff P(Y(c_1,c_2)) \ge 0 \iff Y \le y_0(r) \tag{6.23}$$

and

$$F(c_1,c_2) \le 0 \iff Y(c_1,c_2) \ge y_0(r). \tag{6.24}$$

Therefore $A(r)$ rewrites

$$\int_{c_1=0}^{1}\int_{c_2=0}^{c_1} \{c_1 F(c_1,c_2) + c_2 F(c_2,c_1)\} 1_{F(c_1,c_2)\ge 0} \mathrm{d}c_2 \mathrm{d}c_1 \tag{6.25}$$

$$= \int_{c_1=0}^{1}\int_{c_2=0}^{c_1} \{c_1 P(Y(c_1,c_2)) + c_2 P(Y(c_2,c_1))\} 1_{Y(c_1,c_2)\le y_0(r)} \mathrm{d}c_2 \mathrm{d}c_1 \tag{6.26}$$

$$= \int_{c_1=0}^{1}\int_{Y=1}^{0} \{c_1 P(Y) + c_1 \frac{1-Y}{1+Y} P(-Y)\} 1_{Y\le y_0(r)} \frac{-2c_1}{(1+Y)^2} \mathrm{d}Y \mathrm{d}c_1 \tag{6.27}$$

$$= 2\int_{c_1=0}^{1} c_1^2 \mathrm{d}c_1 \int_{Y=0}^{y_0(r)} \{P(Y) + \frac{1-Y}{1+Y} P(-Y)\} \frac{1}{(1+Y)^2} \mathrm{d}Y \tag{6.28}$$

$$= \frac{2}{3r}\int_{Y=0}^{y_0(r)} \frac{2dr - Y^2}{(1+Y)^3} \mathrm{d}Y = \frac{2}{3r}\int_{Y=0}^{y_0(r)} \frac{-1}{1+Y} + \frac{2}{(1+Y)^2} + \frac{2rd-1}{(1+Y)^3} \mathrm{d}Y \tag{6.29}$$

Here are some justifications:

1. Definitions of y_0, Y and P,
2. Change of variable $Y = (c_1 - c_2)/(c_1 + c_2) \iff c_2 = c_1(1-Y)/(1+Y)$,
3. Simplification and Fubini theorem,
4. Simplification
5. Check that $\frac{-1}{1+Y} + \frac{2}{(1+Y)^2} + \frac{2rd-1}{(1+Y)^3} = \frac{2dr-Y^2}{(1+Y)^3}$

Thus

$$A(r) = \frac{2}{3r}\left(\frac{2rd-1}{2}\frac{y_0(r)^2 + 2y_0(r)}{(1+y_0(r))^2} + \frac{2y_0(r)}{1+y_0(r)} - \ln(1+y_0(r))\right). \tag{6.30}$$

Then follows similarly the computation of $B(r)$:

$$B(r) = \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} c_2 1_{F(c_1,c_2)} \tag{6.31}$$

$$= \int_{c_1=0}^{1} \int_{c_2=0}^{c_1} c_2 1_{Y(c_1,c_2)>y_0(r)} \mathrm{d}c_2 \mathrm{d}c_1 \tag{6.32}$$

$$= -\int_{c_1=0}^{1} \int_{Y=1}^{0} c_1 \frac{1-Y}{1+Y} 1_{Y>y_0(r)} c_1 \frac{2}{(1+Y)^2} \mathrm{d}Y \mathrm{d}c_1 \tag{6.33}$$

$$= 2 \int_{c_1=0}^{1} c_1^2 \mathrm{d}c_1 \int_{Y=y_0(r)}^{1} \frac{1-Y}{(1+Y)^3} \mathrm{d}Y \tag{6.34}$$

$$= \frac{2}{3} \int_{Y=y_0(r)}^{1} \frac{2}{(1+Y)^3} - \frac{1}{(1+Y)^2} \mathrm{d}Y \tag{6.35}$$

$$= \frac{2}{3} \left(\frac{1}{(1+y_0(r))^2} - \frac{1}{4} - \frac{1}{(1+y_0(r))} + \frac{1}{2} \right) \tag{6.36}$$

$$= \frac{1}{6} \left(\frac{1-y_0(r)}{1+y_0(r)} \right)^2 \tag{6.37}$$

Note that $\bar{q} = \frac{2y_0}{r}$ We conclude:

$$G(r) = \frac{8}{3r} \left(\frac{2rd-1}{2} \frac{y_0(r)^2 + 2y_0(r)}{(1+y_0(r))^2} + \frac{2y_0(r)}{1+y_0(r)} - \ln(1+y_0(r)) \right) + \frac{4y_0(r)}{3r} \left(\frac{1-y_0(r)}{1+y_0(r)} \right)^2 \tag{6.38}$$

Because of the logarithm, G is not linear in r. We set $d=1$ for the plot displayed in Figure 6.5.

6.9 Appendix

Proof of lemma 6.3

If (q,h,x) is feasible, then

$$U_i(c_i,c_i) \geq U_i(c_i,c_i') \text{ and } U_i(c_i',c_i') \geq U_i(c_i',c_i) \tag{6.39}$$

implies

$$(Q_i(c_i) - Q_i(c_i'))(c_i - c_i') \leq 0 \tag{6.40}$$

which means that Q_i is non-increasing. We get (6.4) with the envelop theorem.
Conversely if (q,h,x) satisfies (6.3) to (6.6), then

$$U_i(c_i,c_i) - U_i(c_i,c_i') = X(c_i) - X(c_i') + c_i(Q_i(c_i') - Q_i(c_i)) \tag{6.41}$$

$$= V(c_i) + c_i Q(c_i) - (V(c_i') + c_i' Q(c_i')) + c_i(Q_i(c_i') - Q_i(c_i)) \tag{6.42}$$

$$= \int_{c_i}^{c_i'} Q_i(s)\mathrm{d}s - (c_i' Q(c_i') - c_i(Q_i(c_i'))) = \int_{c_i}^{c_i'} Q_i(s) - Q(c_i')\mathrm{d}s, \tag{6.43}$$

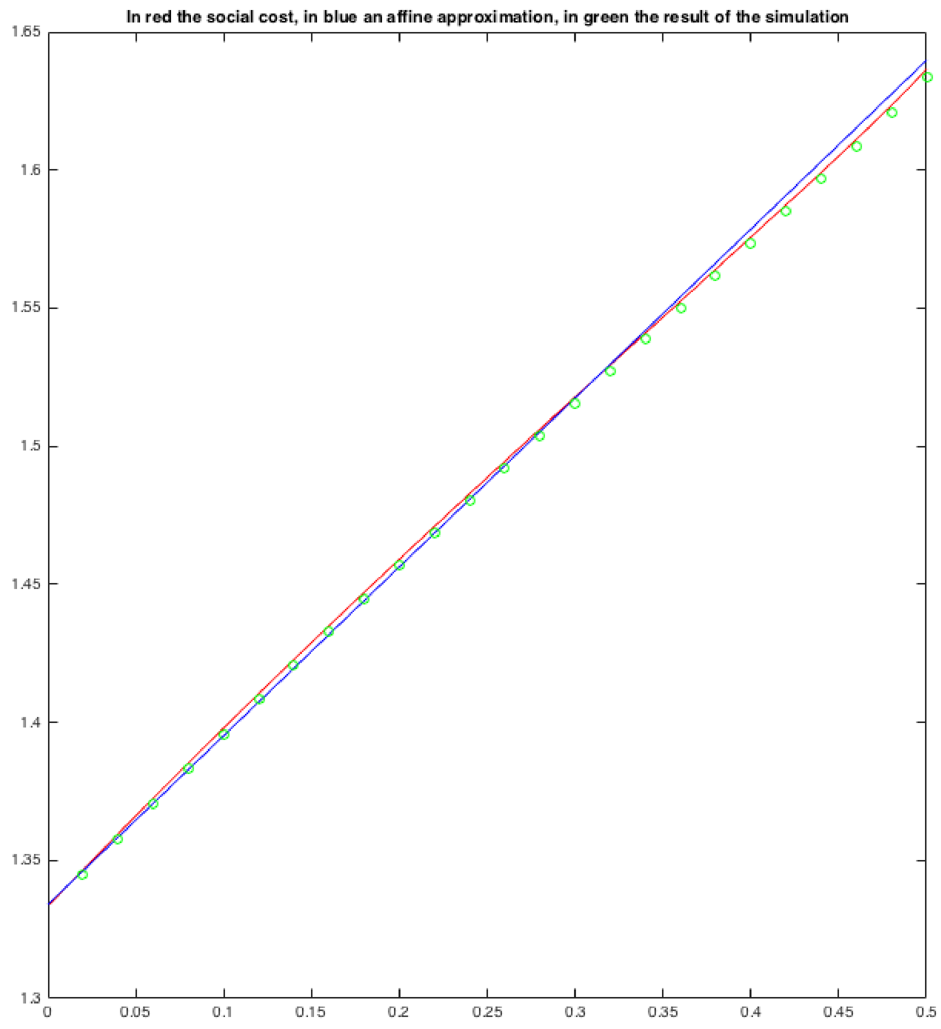


Figure 6.5 – The social cost (red) is well approximated by an affine function (blue).

which is non-negative since Q_i is non-increasing. Then we get the voluntary participation constraint combining (6.4) with $V_i(\overline{c}_i) \geq 0$.

Proof of lemma 6.4
From (6.2) we get that

$$\int\limits_{C} x_i(c)f(c)dc = \int\limits_{C} c_i q_i(c_i, c_{-i})f(c)dc + \int\limits_{C_i} V_i(c_i)dc_i,$$

The second term on the right-hand side can be written (by using (6.4) and changing the order of integration) as:

$$\begin{aligned}
\int\limits_{C_i} V_i(c_i)dc_i &= \int\limits_{C_i} [V_i(\overline{c}_i) + \int\limits_{c_i}^{\overline{c}_i} Q_i(s_i)ds_i] f_i(c_i)dc_i \\
&= V_i(\overline{c}_i) + \int\limits_{C_i} Q_i(s_i)[\int\limits_{\underline{c}_i}^{s_i} f_i(c_i)dc_i]ds_i \\
&= V_i(\overline{c}_i) + \int\limits_{C_i} Q_i(c_i)F_i(c_i)dc_i \\
&= V_i(\overline{c}_i) + \int\limits_{C_i}\int\limits_{C_{-i}} q_i(c_i, c_{-i})F_i(c_i)dc_i \\
&= V_i(\overline{c}_i) + \int\limits_{C} q_i(c)\frac{F_i(c_i)}{f_i(c_i)}f(c)dc
\end{aligned}$$

Replacing this last expression and noticing that in any optimal mechanism $V_i(\overline{c}_i) = 0$ the result follows.

The Solution to the pointwise optimization problem

For given c_1 and c_2, we analyze the seller's cost minimization problem, where he faces modified costs $L_1(c_1)$ and $L_2(c_2)$. In the case of section 3, $L_i(c_i) = J_i(c_i)$, while in section 4, $L_i(c_i) = c_i$.

$$\begin{array}{ll}
\min\limits_{q,h} & \sum\limits_{i=1}^{2} L_i(c_i)q_i \\
s.t. & q_i - h_i + h_{-i} \geq \frac{r}{2}[h_1^2 + h_2^2] + d \text{ for } i = 1,2 \\
& q_i, h_i \geq 0 \text{ for } i = 1,2
\end{array}$$

Without loss of generality we consider $c_1 \leq c_2$ (we can just change indices otherwise). In that case we get $h_2 = 0$, and the constraints $q_1 \geq 0, h_1 \geq 0$ never bind. The problem then becomes

$$
\begin{array}{ll}
\min\limits_{q,h} & \sum\limits_{i=1}^{2} L_i(c_i)q_i \\
s.t. & q_1 - h_1 \geq \frac{r}{2}[h_1^2 + h_2^2] + d \\
 & q_2 + h_1 \geq \frac{r}{2}[h_1^2 + h_2^2] + d \\
 & q_i, h_i \geq 0 \text{ for } i = 1, 2
\end{array}
$$

Writing the Lagrangian and taking FOC we get:

$$
\begin{aligned}
-L_1(c_1) + \lambda_1 &= 0 && (6.44) \\
-L_2(c_2) + \lambda_2 + \mu &= 0 && (6.45) \\
-\lambda_1 - \lambda_1 r h_1 + \lambda_2 - \lambda_2 r h_1 &= 0 && (6.46) \\
-q_1 + h_1 + d + \frac{r}{2}h_1^2 &\leq 0 && (6.47) \\
-q_2 - h_1 + d + \frac{r}{2}h_1^2 &\leq 0 && (6.48) \\
-q_2 &\leq 0 && (6.49) \\
\mu, \lambda_1, \lambda_2 &\geq 0 && (6.50)
\end{aligned}
$$

The solution can be broadly divided into two cases, depending on whether the positivity constraint $q_2 \geq 0$ binds or not.

Case 1: $d - \frac{L_2(c_2)-L_1(c_1)}{r(L_2(c_2)+L_1(c_1))} + \frac{(L_2(c_2)-L_1(c_1))^2}{2r(L_2(c_2)+L_1(c_1))^2} \geq 0$

This case corresponds to the non-binding case, and the solution is given by

$$
\begin{aligned}
\mu &= 0 \\
\lambda_i &= L_i(c_i) \\
q_1 &= d + \frac{L_2(c_2) - L_1(c_1)}{r(L_2(c_2) + L_1(c_1))} + \frac{(L_2(c_2) - L_1(c_1))^2}{2r(L_2(c_2) + L_1(c_1))^2} \\
q_2 &= d - \frac{L_2(c_2) - L_1(c_1)}{r(L_2(c_2) + L_1(c_1))} + \frac{(L_2(c_2) - L_1(c_1))^2}{2r(L_2(c_2) + L_1(c_1))^2} \\
h_1 &= \frac{L_2(c_2) - L_1(c_1)}{r(L_2(c_2) + L_1(c_1))}
\end{aligned}
$$

Case 2: $d - \frac{L_2(c_2)-L_1(c_1)}{r(L_2(c_2)+L_1(c_1))} + \frac{(L_2(c_2)-L_1(c_1))^2}{2r(L_2(c_2)+L_1(c_1))^2} < 0$

This is the binding case (it can only happen if $2dr < 1$), and the solution is:

$$
\begin{aligned}
\mu &= L_1(c_1) + L_2(c_2) - \frac{2L_1(c_1)}{\sqrt{1 - 2dr}} \\
\lambda_1 &= L_1(c_1)
\end{aligned}
$$

$$
\begin{aligned}
\lambda_2 &= \left[\frac{2-\sqrt{1-2dr}}{\sqrt{1-2dr}}\right] L_1(c_1) \\
q_1 &= 2\left[\frac{1-\sqrt{1-2dr}}{r}\right] \\
q_2 &= 0 \\
h_1 &= \frac{L_2(c_2)-L_1(c_1)}{r(L_2(c_2)+L_1(c_1))}
\end{aligned}
$$

Chapter 7

Mechanism design and allocation algorithms for network markets with piece-wise linear costs externalities and externalities

This is a joint work with Alejandro Jofré.

Contents

Motivated by market power in electricity market we introduce a mechanism design in [38] for simplified markets of two agents with linear production cost functions. In standard procurement auctions, the market power resulting from the quadratic transmission losses allow the producers to bid above their true value (i.e. production cost). The mechanism proposed in the previous paper reduces the producers margin to the society benefit. We extend those results to a more general market made of a finite number of agents with piecewise linear cost functions, which make the problem more difficult, but at the same time more realistic. We show that the methodology works for a large class of externalities. We also provide two algorithms to solve the principal allocation problem.

7.1 Introduction

Our purpose is to show how monopolistic behaviors in network markets can be opposed using mechanism design. We point out that the optimal mechanism we obtain has a surprisingly simple expression. We complete this work with algorithmic tools for the computation of this mechanism. Following a model proposal already discussed in [35, 36, 38], we consider a geographically extended market where a divisible good is traded. Each market participant is located on a node of a graph, and the nodes are connected by edges. The good can travel from one node to another through those edges at the cost of a quadratic loss. We will use the word principal to designate what could also be called in the literature a central operator, or in the context of electricity markets, an ISO. This principal, who aggregates the (inelastic) demand side, has to match locally -i.e. at each node - production and demand at the lowest expense through a procurement auction. As argued in [38] this setting is relevant to describe some real electricity markets, but it could also be used in other markets where a good is transported. There is a clear antagonism between the market participants: the operator wants to minimize its expected cost while the producers want to maximize their expected profit. So there is a transaction and a commitment between each agent and the principal, and at the same time, there is a competition among the agents. In a standard procurement auction, the market power resulting from the quadratic line losses allow the producers to bid above their true value (i.e. production cost) [35]. The mechanism reduces the producers margin and decrease the social cost represented in this case by the optimal value of the principal. The optimal auction design was introduced by Myerson in 1981 [70]. We build on an electricity market model introduced by the second author in two previous papers [36] and [35]. The authors wrote a brief presentation of this model in [50]. Other models were proposed for example in [8], [2], and [52], with a focus on the existence of a market equilibrium. Concerning the techniques we use in this paper the reader can refer to [61], [72], [7] chapter 45 and [87] for general introductions on principal agent theory, mechanism design, game theory and lattices theory respectively.

We consider -similarly to [38]- that everybody knows the demand at each node before the interactions start and that the production cost of each agent is private information. In a standard setting the agents first bid their cost and then the principal, knowing the bids, a posteriori minimizes its cost. So in a standard setting the principal is a bid taker. The producers know they influence the allocation and compete with each other to maximize their individual profit. Since the demand is known by everyone, everyone can guess the principal reaction once the bids have been announced: we can virtually remove the principal from the interaction in the standard setting and consider that the agents are the players of a game with incomplete information (since the agents do not know their fellow agents preferences). This equivalence is true provided that the agents are not communicating with each others. The mechanism design consists in changing the payoff function of this game -subject to constraints we detail in this article- so as to minimize a priori (i.e. before the bids are announced) the principal cost. Allowing the principal to strike first by revealing a committing rule gives him a strategic advantage in the negotiation.

We restrict our discussion to determinist demand, but the reasoning extends naturally to random demand as long as any possible realization of the demand satisfies the model assumptions. Indeed since the optimal mechanism constructed in this article is incentive compatible, then a random version (where the demand is revealed after the producers bidding phase, as in [36]) would be realization-wise incentive compatible, and so incentive compatible. Observe the mechanism we propose in the following could be adapted to elastic, piecewise linear demand.

Our first main result is the mechanism design characterization. Interestingly the allocation procedures for the optimal and the standard mechanism are the same (one just needs to modify the input of the allocation procedure of the standard mechanism to get the allocation of the optimal mechanism). Our second main result is a principal allocation algorithm based on a fixed point. The fixed point could be interpreted as cooperating agents trying to minimize a global criteria by sharing relevant information. Our implementation of the algorithm gives good results against standard methods. We point out that the numerical computation of Nash equilibrium for the procurement auction (important to compare the optimal mechanism and the standard auction setting) requires an efficient algorithm to compute the allocation. Some other additional facts are presented within the paper: the smoothness of the allocation functions (q and Q), a decreasing rate estimation for the fixed point iterations, some results of numerical experiments with the fixed point algorithm, and a specific algorithm for the two-agent case.

We describe the market in the next section. In §7.3 we introduce and solve the mechanism design problem. In §7.4, we study the standard allocation problem and propose an algorithm to solve it. In §7.5 we propose a different algorithm for the 2-agent standard allocation problem. In §7.6 we sum up and comment the main results and propose some continuations for this work. A reader only interested in mechanism design could read §7.2, §7.3 and §7.4 only, whereas readers interested only in allocation algorithms could concentrate on §7.4 and §7.5.

7.2 Market description

The production cost of each agent is assumed to be piecewise linear, non decreasing and convex in the quantity produced. This class of functions is sufficiently rich to represent real life problems and sufficiently simple for theoretical study. In this work we need to assume that the production levels at which there is a slope change are known in advance and exogenous (i.e. the agents cannot choose them). Then without loss of generality we assume that there is a quantity $\bar{q}$ such that the changes of slope only occur at the multiples of $\bar{q}$. Thus, the authors find it practical to write the production cost functions in the form

$$C^c(q) = \sum_{j=1}^{N} c_j \min((q-(j-1)\bar{q})^+, \bar{q}), \tag{7.1}$$

where $N \in \mathbb{N}$ and the c_j are some slopes coefficients specific to the agent, while q is the quantity produced. We will sometimes refer to the vector of the c_j as the cost vector (of the agent). If we denote by q_i^j the quantity produced by agent i at marginal cost c_i^j, then $q_i^j = \min((q_i - (j-1)\bar{q})^+, \bar{q})$, where q_i is the total quantity produced by this agent. Let $c_* < c^* \in \mathbb{R}^{*+}$ and $\boldsymbol{C}$ a set of non-decreasing N-tuples of $[c_*, c^*]$. To each element c of $\boldsymbol{C}$ we associate the piecewise linear cost function $q \to C^c(q)$. Throughout the paper we set, for any $c \in \boldsymbol{C}$, $c^{N+1} = c^*$ to simplify notations in some proofs. Note that in practice a capacity constraint of the type $q \leq j\bar{q}$ for a given agent can be implemented by setting its $(j+1)^{th}$ slope c_{j+1} equal to a big positive number. If an agent of cost vector c produces a quantity q and receives a transfer x, then its profit is

$$u_i = x - C^c(q). \tag{7.2}$$

There are n agents numbered from 1 to n in the market. We denote $I = [1 \ldots n]$ and use generically the letter i to refer to a specific agent, and $-i$ to refer to $I \backslash \{i\}$. We denote $J = [1 \ldots N]$ and we will use generically j for the cost coefficients of the jth segment (starting from 1). The agents are dispatched on the n nodes of a graph. At each node i we find the corresponding agent i and a local demand d_i. The nodes are connected by undirected edges. We write $V(i)$ the set of nodes different from i connected to i. Obviously if $i_1 \in V(i_2)$ then $i_2 \in V(i_1)$. We denote $E = \{(i_1, i_2) : i_1 \in V(i_2)\}$ the set of undirected edges. For each $(i_1, i_2) \in E$, we introduce a quadratic loss coefficient r_{i_1,i_2} such that $r_{i_1,i_2} = r_{i_2,i_1}$. In the context of electricity market, this quadratic coefficient corresponds to the Joule effect within the lines. We make the non restricting assumption that N is big enough so that in what follows production at each node is smaller than $\bar{q}N$.

We assume that both the agents and the principal are risk neutral: they maximize their expected profit. If the principal proposes to pay a price x_i to agent i to make her produce a quantity q_i - this agent being free to accept or decline the offer- and if the agent i has a production cost defined by c_i, then she accepts the offer if

$$x_i - C^{c_i}(q_i) \geq 0. \tag{7.3}$$

So for agent i, either $x_i \geq C^{c_i}(q_i)$ or $q_i = 0$. Thus, if the principal knew the cost vectors c_i, he would solve an allocation problem with those c_i, and then bid to the agents the quantity and the payments corresponding to the solution of the allocation problem. But the principal does not know the cost vectors, so instead what happens is that the agents tell her some values for the c_i (not necessary their real cost vectors), and then the principal decides based on those values. In this case, previous works [35] showed that the agents can get non-zero profits and bid above their production costs. The question we adress is how to reduce their margins.

To do so, we need to consider an intermediate scenario between the one in which the agent knows nothing (and is a price taker), and the one in which he knows everything (and optimizes directly the whole system as a global optimizer). Each agent is characterized by an element f_i, which is a probability density of support included in $\boldsymbol{C}$ and an element c_i of $\boldsymbol{C}$ drawn according to f_i. Only agent i knows c_i, which is private information. The other agents and the principal only know the probability f_i with which it was drawn. The density f_i corresponds to the public knowledge on agent i production costs so the principal won't accept any bid c_i that is not in the support of f_i. We assume that the cost slopes are not correlated for a given agent and between agents, i.e. their laws f_i^j are independent. In particular $f_i(c_i) = \prod_{j \in J} f_i^j(c_i^j)$. In such situation, it makes sense to define

$$f_{-i}(c_{-i}) = \prod_{i' \in I \setminus i} f_{i'}(c_{i'}) \quad \text{and} \quad f(c_1, .., c_n) = \prod_{i \in I} f_{i'}(c_i), \tag{7.4}$$

and $\mathbb{E}$ (respectively $\mathbb{E}_{c_{-i}}$) the mean operator with respect to f (respectively f_{-i}). The density f (resp. f_{-i}) represents the uncertainty from the principal (resp. agent i) perspective. To simplify notations we will use the symbole $\boldsymbol{C}^n$ to denote the product of the supports of the f_i. We denote by $\mathcal{Q}$ the set of allocation functions -which are the applications from $\boldsymbol{C}^n$ to $\mathbb{R}_+^n$, by $\mathbb{X}$ the set of payments functions -which are the applications from $\boldsymbol{C}^n$ to $\mathbb{R}^n$, and by $\mathbb{H}$ the set of flow functions - which are the applications from $\boldsymbol{C}^n$ to $\mathbb{R}^E$. A *direct* *mechanism* is a triple $(q, x, h) \in (\mathcal{Q}, \mathbb{X}, \mathbb{H})$. Let $(q, x) \in (\mathcal{Q}, \mathbb{X})$. For this payment function and this allocation function, the expected profit of agent i of type c_i and bid c_i' is

$$U_i(c_i, c_i') = \mathbb{E}_{-i} u_i = X_i(c_i') - \sum_{j \in J} c_i^j Q_i^j(c_i'). \tag{7.5}$$

where the capitalized quantities

$$Q_i^j(c_i) = \mathbb{E}_{-i} \min((q_i(c_i, c_{-i}) - (j-1)\bar{q})^+, \bar{q}) \quad \text{and} \quad X_i(c_i) = \mathbb{E}_{-i} x_i(c_i, c_{-i}) \tag{7.6}$$

correspond to the average of their non capitalized counterpart. We also denote by

$$V_i(c_i) = U_i(c_i, c_i). \tag{7.7}$$

the expected profit of agent i if she is of type c_i and bids her true production cost.

In this work we make five assumptions.

- First, the *non overlapping working zones assumption* is that if we denote by $\boldsymbol{C}_i$ the support of f_i, then $\boldsymbol{C}_i$ should be of the form:

$$\boldsymbol{C}_i = [c_i^{1-}, c_i^{1+}] \times \ldots \times [c_i^{N-}, c_i^{N+}] \tag{7.8}$$

  with $c_i^{1-} < c_i^{1+} < \ldots < c_i^{N-} < c_i^{N+}$. We could interprete each segments over which the agent has a constant marginal cost as a working zone with identified productive assets. The expertise of the market participants should allow them to, based on the working zone, assess the marginal cost of the agent. This makes senses for instance if the setting is repeated over time. This estimation need to be precise enough so that there is no chance that it corresponds to another working zone. We use this item in particular in the proof of lemma 7.6.

- For $i \in I$, $j \in J$ and $c_i \in \boldsymbol{C}_i$ let

$$K_i^j(c_i) = \frac{\int_{c_i^{j-}}^{c_i^j} f_i(c_i^{-j}, s) ds}{f_i(c_i)}. \tag{7.9}$$

  We point out that by independence of the laws of the c_i^j, $K_i^j(c_i) = \int_{c_i^{j-}}^{c_i^j} f_i^j(s) ds / f_i^j(c_i^j) = K_i^j(c_i^j)$. So K_i^j is simply the ratio of the cumulative distribution and the probability density for c_i^j. The second assumption is the *discernability assumption.* For all $i \in I$ and $c_i \in \boldsymbol{C}_i$, the virtual cost $J_{i,j}(c_i^j) = c_i^j + K_i^j(c_i^j)$ is increasing in j. As demonstrated in the next section, the virtual cost could be interpreted as the real marginal cost augmented by a marginal information rent. The item imposes the marginal information rent to be such that for any bid, the virtual marginal prices are increasing, i.e. the virtual production cost function is convex. The item is necessary to show the independence property of the reformulation in Lemmas 7.8 and 7.9.

- Third, in the following we assume that, for all $j \in J$, $i \in \boldsymbol{I}$ and $c_i \in \boldsymbol{C}_i$,

$$c_i^j \to c_i^j + K_j^i(c_i^j) \tag{7.10}$$

is increasing in c_i^j. This is the piecewise linear adaptation of the classic *monotone likelihood ratio property assumption* encountered in mechanism design [70, 29]. It is true in particular for log-concave functions. The assumption ensures that the pointwise allocation resulting from the mechanism design problem reformulation is decreasing in the bids. We refer to this assumption in the proof of Theorem 7.11.

- Fourth, for §7.3 and §7.4 only, we assume that

$$d_i - \sum_{i' \in V(i)} \frac{1}{2r_{i,i'}} < 0, \quad \text{and} \quad d_i + \sum_{i' \in V(i)} \frac{3}{2r_{i,i'}} > N\bar{q}, \tag{7.11}$$

i.e. we require the $r_{i,j}$ to be small enough. Note that the bigger the demand, the smaller the r should be, which is a limit to the generality of the approach. This assumption ensures that, for any agent i and working zone k, no matter what the other agents are doing, it is still possible to find a (virtual) marginal price that would ensure a production of exactly $k\bar{q}$ in an optimal allocation. If the loss rates $r_{ii'}$ are all too big for a given agent i, then the line losses can be bigger than the flow through the lines: the lines of agent i can be all saturated. This hypothesis is necessary to ensure the existence of one of the building block of the fixed point operator presented in §7.4. We point out that this is the multidimensional version of the assumption $1 - 2rd \geq 0$ in [38].

- Fifth, for regularity issues we make the non restrictive assumption that it is not possible to produce a multiple of $\bar{q}$ at each node and satisfy exactly the nodal constraints. This is non restrictive because if this was the case we could perturb the demand to ensure the condition is satisfied. This hypothesis will be important in the proof of the regularity of q (in lemma 7.16), from which the regularity of Q follows.

To finish with the market presentation, we introduce the products of the type sets $\boldsymbol{C}^n = \prod_{i \in I} \boldsymbol{C}^{i'}$ and $\boldsymbol{C}^{-i} = \prod_{i' \in I \setminus \{i\}} \boldsymbol{C}^{i'}$.

7.3 Mechanism Design

We start with the revelation principle as expressed in [40].

Theorem 7.1 (Revelation Principle)**.** *To any Bayesian Nash equilibrium of a game of incomplete information, there exists a payoff-equivalent direct revelation mechanism that has an equilibrium where the players truthfully report their types.*

According to the revelation principle, we can look for direct truthful mechanisms. There is a priori no reason why the agents should willingly report their types. So we need to add a constraint on the design to enforce truthfulness. This means that the profit of any agent i of type c_i should be maximal when agent i bids her true type c_i i.e. for all (c_i', c_i)

$$U_i(c_i, c_i) \geq U_i(c_i, c_i'). \quad (IC) \tag{7.12}$$

This is the incentive compatibility *(IC)* constraint. In addition, since we want all agents to participate in the market, we need the *participation constraint* imposing that for all c_i

$$U_i(c_i, c_i) \geq 0. \quad (PC) \tag{7.13}$$

Without this constraint, the principal would optimize as if the agents would accept any deal (even deals where they would make a negative profit). The last constraint is that the supply should be at least equal to the demand at every node. The supply available at a given node is equal to the production augmented by the imports minus the exports and the line losses. As explained earlier, there is a loss when some quantity $h_{i,i'}$ of the divisible good is sent from one node i to another i'. This loss is equal to $r_{i,i'}h_{i,i'}^2$, where $r_{i,i'}$ is a multiplicative constant. In order to obtain symmetric expressions, we will proceed as if half of this quantity was lost by the sender, and the other half by the receiver (see for instance [35]). Note that we could have equivalently used signed flows, but we would have lost some symmetry in the formulation. Then the *supply and demand constraint* writes, for all $i \in I$ and $c \in \boldsymbol{C}^n$,

$$q_i(c) + \sum_{i' \in V(i)} h_{i',i}(c) - h_{i,i'}(c) - \frac{h_{i,i'}^2(c) + h_{i',i}^2(c)}{2} r_{i,i'} \geq d_i. \quad (SD) \tag{7.14}$$

We point out that for an optimal allocation (see §7.4) , $h_{i,i'}h_{i',i} = 0$.

The principal decision is a triple $(q, x, h) \in (\mathcal{Q}, \mathbb{X}, \mathbb{H})$. This decision is made under the constraints (IC), (PC) and (SD). Since we assume that the principal is risk neutral, his goal is to minimize his average cost, i.e. mathematically his criterion is equal to the average of the sum of the payments. Finally the optimal mechanism is the solution of

Problem 1.

$$\underset{(q,x,h)\in(\mathcal{Q},\mathbb{X},\mathbb{H})}{minimize} \sum_{i\in I} \mathbb{E}x_i(c)$$

subject to

$$\forall c \in \boldsymbol{C}^n, \forall i \in I : \quad q_i(c) + \sum_{i'\in V(i)} h_{i',i}(c) - h_{i,i'}(c) - \frac{h^2_{i,i'}(c) + h^2_{i',i}(c)}{2} r_{i,i'} \geq d_i \ (SD)$$

$$\forall c \in \boldsymbol{C}^n, \forall (i,i') \in E : \quad h_{i,i'}(c) \geq 0$$

$$\forall i \in I, \forall (c'_i, c_i) \in \boldsymbol{C}^2_i : \quad U_i(c_i, c_i) \geq U_i(c_i, c'_i) \ (IC)$$

$$\forall i \in I, \forall c_i \in \boldsymbol{C}_i : \quad U_i(c_i, c_i) \geq 0 \quad (PC).$$

We now proceed to solve the optimal mechanism design problem, which is a functional optimization problem with an infinity of constraints, some of which are expressed with integrals. The essential observation is that this complicated problem is equivalent to a much simpler one. The proof relies on the comparison with two intermediate problems:

Problem 2.

$$\underset{(q,x,h)\in(\mathcal{Q},\mathbb{X},\mathbb{H})}{minimize} \sum_{i\in I} \mathbb{E}x_i(c)$$

subject to.

$$\forall c \in \boldsymbol{C}^n, \forall i \in I : \quad q_i(c) + \sum_{i'\in V(i)} h_{i',i}(c) - h_{i,i'}(c) - \frac{h^2_{i,i'}(c) + h^2_{i',i}(c)}{2} r_{i,i'} \geq d_i (SD)$$

$$\forall c \in \boldsymbol{C}^n, \forall (i,i') \in E : \quad h_{i,i'}(c) \geq 0$$

$$\forall i \in I, \forall j \in J, (c^{-j}, t_1, t_2), (c^1, \ldots, t_k, \ldots, c^N) \in \boldsymbol{C}_i, : V_i(c^1, .., c^{j-1}, t_1, c^{j+1} .., c^N)$$

$$- V_i(c^1, .., c^{j-1}, t_2, c^{j+1} .., c^N) = \int_{t_1}^{t_2} Q^j_i(c^1, .., c^{j-1}, s, c^{j+1} .., c^N) \mathrm{d}s \quad (H1)$$

$$\forall i \in I, \forall (c, c') \in c^2 : \quad (c - c').(Q_i(c) - Q_i(c')) \leq 0, \qquad (H2)$$

$$\forall i \in I, \forall c_i \in \boldsymbol{C}_i : \quad V_i(c_i) \geq 0 \quad (PC),$$

and

Problem 3.

$$\underset{(q,h)\in(\mathcal{Q},\mathbb{H})}{minimize}\mathbb{E}\sum_{i\in I}\sum_{j\in J} q_i^j(c)(c_i^j + K_i^j(c_i^j))$$

subject to

$$\forall(c,i)\in \boldsymbol{C}^n\times I : q_i(c) + \sum_{i'\in V(i)} h_{i',i}(c) - h_{i,i'}(c) - \frac{h^2_{i,i'}(c)+h^2_{i',i}(c)}{2} r_{i,i'} \geq d_i(SD)$$

$$\forall c\in \boldsymbol{C}^n, \forall(i,i')\in E : \quad h_{i,i'}(c)\geq 0.$$

$$\forall c\in \boldsymbol{C}_i, \forall i\in I : x_i(c) = \sum_{j\in J} q_i^j(c)c_i^j + \int_{c_i^j}^{c_i^{j+}} q_i^j(c_i^1\dots c_i^{j-1}, t, c_1^{(j+1)+}\dots c_i^{N+}; c_{-i})\mathrm{d}t.$$

The inequality on the scalar product in (H2) is the piecewise linear equivalent of a monotonicity condition already encountered in [38]. The first two problems are very similar, but (IC) has been replaced by (H1) and (H2) and (PC) is expressed in terms of V instead of U. This replacement is a trick introduced by Myerson in his 1981 paper. We will show later on how we can compare Problems 2 and 3, but note that Problem 3 is really simpler, as the optimization part can be solved pointwise (and x can be deduced from this pointwise optimization). The main result of this paper is that the three problems have the same solution.

7.3.1 Necessary conditions for Problem 1

We derive some necessary conditions for a solution of Problem 1. In fact, we only use constraint (IC) to deduce the two next results. The first lemma indicates that any solution of the first problem should be such that Q is monotonous. This is a classic result already introduced for instance in [70] and [38]. The novelty here is that in the context of piecewise linear production cost functions, this monotonicity result is expressed in a vectorial sense.

Lemma 7.2 (Q monotonicity)**.** *If (q,x,h) is admissible for Problem 1, then for all agent $i\in I$ and all $(c_i,c_i')\in \boldsymbol{C}_i^2$*

$$(c_i - c_i').(Q_i(c_i) - Q_i(c_i'))\leq 0 \tag{7.15}$$

where . is the scalar product in $\mathbb{R}^N$.

Proof. We omit the i in the proof, as it plays no role. First, let $(c,c')\in \boldsymbol{C}_i^2$ by the (IC) constraint,

$$U(c,c)\geq U(c,c') \quad\text{and}\quad U(c',c')\geq U(c',c) \tag{7.16}$$

i.e.

$$\begin{aligned} X(c) - \sum_{j\in J} c^j Q^j(c) &\geq X(c') - \sum_{j\in J} c^j Q^j(c') \\ X(c') - \sum_{j\in J} c^{j\prime} Q^j(c') &\geq X(c) - \sum_{j\in J} c^{j\prime} Q^j(c). \end{aligned} \tag{7.17}$$

We get the lemma after summation of the two inequalities and simplification. □

Lemma 7.2 indicates that an agent should be producing less on average in her ith working zone if she is biding a higher marginal cost for this working zone.

Lemma 7.3. *If (q, x, h) is admissible for Problem 1 then for any agent (omitting i) for any c, t_1 and t_2*

$$\begin{aligned} V(c^1, \ldots, c^{j-1}, t_1, c^{j+1}, \ldots, c^N) =& V(c^1, \ldots, c^{j-1}, t_2, c^{j+1}, \ldots, c^N) \\ &- \int_{t_2}^{t_1} Q^j(c^1, \ldots, c^{j-1}, s, c^{j+1}, \ldots, c^N)\mathrm{d}s \end{aligned} \tag{7.18}$$

Proof. The inequality $U(c, c) \leq U(c, c')$ implies that $c' \to U(c, c')$ is maximal at c for any $c \in \mathcal{C}_i$. Moreover,

$$t \to U((c^1, .., c^{j-1}, t, c^{j+1} .., c^N), c) = X(c) - \sum_{k \in J \setminus \{j\}} c^k Q^k(c) - tQ^j(c) \tag{7.19}$$

is absolutely continuous, differentiable with respect to t for all c, and its derivative is $-Q^j(c)$. By definition of q^j, $Q^j \leq \bar{q}$. So applying the envelope theorem we get the result. □

7.3.2 Necessary conditions for Problem 2

We derive some necessary conditions for a solution of Problem 2.

Lemma 7.4. *If (q, x, h) is an optimal solution of Problem 2 then (omitting i) for all $c \in \boldsymbol{C}_i$*

$$V(c) = \sum_{j \in J} \int_{c^j}^{c^{j+}} Q^j(c^1 \ldots c^{j-1}, t, c^{(j+1)+}, \ldots, c^{N+})\mathrm{d}t. \tag{7.20}$$

Proof. According to (H1)

$$\begin{aligned} \sum_{j \in J} \int_{c_j}^{c^{j+}} Q^j(c^1 \ldots c^{j-1}, t, c^{(j+1)+}, \ldots, c^{N+})\mathrm{d}t = \\ \sum_{j \in J} V(c^1, .., c^{j-1}, c^j, c^{(j+1)+}, \ldots, c^{N+}) - V(c^1, .., c^{j-1}, c^{(j)+}, \ldots, c^{N+}) \\ = V(c) - V(c^{1+}, \ldots, c^{N+}). \end{aligned}$$

This is an expression for $V(c)$ as a sum of a positive function of c and a constant $V(c^{1+}, \ldots, c^{N+})$. It is clear that to optimize the criteria, this constant should be as small as possible. The participation contraint (PC) imposes that $V(c^{1+}, \ldots, c^{N+}) \geq 0$, therefore $V(c^{1+}, \ldots, c^{N+}) = 0$. □

A consequence of this is:

Corollary 7.5. *If (q, x, h) is an optimal solution of Problem 2 then for all $i \in I$,*

$$V_i(c_i^{1+}, \ldots, c_i^{N+}) = 0. \tag{7.21}$$

Proof. See the proof of Lemma 7.4. □

Corollary 7.5 means that if an agent bids a production cost functions that is the maximum of what he could bid, he should not make any profit, and so he should be paid exactly his production cost. We see with this lemma that if the public information is inaccurate and the real cost of an agent is higher than what could be expected, then there is a risk that the participation constraint is not satisfied. On the other hand, it should not be surprising that an agent can have a zero profit: remember that in the extreme case in which the principal knows everything (discussed in §7.2), the agents do not make any profit.

Another consequence of lemma 7.4 is

Lemma 7.6. *If (q, x, h) is an optimal solution of Problem 2, the expected profit of agent i (over his type) is*

$$\mathbb{E}V_i(c) = \sum_{j \in J} \int_{(c_1..c_n) \in \boldsymbol{C}_i} Q_i^j(c^1, \ldots, c^j, c^{(j+1)+}, \ldots c^{N+}) K_i^j(c) f_i(c) \mathrm{d}c. \tag{7.22}$$

Proof. By Lemma 7.4 and Fubini's lemma, $\mathbb{E}V_i(c)$ is equal to

$$\mathbb{E} \sum_{j \in J} \int_{c^j}^{c^{j+}} Q_i^j(c^1, \ldots, c^{j-1}, t, c^{(j+1)+}, \ldots c^{N+}) \mathrm{d}t$$

$$= \sum_{j \in J} \int_{c^{-j} \in \boldsymbol{C}^{-j}} \int_{c^j = c^{j-}}^{c^{j+}} \int_{t = c^j}^{c^{j+}} Q_i^j(c^1, \ldots, c^{j-1}, t, c_i^{(j+1)+}, \ldots c_i^{N+}) f_i(c) \mathrm{d}t \mathrm{d}c^j \mathrm{d}c^{-j}.$$

Our task is now to compute the inner term. Applying again Fubini's lemma, this term is equal to

$$\begin{aligned}
&\int_{c^j = c^{j-}}^{c^{j+}} \int_{t = c^j}^{c^{j+}} Q_i^j(c^1, \ldots, c^{j-1}, t, c^{(j+1)+}, \ldots c^{N+}) f_i(c) \mathrm{d}t \mathrm{d}c^j = \\
&\int_{t = c^{j-}}^{c^{j+}} \int_{c^j = c^{j-}}^{t} Q_i^j(c^1, \ldots, c^{j-1}, t, c^{(j+1)+}, \ldots c^{N+}) f_i(c) \mathrm{d}c^j \mathrm{d}t = \\
&\int_{t = c^{j-}}^{c^{j+}} Q_i^j(c^1, \ldots, c^{j-1}, t, c^{(j+1)+}, \ldots c^{N+}) \left(\int_{c^j = c^{j-}}^{t} f_i(c) \mathrm{d}c^j\right) \mathrm{d}t = \\
&\int_{t = c^{j-}}^{c^{j+}} Q_i^j(c^1, \ldots, c^{j-1}, t, c^{(j+1)+}, \ldots c^{N+}) \left(\int_{c^j = c^{j-}}^{t} \frac{f_i(c)}{f_i(c^{-j}, t)} \mathrm{d}c^j\right) f_i(c^{-j}, t) \mathrm{d}t = \\
&\int_{t = c^{j-}}^{c^{j+}} Q_i^j(c^1, \ldots, c^{j-1}, t, c^{(j+1)+}, \ldots c^{N+}) K_i^j(t) f_i(c^{-j}, t) \mathrm{d}t =
\end{aligned}$$

$$\int_{c^j=c^{j-}}^{c^{j+}} Q_i^j(c^1,\ldots,c^{j-1},c^j,c^{(j+1)+},\ldots c^{N+})K_i^j(c^j)f_i(c_i)\mathrm{d}c^j$$

We get the lemma by summing all the inner terms. □

Lemma 7.7. *If (H1) is satisfied, then for any* $(a,b)\in \boldsymbol{C}_i^2$ *(omitting i)*

$$X(a)-X(b)=\sum_{j\in J}[a^jQ^j(a)-b^jQ^j(b)+\int_{a^j}^{b^j} Q^j(b^1\ldots b^{j-1},t,a^{j+1}\ldots a^N)\mathrm{d}t] \tag{7.23}$$

Proof. Because of its length the proof is detailled in Appendix 7.7 □

Lemma 7.8. *If* (q,x,h) *verifies (H1) and (H2) and* Q_i^j *is independent of* $c_i^{j'}$ *for* $j'>j$*, then for all* $(c,\tilde{c})\in\boldsymbol{C}^2$

$$U(c,c)\geq U(c,\tilde{c}). \tag{7.24}$$

Proof. Since (H1) is satisfied, equation (7.23) of Lemma 7.7 applies. We combine this relation with the definition of the expected profit U from (7.5). We obtain:

$$\begin{aligned}U(c,c)-U(c,\tilde{c}) &= \sum_{j\in J} c^jQ^j(c)-\tilde{c}^jQ^j(\tilde{c})+\\ &\int_{c^j}^{\tilde{c}^j} Q^j(\tilde{c}^1,...,\tilde{c}^{j-1},t,c^{j+1},...c^N)\mathrm{d}t+c^jQ^j(\tilde{c})-c^jQ^j(c)\\ &=\sum_{j\in J}(c^j-\tilde{c}^j)Q^j(\tilde{c}^1,...,\tilde{c}^{j-1},\tilde{c}^j))+\int_{c^j}^{\tilde{c}^j} Q^j(\tilde{c}^1,...,\tilde{c}^{j-1},t)\mathrm{d}t\\ &=\sum_{j\in J}\int_{c^j}^{\tilde{c}^j} Q^j(\tilde{c}^1,...,\tilde{c}^{j-1},t)-Q^j(\tilde{c}^1,...,\tilde{c}^{j-1},\tilde{c}^j)\mathrm{d}t,\end{aligned}$$

where we used the independence hypothesis for the second equality. By (H2), which implies the decreasingness of Q^j with respect to c_i^j when all other quantities are fixed, if $c^j<\tilde{c}^j$ then for any $t\in[c^j,\tilde{c}^j]$, $Q^j(t)-Q^j(\tilde{c}^j)\geq 0$. Otherwise, we use the formula $\int_a^b=-\int_b^a$ and the fact that any $t\in[\tilde{c}^j,c^j]$ verifies $Q^j(t)-Q^j(\tilde{c}^j)\leq 0$. So $U(c,c)-U(c,\tilde{c})$ is non negative. □

7.3.3 Necessary conditions for Problem 3

We derive some properties for Problem 3.

Lemma 7.9. *There is an optimal solution* (q,x,h) *for Problem 3 such that* q_i^j *(and* Q_i^j*) is independent of* c_i^k *for* $k\neq j$*.*

Proof. First note that x is not taking any role in the optimization problem: it is defined afterward. The only real optimization variables are then q and h. Remember that q_i^j is defined as a function of q by $q_i^j = \min((q_i - (j-1)\bar{q})^+, \bar{q})$. The constraints are defined for each $c \in \boldsymbol{C}^n$ and the integral criterion is in fact a sum of independent criteria depending on $q(c)$ for $c \in \boldsymbol{C}^n$. Therefore we can solve Problem 3 with a pointwise optimization. By the *discernability assumption*, for any $c \in \boldsymbol{C}^n$ and $i \in I$, $c_i^j + K_i^j(c_i^j)$ is increasing in j. So for all $c \in \boldsymbol{C}^n$, $i \in I$, $\sum_{j \in J} q_i^j(c)(c_i^j + K_i^j(c_i^j))$ is a convex criteria in q_i and so the pointwise problem corresponds to Problem 4 of §7.4. In particular, we can apply Lemma 7.15 from the next section. So q_i^j only depends on c_i^j and c_{-i}. This property is preserved by integration over the c_{-i}: Q_i^j only depends on c_i^j. □

We point out that, since the pointwise problem has a unique solution, the pointwise optimal solution introduced in the proof is uniquely defined.

Theorem 7.10. *If (q, x, h) is the pointwise optimal solution of Problem 3 and K_i^j is smooth in c_i^j for $(i, j) \in I \times J$ and $c \in \boldsymbol{C}_i$, then for all $i \in I$, Q_i is C^∞ over $\boldsymbol{C}_i$.*

Proof. We will use some results and notations from 7.4.2. Remember that $c_i^j \to c_i^j + K_i^j(c_i^j)$ is increasing, so by composition with smooth bijection, we can do the reasoning as if the costs involved were c_i^j instead of $c_i^j + K_i^j(c_i^j)$. First according to Lemma 7.16, q_i is continuous. Since q_i is bounded, we can apply the dominated convergence theorem to show that Q_i is continuous. Then we proceed by mathematical induction. Assume that Q_i is C^l, then take $c_i^0 \in \boldsymbol{C}_i$ and c_i^k a sequence in $\boldsymbol{C}_i$ that converges to c_i^0. Since $\hat{\mathcal{S}} = \cup_{k \in \mathbb{N}} \mathcal{S}(c_i^k)$ is a countable union of null measured set (by Lemma 7.34), its measure is zero. So without changing the results, we can compute the integrals on $\boldsymbol{C}^{-i} \backslash \hat{\mathcal{S}}$ instead of $\boldsymbol{C}^{-i}$. Since q_i and its derivatives are bounded, we can apply the dominated convergence theorem to compute the limit of $\frac{Q_i^{(l)}(c_i^0) - Q_i^{(l)}(c_i^k)}{c_i^0 - c_i^k}$ as k goes to $+\infty$ as the integral of a limit. Since we removed the point over which this limit was not defined, we get that $\frac{Q_i^{(l)}(c_i^0) - Q_i^{(l)}(c_i^k)}{c_i^0 - c_i^k}$ has a limit, and this limit does not depend on the sequence c_i^k. So Q_i is $l+1$ times derivable at c_i, for all c_i. We conclude by induction. □

7.3.4 Resolution of the mechanism design problem

Last but not least, we state the main result of the Section.

Theorem 7.11. *Let (q_i^j, h) be defined such that for any $c \in \boldsymbol{C}^n$, $(q_i^j(c_i^j, c_{-i}), h(c))$ solves*

$$\begin{aligned}
&\underset{q_i^j,x,h}{\textit{minimize}} \sum_{i\in I}\sum_{j\in J} q_i^j(c_i^j, c_{-i})(c_i^j + K_i^j(c_i^j))\\
&\textit{subject to}\\
&0 \le q_i^j \le \bar{q}\\
&\sum_{j\in J} q_i^i(c_i^j, c_{-i}) + \sum_{i'\in V(i)} h_{i',i}(c) - h_{i,i'}(c) - \frac{h_{i,i'}^2(c) + h_{i',i}^2(c)}{2} r_{i,i'} \ge d_i\\
&h_{i,i'}(c) \ge 0,
\end{aligned}$$

and set

$$q_i(c) = \sum_{j\in J} q_i^j(c_i^j, c_{-i}) \text{ and } x_i(c) = \sum_{j\in J} q_i^j(c_i^j, c_{-i})c_i^j + \int_{c_i^j}^{c_i^{j+}} q_i^j(t, c_{-i})\mathrm{d}t, \tag{7.25}$$

then (q, h, x) solves the optimal mechanism design problem (Problem 1).

Proof.

- First note that (q, h, x) is the pointwise solution of Problem 3 so it is optimal for Problem 3, moreover, by construction (q, h, x) satisfies (SD) and $h \ge 0$.

- Then note that by Lemma 7.6, (q, h, x) solves a relaxation of Problem 2, but is it admissible for Problem 2 ?

- By definition of V (omitting i),

$$\begin{aligned}
V(c_1 \dots a_j \dots c_N) - V(c_1 \dots b_j \dots c_N) =\\
\mathbb{E}x(c_1 \dots a_j \dots c_N) - x(c_1 \dots a_j \dots c_N) - [Q^j(a^j)a^j - Q^j(b^j)b^j] =\\
\mathbb{E}q_i^j(a^j, c_{-i})a^j + \int_{a^j}^{c_i^{j+}} q_i^j(t, c_{-i})\mathrm{d}t - \mathbb{E}q_i^j(b^j, c_{-i})b^j - \int_{b^j}^{c_i^{j+}} q_i^j(t, c_{-i})\mathrm{d}t\\
-[Q^j(a^j)a^j - Q^j(b^j)b^j] = \mathbb{E}\int_{a^j}^{b^j} q_i^j(t, c_{-i})\mathrm{d}t = \int_{a^j}^{b^j} Q_i^j(t)\mathrm{d}t
\end{aligned}$$

  where we used the definition of x, the definition of Q and Fubini lemma's for the second, third and fourth equalities. So (q, h, x) satisfies (H1).

- By construction, q_i^j is non-increasing in $c_i^j + K_i^j(c_i^j)$ then using the third assumption, q_i^j is non-increasing in c_i^j so for any $(a, b, c_{-i}) \in \boldsymbol{C}^2 \times \boldsymbol{C}^{-i}$, $(a_i^j - b_i^j)(q_i^j(a_i^j, c_{-i}) - q_i^j(b_i^j, c_{-i})) \le 0$, so by integration with respect to c_{-i}, $(a_i^j - b_i^j)(Q_i^j(a_i^j) - Q_i^j(b_i^j) \le 0$ and then by summation over j, $(c - c').(Q_i(c) - Q_i(c')) \le 0$, i.e. (H2) is satisfied.

- Since (H1) is satisfied, $V_i(c_i) \ge V_i(c_i^+)$. Moreover, $V_i(c_i^+) = 0$ by construction of x. So the participation constraint (PC) is satisfied.

- Therefore (q, h, x) is admissible for Problem 2. So it solves Problem 2.

- Since (q, h, x) solves Problem 2, by Lemma 7.8 the incentive compatibility constraint (IC) is satisfied. Moreover, by Lemma 7.4, (PC) is satisfied. So (q, h, x) is admissible for Problem 1, but is it optimal ?

- By Lemmas 7.2 and 7.3, any optimal solution of Problem 1 should be admissible for Problem 2. Since the criteria are the same, we conclude that (q, h, x) is an optimal solution of Problem 1.

$\square$

7.3.5 Comments

In the optimal mechanism, the agents are paid at a marginal price that is equal to their bid augmented by an information rent. This information rent depends on the problem structure by the fact that it is built from a collection of allocation problems, and it depends on the available information by the fact that, in these optimization problems, the marginal prices are replaced by the virtual marginal prices $c_i^j + K_i^j(c_i^j)$. We point out that, as already noted for instance in [29], the computation of such rent may pose a practical difficulty for large problems.

Notice that, by construction, the optimal mechanism is incentive compatible in dominant strategies no matter K as (H1) is verified anyway as long as the hypothesis are satisfied. If this market is repeated over time, the principal can dynamically enhance his probabilities.

The model extends to the more realistic case when some nodes do not have a producer and for some others, the demand is null. In particular, we can consider the buyer/suppliers setting where there is demand only at one node.

One may argue that one limit of the current result is that it does not take into account any network constraints. Nonetheless, the structure of the proof makes it clear that we exploited only some properties of the allocation problem. Therefore, the optimal mechanism construction is valid for any market for which the allocation problem satisfies these properties. We discuss more on this point in §7.3.6.

In addition, the optimal mechanism construction is valid for limiting case with $r = 0$ at some edges. In this case, one needs to specify the definition of q of as the solution of the allocation problem may not be a singleton. If all the agents are identical and $r = 0$ for all edges, this corresponds to a second best auction.

We have not tried any ironing techniques to get rid of the monotone likelihood ratio assumption ; this is probably something to look at.

7.3.6 Generalization

The study of this subsection could be postponed to a second reading. We extend Theorem 7.11 to a more general network market. In this subsection we use specific notations. The letter e is used generically to refer to a line. The network flow is now subject to a constraint of the form $N(h) \in \mathbb{R}^m_-$, where $N(h)$ is a convex and smooth function from $\mathbb{R}^E$ to $\mathbb{R}^m$, where $m \in \mathbb{N}$. We call this constraint the network constraint. To model the piecewise linear prices, we use positive variables $q_i^j \leq \bar{q}_i^j$. Thus, the working zones are not assumed to be of equal sizes anymore. The marginal rates c_i^j are assumed to be increasing in j. The criterion is still $J(q) = \sum_{i\in I}\sum_{j\in J} q_i^j c_i^j$. We write $\mathcal{K}_1$ the set of decisions (q_i^j, h_e) such that $N(h) \in \mathbb{R}^m_-$ and $0 \leq q_i^j \leq \bar{q}_i^j$. We assume that $\mathcal{K}_1$ is non-empty. The nodal constraints are replaced by constraints of the form (for all $i \in I$) $\sum_j q_i^j + g_i(h) \geq 0$ where the g_i are smooth strictly concave functions from $(\mathbb{R}^+)^E$ to $\mathbb{R}$. We introduce the set $\mathcal{K}_2 = \{(q_i^j, h_e) \in \mathcal{K}_1; \quad \forall i \in I \quad -\sum_j q_i^j - g_i(h) \leq 0\}$. Then the allocation problem corresponds to the following optimization program:

$$\begin{aligned} &\underset{(q_i^j,h)}{\text{minimize}}\ J(q) \\ &\text{subject to } (q_i^j, h_e) \in \mathcal{K}_2. \end{aligned} \tag{7.26}$$

It is clear that q_i^j is non-increasing in c_i^j. We point out that at optimality, the nodal constraint should be binding. Moreover, by the strict concavity of the g_i the solution of problem (7.26) is unique.[1] Note that J is smooth and its gradient at (c_i^j, h_e) is $(c_1^1, \dots c_1^N, \dots c_n^N, 0, \dots, 0)$, where the last $|E|$ null coordinates correspond to the variable h. We denote by $N_{\mathcal{K}_1}(q_i^j, h_e)$ and $N_{\mathcal{K}_2}(q_i^j, h_e)$ the normal cones to K_1 and K_2 at (q_i^j, h_e). Applying Theorem 10 from [82] (we can check that the constraint qualification is satisfied if q is not identically equal to zero), we can express $N_{\mathcal{K}_2}(q_i^j, h_e)$ as

$$\{\sum_{i\in I} \lambda_i \nabla f_i(q_i^j, h_e) + z; \quad (\lambda_1, \dots, \lambda_n) \in (\mathbb{R}_+)^n, z \in N_{\mathcal{K}_1}(q_i^j, h_e)\} \tag{7.27}$$

where $f_i(q,h) = -\sum_j q_i^j - g_i(h)$. Applying Theorem 9 from [82], the solution of (7.26) should satisfy

$$-\nabla J(q_i^j, h_e) \in N_{\mathcal{K}_2}(q_i^j, h_e). \tag{7.28}$$

Observe that since the problem is convex and the solution unique, this is in fact a necessary and sufficient condition for the unique solution of the problem. The N first rows of this relation gives:

$$(-c_1^1, \dots, -c_1^N) = \lambda_1(-1, \dots, -1) + (z_1, \dots, z_N). \tag{7.29}$$

[1] Take two optimal solutions, then check that the solution build with the average of the two flow vectors is admissible by convexity of the problem and strictly better by concavity of g.

where $\lambda_1 \geq 0$ and

$$z_j \begin{cases} \geq 0 & \text{if } q_i^j = \bar{q}_i^j \\ \leq 0 & \text{if } q_i^j = 0 \\ 0 & \text{else} \end{cases} \tag{7.30}$$

Note that if $z_j = 0$ then $c_i^j = \lambda_1$, and since the c_i^j are increasing in j, there is at most one j such that $z_j = 0$. Moreover, by 7.29 for all j we have $z_j = \lambda_1 - c_1^j$. So the z_j are strictly decreasing in j. From the product structure of $\mathcal{K}_1$ we deduce the product structure of its normal cone. We can then write with obvious notations: $N_{\mathcal{K}_1}(q,h) = N_{\mathcal{K}_1}(q) \times N_{\mathcal{K}_1}(h)$ From the rows corresponding to h in the first order condition we derive the relation:

$$\sum_{i \in I} \lambda_i \nabla g_i(h) \in N_{\mathcal{K}_1}(h). \tag{7.31}$$

Lemma 7.12 is a generalization of Lemma 7.15.

Lemma 7.12. *Let $(q(c), h(c))$ be a solution of Problem 7.26. Assume q_i continuous with respect to c_i, then for any $i \in I$, $j \in J$, $q_i^j(c)$ does not depend on c_i^l for $l \neq j$.*

Proof. Take $(c_i, c_{-i}) \in \boldsymbol{C}^n$. If $q_i^j(c_i, c_{-i}) \in]0, \bar{q}_i^j[$, then $\lambda_i = c_i^j$ and c_i^k $(k \neq j)$ does not intervene the first order conditions (7.30) and (7.31), so that the solution does not depends on it. So without loss of generality we assume $q_i^j(c_i, c_{-i}) = 0$ (the case $q_i^j(c_i, c_{-i}) = \bar{q}_i^j$ could be treated in the same manner). By the continuity assumption we can restrict even more to the case where $q_i^{j-1} = \bar{q}_i^{j-1}$ and $q_i^j(c_i, c_{-i}) = 0$. Then by the first order condition, $\lambda_i \in [c_i^{j-1}, c_i^j]$. Using the same the first order condition argument we used at the beginning of this proof, we see that the solution only depends on c_i^{j-1} and c_i^j. If c_i^{j-1} increases, then q_i decrease so that q_i^j stays equal to zero. If c_i^{j-1} decreases, then the first order condition $\lambda_i \in [c_i^{j-1}, c_i^j]$ stays true for the current λ_i, the whole first order condition is still satisfied. Therefore the solution does not change. The lemma follows. □

Notice that we can write q_i as a strictly convex function of h $q_i = -g_i(h)$, and then the cost associated with q_i is the composition of an increasing convex function of $\mathbb{R}$ and a convex function from $\mathbb{R}^{|E|}$ to $\mathbb{R}$, therefore it is convex with respect to h, then we can rewrite the problem with only h as a decision variable, the problem would be defined on a convex set and with a strictly convex cost, and parametrized by $c \in \boldsymbol{C}$. Then we can apply Berge maximum principle (see Theorem 9.17 in [85]) in a convex setting to get the continuity of q. From Lemma 7.12 and the monotony of q, we conclude that we can extend Theorem 7.11 to a more general setting.

7.3.7 Examples with log-concave functions

We point out that a sufficient condition to check the monotone likelihood ratio property is that F/f is increasing. If F is a smooth cumulative distribution function with f the corresponding smooth and positive density, then F/f is increasing iff f/F is decreasing iff $\ln F'$ is decreasing iff $\ln F$ is concave. A function f is said to be *log-concave* if $\ln f$ is concave. Many density functions encountered in the economic and engineering literature are *log-concave*: the uniform, the normal, the exponential, the power function and the Laplace distribution have log-concave density function. We refer to [10] for the results we use on this class of functions. The class of *log-concave* is stable by monotonic transformation and truncation. Moreover, it happens that if a probability density distribution is log-concave, then the corresponding cumulative distribution is log-concave. In mechanism design theory, it is standard to assume F to be log-concave [62].

We want to see the implication of the *discernability assumption*. This assumption imposes a gap Δ equals to $K_i^j(c_i^{j+})$ between c_i^{j+} and $c_i^{(j+1)-}$. We compute this gap for some standard cases. To simplify the notations and the computation, we assume without loss of generality that $c^{j-} = 0$ and write $c^{j+} = c^+$. We get the following table:

Table 7.1 – The gap Δ for some standard probabilities

Name	$\propto f(x)$	$\propto F(x)$	$K(x)$	Δ
Uniform	1	x	x	c^+
Power Function	$\lambda(\frac{x}{c^+})^{\lambda-1}$	$c^+(\frac{x}{c^+})^\lambda$	$\frac{x}{\lambda}$	$\frac{c^+}{\lambda}$
Weibull	$\lambda(\frac{x}{c^+})^{\lambda-1}e^{(-\frac{x}{c^+})^\lambda}$	$c^+(1-e^{-(\frac{x}{c^+})^\lambda})$	$\frac{c^+}{\lambda}(\frac{c^+}{x})^{\lambda-1}(e^{(\frac{x}{c^+})^\lambda}-1)$	$c^+\frac{e-1}{\lambda}$
Laplace	$\frac{1}{2}e^{-\lambda\lvert x-\frac{c^+}{2}\rvert}$	$x>\frac{c^+}{2}$, $\frac{2-e^{-\lambda\frac{c^+}{2}}e^{-\lambda(x-\frac{c^+}{2})}}{2\lambda}$		$\frac{2}{\lambda}(e^{\frac{c^+}{2}\lambda}-1)$
Exponential (reversed)	$\lambda e^{-(c^+-x)\lambda}$	$e^{-c^+\lambda}(e^{x\lambda}-1)$	$\frac{1-e^{-x\lambda}}{\lambda}$	$\frac{1-e^{-c^+\lambda}}{\lambda}$

We truncate the probabilities so that they have support in $[0, c^+]$. The symbole $\propto$ means that we express f and F modulo the multiplication by a common constant (due to the truncation) and λ is a positive parameter that should be greater than 1 for the Power function and the Weibull probability. For the uniform distribution, we see that the interval should be of non-decreasing sizes. For instance, one could take $c^1 \in [\bar{c}, 2\bar{c}]$, $c^2 \in [3\bar{c}, 4\bar{c}]$, $c^3 \in [5\bar{c}, 6\bar{c}]$, etc. For the Power function, the Weibull function and the exponential, we see that the gap could be made smaller. We do not address in this work the question of the practical implementation of an optimal mechanism. The *discernability assumption* raises an additional practical issue.

7.4 Study of the allocation problem

7.4.1 The standard auction problem

The previous section motivates the study of the allocation problem for different reasons. First, as we have seen in the proofs, the results of §7.3 rely on some properties of the solution of the standard allocation problem. In addition to those properties, we derive in this section two algorithms to compute the solution of the standard allocation problem. According to 7.11, those algorithms can be used for both the original auction problem and the optimal mechanism design. To benchmark the mechanism design equilibrium against an equilibrium of the Bayesian game related to the standard auction, numerical efficiency is pivotal: indeed the Bayesian equilibrium requires a lot of allocations computations.

Let us first introduce the standard allocation problem. In a standard mechanism, the principal solves an allocation problem based on the bids he receives. Those bids will be denoted by c_i^j, where as before $i \in I$ corresponds to the ith agent and $j \in J$ corresponds to the jth working zone with constant marginal price. To model the fact that the production costs are piecewise linear, we use some positive variables q_i^j so that $q_i^j \leq \bar{q}$, for any $i \in I$, the quantity produced by agent i is $q_i = \sum_{j\in J} q_i^j$ and the related production cost is $\sum_{j\in J} c_i^j q_i^j$. As before, an allocation should satisfy the constraint that production exceeds demand. We end up with Problem 4:

Problem 4.

$$
\begin{aligned}
\underset{(q,h)}{\textit{minimize}} \quad & \sum_{i\in I}\sum_{j\in J} q_i^j c_i^j \\
\textit{subject to} \quad & \forall i \in I : \sum_{j\in J} q_i^j + \sum_{i'\in V(i)} h_{i',i} - h_{i,i'} - \frac{h_{i,i'}^2 + h_{i',i}^2}{2} r_{i,i'} \geq d_i \quad (\lambda_i) \\
& \forall (i,i') \in E : h_{i,i'} \geq 0 \quad (\gamma_{i,i'}) \\
& \forall (i,j) \in I \times J : q_i^j \geq 0 \quad (\mu_{i,j}) \\
& \forall (i,j) \in I \times J : q_i^j \leq \bar{q} \quad (\nu_{i,j}).
\end{aligned}
\tag{7.32}
$$

The notations for the dual the variables associated with each constraint are indicated in parentheses. Those variables are in $\mathbb{R}_+$.

For any node $i \in I$, we define the function F_i for $\lambda \in [\min_i c_i^1, \max_i c_i^N]^n$

$$
F_i(\lambda_i, \lambda_{-i}) = d_i + \sum_{i'\in V(i)} \frac{\lambda_{i'} - \lambda_i}{r_{i,i'}(\lambda_i + \lambda_{i'})} + \frac{(\lambda_{i'} - \lambda_i)^2}{2r_{i,i'}(\lambda_i + \lambda_{i'})^2}. \tag{7.33}
$$

We will justify later that this function could be interpreted as the production of agent i when the multipliers are λ_i and λ_{-i}. Its partial derivative with respect to λ_i is

$$
\partial_{\lambda_i} F_i(\lambda_i, \lambda_{-i}) = -\sum_{i'\in V(i)} \frac{4}{r_{i,i'}} \frac{\lambda_{i'}^2}{(\lambda_i + \lambda_{i'})^3} < 0. \tag{7.34}
$$

The derivative is negative: when i increases its price it is assigned smaller production quantities. The partial derivative of F_i for $i' \in I\backslash\{i\}$ is

$$\partial_{\lambda_{i'}} F_i(\lambda_i, \lambda_{-i}) = \begin{cases} \frac{4}{r_{i,i'}} \frac{\lambda_{i'}\lambda_i}{(\lambda_i+\lambda_{i'})^3} > 0 & \text{if } i' \in V(i) \\ 0 & \text{else.} \end{cases} \tag{7.35}$$

When another agent becomes less competitive, i is assigned more production. Let $k \in J \cup \{0\}$. The limit at $+\infty$ and 0 of $F_i(x, \lambda_{-i}) - k\bar{q}$ are

$$\lim_{x \to +\infty} F_i(x, \lambda_{-i}) - k\bar{q} = d_i - k\bar{q} - \sum_{j \in V(i)} \frac{1}{2r_{i,j}} \tag{7.36}$$

and

$$\lim_{x \to +\infty} F_i(x, \lambda_{-i}) - k\bar{q} = d_i - k\bar{q} + \sum_{j \in V(i)} \frac{3}{2r_{i,j}}. \tag{7.37}$$

Using the hypotheses (7.11), the first term is strictly negative and the second strictly positive, so by the intermediate value theorem, $F_i - k\bar{q}$ has a zero. Since $F_i - k\bar{q}$ is decreasing in λ_i, this solution is unique. Now we define for $i \in I$ and $k \in J \cup \{0\}$, g_i^k as the function that associates any $\lambda_{-i} \in [\min_i c_i^1, \max_i c_i^N]^{n-1}$ with the unique x such that and $F_i(x, \lambda_{-i}) = k\bar{q}$ and $x > 0$:

$$\begin{aligned} &F_i(g_i^k(\lambda_{-i}), \lambda_{-i}) = k\bar{q} \\ &g_i^k(\lambda_{-i}) > 0. \end{aligned} \tag{7.38}$$

Lemma 7.13. *For any $i \in I$, $k \in J \cup \{0\}$, $\lambda^{-i} \in [\min_i c_i^1, \max_i c_i^N]^{n-1}$ and $i' \in V(i)$*

$$\partial_{\lambda_{i'}} g_i^k(\lambda_{-i}) > 0. \tag{7.39}$$

In particular, g_i^k is increasing in $\lambda_{i'}$ for $i' \in V(i)$.

Proof. According to the implicit function theorem

$$\frac{\partial g_i^k(\lambda_{-i})}{\partial \lambda_{i'}} = -\frac{\partial F_i}{\partial \lambda_{i'}} / \frac{\partial F_i}{\partial \lambda_i}, \tag{7.40}$$

□

It is clear that $g_i^k(\lambda_{-i})$ is decreasing in k. We proceed with the computation of the dual of Problem 4. If a strong duality theorem applies, then we should have

$$\min_{q,h} \max_{\lambda,\gamma,\nu,\mu} \sum_{i \in I, j \in J} q_i^j c_i^j +$$

$$\sum_{i \in I} \lambda_i \{d_i - (\sum_{j \in J} q_i^j + \sum_{i' \in V(i)} h_{i',i} - h_{i,i'} - \frac{h_{i,i'}^2 + h_{i',i}^2}{2} r_{i,i'})\}$$

$$
\begin{aligned}
&- \sum_{i\in I, j\in J} \gamma_{i,j} h_{i,j} + \sum_{i\in I, j\in J} \nu_{i,j}(q_i^j - \bar{q}) - \mu_{i,j} q_i^j \\
= \max_{\lambda,\gamma,\nu\mu} \min_{q,h} &\sum_{i\in I} \lambda_i d_i - \sum_{i\in I, j\in J} \nu_{i,j}\bar{q} + q_i^j(c_i^j + \nu_{i,j} - \lambda_i - \mu_{i,j}) \\
&+ \sum_{(i,i')\in E} h_{i,i'}\{\lambda_i - \lambda_{i'} - \gamma_{i,j}\} + h_{i,i'}^2 r_{i,i'} \frac{\lambda_i + \lambda_{i'}}{2},
\end{aligned}
$$

so that for any $(i,i') \in E$, by necessary and sufficient first order condition

$$
h_{i,i'} = \frac{\gamma_{i,i'} + \lambda_{i'} - \lambda_i}{r_{i,i'}(\lambda_{i'} + \lambda_i)}. \tag{7.41}
$$

By replacing h by its expression in the dual variables we get something equivalent to

$$
\begin{aligned}
\underset{(\lambda,\gamma,\mu,\nu)}{\text{maximize}} \quad & \sum_{i\in I}\{\lambda_i d_i - \sum_{j\in J} \nu_{i,k}\bar{q} - \sum_{i'\in V(i)} \frac{(\lambda_i - \lambda_{i'} - \gamma_{i,j})^2}{2r_{i,i'}(\lambda_i + \lambda_{i'})}\} \\
\text{subject to} \quad & \forall (i,j) \in I \times J \quad c_i^j + \nu_{i,j} \geq \lambda_i + \mu_{i,j}.
\end{aligned} \tag{7.42}
$$

The expression of γ with respect to λ follows. For any $(i,i') \in E$

$$
\gamma_{i,i'} = \begin{cases} 0 & \text{if } \lambda_i \leq \lambda_{i'} \\ \lambda_i - \lambda_{i'} & \text{else} \end{cases} \tag{7.43}
$$

so the dual problem is equivalent to

$$
\begin{aligned}
\underset{(\lambda,\mu,\nu)}{\text{maximize}} \quad & \sum_{i\in I}\{\lambda_i d_i - \sum_{j\in J} \nu_{i,j}\bar{q} - \sum_{i'\in V(i)} \frac{(\lambda_i - \lambda_{i'})^2}{4r_{i,i'}(\lambda_i + \lambda_{i'})}\} \\
\text{subject to} \quad & \forall (i,j) \in I \times J \quad c_i^j + \nu_{i,j} \geq \lambda_i + \mu_{i,j},
\end{aligned} \tag{7.44}
$$

because μ does not play any role in the admissibility of the other variables nor in the objective, this is equivalent to

$$
\begin{aligned}
\underset{(\lambda,\nu)}{\text{maximize}} \quad & \sum_{i\in I}\{\lambda_i d_i - \sum_{j\in J} \nu_{i,j}\bar{q} - \sum_{i'\in V(i)} \frac{(\lambda_i - \lambda_{i'})^2}{4r_{i,i'}(\lambda_i + \lambda_{i'})}\} \\
\text{subject to} \quad & \forall (i,j) \in I \times J \quad c_i^j + \nu_{i,j} \geq \lambda_i,
\end{aligned} \tag{7.45}
$$

The expression of ν follows. For any $(i,j) \in I \times J$

$$
\nu_{i,j} = \begin{cases} 0 & \text{if } \lambda_i \leq c_i^j \\ \lambda_i - c_i^j & \text{else.} \end{cases} \tag{7.46}
$$

So we can a posteriori justify that we have strong duality: the operator is continuous, convex-concave and the dual variables are restricted to be in a bounded set.

So the dual of the allocation problem writes:

$$\underset{\lambda\geq 0}{\text{maximize}} \quad \sum_{i\in I}\{\lambda_i d_i - \bar{q}\sum_{j\in J}(\lambda_i - c_i^j)\delta_{\lambda_i\geq c_i^j} - \sum_{i'\in V(i)} \frac{(\lambda_i-\lambda_{i'})^2}{4r_{i,i'}(\lambda_i+\lambda_{i'})}\}, \tag{7.47}$$

where

$$\delta_{x\geq y} = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{else.} \end{cases} \tag{7.48}$$

For $i \in I$ we maximize the criteria

$$\lambda_i d_i - \bar{q}\sum_{j\in J}(\lambda_i - c_i^j)\delta_{\lambda_i\geq c_i^j} - \sum_{i'\in V(i)} \frac{(\lambda_i-\lambda_{i'})^2}{4r_{i,i'}(\lambda_i+\lambda_{i'})}, \tag{7.49}$$

which is strictly concave for any λ_{-i} (sum of concave and strictly concave functions). We denote by $\Lambda_i(\lambda_{-i})$ its maximizer. The first order necessary and sufficient condition on Λ_i is:

$$0 \in F_i(\Lambda_i, \lambda_{-i}) - K_i(\Lambda_i), \tag{7.50}$$

where

$$K_i(\lambda_i) = \begin{cases} 0 & \text{if } \lambda_i < c_i^1 \\ [j-1, j]\bar{q} & \text{if } \lambda_i = c_i^j \\ j\bar{q} & \text{if } \lambda_i \in]c_i^j, c_i^{j+1}[, j \neq N \\ N\bar{q} & \text{if } \lambda_i \in \lambda_i \in]c_i^N, \bar{c}[, \end{cases} \tag{7.51}$$

We conclude

Lemma 7.14. *For any $i \in I$ and any $\lambda^{-i} \in [\min_i c_i^1, \max_i c_i^N]^{n-1}$, $\Lambda_i(\lambda_{-i})$ is the unique solution of*

$$F_i(\Lambda_i, \lambda_{-i}) \in K_i(\Lambda_i). \tag{7.52}$$

We point out that the primal (and dual) solution unicity is a desirable property that is not systematic for the allocation problems of centralized market models. The expression of h with respect to λ (7.41) and the fact the supply constraint should be binding at optimality justify the interpretation of F_i proposed at the beginning of this subsection. In the following we use this property many times.

7.4.2 Some properties of the solution

If r and d are set, we can see the solution of Problem 4 as a function of the vector $c \in \boldsymbol{C}^n$. We denote by $q(c)$ the solution of Problem 4 with the cost vector c. Similarly, we define $q_i(c)$, $q_i^j(c)$, $\lambda(c)$ and $\lambda_i(c)$. We give here two properties of the allocation problem solution. By integration, we showed in the previous section that the solution of the mechanism design inherits those properties.

Lemma 7.15. *Let $(q(c), h(c))$ be a solution of Problem 4, then $q_i^j(c)$ does not depend on c_i^l for $l \neq j$:*

$$q_i^j(c^1, \dots c^{j-1}, c^j, c^{j+1} \dots, c^N; c^{-i}) = q_i^j(s^1, \dots s^{j-1}, c^j, s^{j+1} \dots, s^N; c^{-i}) \tag{7.53}$$

Proof. Let $i \in I$, $j \in J$, $c_{-i} \in \boldsymbol{C}^{n-1}$, $c = (c^1, \dots, c^N) \in \boldsymbol{C}$ and $s = (s^1, \dots, s^N) \in \boldsymbol{C}$ such that $s^j = c^j$. We shall prove that $q_i^j(s, c^{-i}) = q_i^j(c, c^{-i})$. We denote by λ^c (resp. λ^s) the dual variables associated with the nodal contraints for the allocation problem parametrized with c (resp. s). First if

$$q_i^j(c, c^{-i}) \in]0, \bar{q}[, \tag{7.54}$$

then by lemma 7.14 $\lambda_i^c = c_i^j$ and so using Lemma 7.14 again, $\lambda_i^s = c_i^j$. Therefore $\lambda^s = \lambda^c$, from which we deduce that $q_i^j(c, c^{-i}) = q_i^j(s, c^{-i})$.

So without loss of generality, we can assume that

$$q_i^j(c, c^{-i}) = \bar{q} \quad \text{and} \quad q_i^j(s, c^{-i}) = 0. \tag{7.55}$$

Then using Lemma 7.14 we get

$$\lambda_i^c \geq c^k \quad \text{and} \quad \lambda_i^s \leq c^k, \tag{7.56}$$

so that $\lambda_i^c \geq \lambda_i^s$. If $\lambda_i^c > \lambda_i^s$, then $\lambda_{-i}^c \geq \lambda_{-i}^s$ by non-decreasingness of $\Lambda_{i'}$, $i' \in I \backslash \{i\}$ (explained in §7.4.3) Therefore all the other agents are producing less, which is absurd since i is already producing less.

□

We extend the notations by setting for all $i \in I$, $c_i^0 = c_*$. We consider the subset $\mathcal{S}$ of $\boldsymbol{C}$ for which at some nodes i, the multiplicator λ_i is equal to the marginal cost *and* the production is a multiple of $\bar{q}$ (i.e. stuck in an angle):

$$\mathcal{S} = \{c \in \boldsymbol{C}^n, q_i(c) = j\bar{q} \text{ and } \lambda_i(c) = c_i^{j'} \text{ for some } i \in I, j \in J \cup \{0\}, j' \in \{j, j+1\}\}. \tag{7.57}$$

The set $\mathcal{S}$ corresponds to the points of transition between the two possibilities defined by the first order condition (7.50). Because of the angle, it is natural to think that this is where irregularities may happen (see the proof of the next lemma). We introduce this set to show some regularity properties of q and Q. We detail the proof in the Appendix. The approach consists in showing that $\mathcal{S}$ is a finit union of sets of zero measure. This is also true for the projection of $\mathcal{S}$ on the $\{c_i\} \times \mathcal{C}^{-i}$. Then we observe that on $\boldsymbol{C}\backslash\mathcal{S}$, the relations between the primal and dual variables are smooth.

Lemma 7.16. *The function q is C^∞ on $\boldsymbol{C}^n\backslash\mathcal{S}$ and C^0 on $\boldsymbol{C}^n$.*

Proof. We postpone the proof to Appendix 7.8 □

7.4.3 Fixed point

In this subsection we show that the solution of the dual problem is the unique fixed point of a monotone operator. We define

$$\Lambda(\lambda_1, ..., \lambda_n) = (\Lambda_1(\lambda_{-1}), ..., \Lambda_n(\lambda_{-n})). \tag{7.58}$$

Lemma 7.17. *For any $i \in I$, Λ_i is non-decreasing.*

Proof. Let $\lambda_{-i} < \lambda'_{-i}$ and the corresponding Λ_i and Λ'_i. Assume $\Lambda_i > \Lambda'_i$. Since F_i is decreasing in the first variable and increasing in the second

$$F_i(\Lambda_i, \lambda_{-i}) < F_i(\Lambda'_i, \lambda'_{-i}) \tag{7.59}$$

Moreover for any $x \in K(\Lambda'_i)$ and $y \in K(\Lambda_i)$, $x \leq y$ and $F_i(\Lambda_i, \lambda_{-i}) \in K(\Lambda_i)$, $F_i(\Lambda'_i, \lambda'_{-i}) \in K(\Lambda'_i)$. Therefore $F_i(\Lambda'_i, \lambda'_{-i}) \leq F_i(\Lambda_i, \lambda_{-i})$ which is absurd. □

We will use the following classical result (see [87] for a proof and definition of complete lattice).

Theorem 7.18 (Knaster-Tarski fixed point)**.** *Let L be a complete lattice and let f an application from L to L and order preserving. Then the set of fixed points of f in L is a complete lattice.*

In particular, the set of fixed points is non empty. Since Λ is order preserving and $[c_*, c^*]^n$ is a lattice when we consider the natural order, there is a fixed point, and the set of fixed points is a lattice.

Lemma 7.19. *λ is optimal for the dual $\Leftrightarrow$ λ is a fixed point of Λ.*

Proof.

- If λ is optimal for the dual, then each component i maximizes the criteria (7.49), so λ is a fixed point of Λ.
- If λ is a fixed point of Λ, then by definition, each component i maximizes the criteria (7.49). So since the problem is (strictly) concave, λ is optimal.

□

A consequence of the previous lemma is that

Lemma 7.20. *The set of fixed points of Λ is a singleton.*

Definition 7.21 (Continuous for monotone sequence)**.** *We consider the natural partial order on $\mathbb{R}^n$. We say that a function G is continuous for monotone (resp. increasing, decreasing) sequences if for any monotone (resp. increasing, decreasing) sequence x_n converging to a point x in the domain of G, $G(x_n)$ goes to $G(x)$ as n goes to infinity.*

Obviously, a function is continuous for monotone sequences if and only if it is continuous for increasing and decreasing sequences.

Lemma 7.22. *The operator Λ is continuous for monotone sequences.*

The intuition of the proof is that we can use the monotony of the sequence and Lemma 7.14 to characterize the behaviour of Λ on the neighborhood. We find that Λ is either constant or characterized by the implicit function theorem.

Proof. Let $\bar{\lambda}_{-i}$, $j \in [1 \dots N]$, we first deal with the 'nice' case, that corresponds to $F_i(\Lambda(\bar{\lambda}_{-i}), \bar{\lambda}_{-i}) \in]j-1, j[\bar{q}$

- If $\Lambda_i(\bar{\lambda}_{-i}) \in]c_i^j, c_i^{j+1}[$ (we do not treat the case $j = N$, which is very similar to what follows) then since F_i is C^∞ and of invertible derivative (non zero) in λ_i, the implicit function theorem tells us that the solution ψ of $F_i(\psi(\bar{\lambda}_{-i}), \bar{\lambda}_{-i}) = j\bar{q}$ is continous in a neighborhood B of $\bar{\lambda}_{-i}$. So we can take B small enough so that for $\lambda_{-i} \in B$, $\psi(\lambda_{-i}) \in]c_i^j, c_i^{j+1}[$. On this neighborhood, ψ satisfies the first order conditions and so by unicity of the solution of the optimization problem, since those conditions are sufficient, $\psi = \Lambda_i$ on B. So Λ_i is continous at $\bar{\lambda}_{-i}$.

- If $\Lambda_i(\bar{\lambda}_{-i}) = c_i^j$ (as before, we do not treat the case $j = N$), then by Lemma 7.14 $F_i(\Lambda_i(\bar{\lambda}_{-i}), \bar{\lambda}_{-i}) = [j-1, j]\bar{q}$, if $F_i \in]j-1, j[\bar{q}$ (we deal with the border case in the next point) then since F_i is continuous, there is a neighborhood B of $\bar{\lambda}_{-i}$ such that $F_i(\Lambda_i(\bar{\lambda}_{-i}), \lambda_{-i}) \in]j-1, j[\bar{q}$, so on B Λ_i is constant so continuous.

- We proceed with the borders. If $F_i(\Lambda_i(\bar{\lambda}_{-i}), \bar{\lambda}_{-i}) = (j-1)\bar{q}$ and $\Lambda_i(\bar{\lambda}_{-i}) = c_i^j$.

  - Decreasing case: Let us take $\epsilon \in \mathbb{R}_+^{n-1}$ such that $F_i(\Lambda_i(\bar{\lambda}_{-i}), \bar{\lambda}_{-i} + \epsilon) \in [j-1, j]\bar{q}$ (F_i is continuous and increasing in λ_{-i}). Then $\Lambda_i(\bar{\lambda}_{-i} + \epsilon) = \Lambda_i(\bar{\lambda}_{-i})$ checks the first order condition so Λ is constant, so we get the continuity for decreasing sequences.

  - Increasing case: $F_i(\Lambda_i(\bar{\lambda}_{-i}), \bar{\lambda}_{-i}) = (j-1)\bar{q}$ and so there exists a ball B such that the implicit function theorem applies and there exists ψ such that $F_i(\psi(\bar{\lambda}_{-i} - \epsilon), \bar{\lambda}_{-i} - \epsilon) = (j-1)\bar{q}$ and $\psi(\bar{\lambda}_{-i}) = \Lambda_i(\bar{\lambda}_{-i}) = c_i^j$ (remember that $\Lambda_i(\bar{\lambda}_{-i}) = c_i^j$

by hypothesis) . Since F_i is increasing in the second variable and decreasing in the first, ψ is increasing. For ϵ of positive components and sufficiently small, $\psi(\bar{\lambda}_{-i}-\epsilon) \in]c_i^{j-1}, c_i^j[$ (since $\psi(\bar{\lambda}_{-i}) = \Lambda_i(\bar{\lambda}_{-i}) = c_i^j$) and so check the first order condition. So for ϵ of positive components and sufficiently small, $\psi = \Lambda_i$ by uniqueness of the solution. So Λ_i is continuous for increasing sequence.

- We do the same analysis if $F_i(\Lambda_i(\bar{\lambda}_{-i}), \bar{\lambda}_{-i}) = j\bar{q}$ and $\Lambda_i(\bar{\lambda}_{-i}) = c_i^j$.

The conclusion follows. □

We could have alternatively used Berge Maximum theorem for strictly concave criterion to get the continuity of Λ. Yet, we choose to present this proof for pedagogical reasons since it contains some key ideas we will use later (see appendix).

Theorem 7.23. *The sequence $(\Lambda^k(c_1^N...c_n^N))_{k\in\mathbb{N}}$ converges to the solution of the dual.*

Proof. Since $\Lambda(c_1^N...c_n^N) \leq (c_1^N...c_n^N)$, and since Λ is order preserving, the sequence $\Lambda^k(c_1^N...c_n^N) = \lambda^k$ is non increasing and bounded, so converge to a point x. Since Λ is continuous for monotone sequence, x is a fixed point. □

Theorem 7.24. *For any $i \in I$, $\lambda_{-i} \in [c_*, c^*]^{n-1}$, $\Lambda_i(\lambda_{-i})$ has the following explicite expression:*

$$\Lambda_i(\lambda_{-i}) = \min\{c_i^N, \min_{j\in J}\{c_i^j 1_{F_i(c_i^j,\lambda_{-i})<j\bar{q}}\}, \min_{k\in[0..N-1]}\{g_i^k(\lambda_{-i})1_{g_i^k(\lambda_{-i})\in[c_i^k,c_i^{k+1}]}\}\} \tag{7.60}$$

Proof. We denote by G_i the RHS of (7.60) and show that for any i

$$F_i(G_i(\lambda_{-i}), \lambda_{-i}) \in K(G(\lambda_{-i})), \tag{7.61}$$

and then we conclude with a uniqueness argument.

If there is $j \in J$ such that $G_i(\lambda_{-i}) = c_i^j$, then either $F_i(c_i^j, \lambda_{-i}) < j\bar{q}$ or $g_i^j(\lambda_{-i}) = c_i^j$. This last possibility implies by definition of g_i^j that $F_i(c_i^j, \lambda_{-i}) = j\bar{q}$. So anyway $F_i(c_i^j, \lambda_{-i}) \leq j\bar{q}$. Remember that $K(G(\lambda_{-i})) = [j-1, j]\bar{q}$. So we need to prove that $F_i(c_i^j, \lambda_{-i}) \geq (j-1)\bar{q}$. Suppose the contrary, i.e. $F_i(c_i^j, \lambda_{-i}) < (j-1)\bar{q}$. Then since $G_i(\lambda_{-i}) = c_i^j$, $F(c_i^j, \lambda_{-i}) < (j-1)\bar{q}$, which in turn implies that

$$g_i^j(\lambda_{-i}) < c_i^j. \tag{7.62}$$

Now observe that since $G_i(\lambda_{-i}) = c_i^j$, $F(c_i^{j-1}, \lambda_{-i}) > (j-1)\bar{q}$, which implies that

$$g_i^j(\lambda_{-i}) > c_i^{j-1}. \tag{7.63}$$

Combining (7.62) and (7.63) with the definition of G, we see that $G(\lambda_{-i}) \leq g_i^j(\lambda_{-i})$. But $G(\lambda_{-i}) = c_i^j$ and $g_i^j(\lambda_{-i}) < c_i^j$, so this is absurd. Therefore $F_i(c_i^j, \lambda_{-i}) \geq (j-1)\bar{q}$.

Else let us assume that there is not such j. Then there is $k \in [0 \dots N-1]$ such that $G_i(\lambda_{-i}) = g_i^k(\lambda_{-i})$. By definition of g_i^k, $F_i(G_i(\lambda_{-i}), \lambda_{-i})) = k\bar{q}$ and by definition of G, $G_i(\lambda_{-i}) \in [c_i^k, c_i^{k+1}]$. So again $F_i(G_i(\lambda_{-i}), \lambda_{-i})) \in K(G_i(\lambda_{-i}))$. We can now conclude that $\Lambda = G$. □

We can interpret the fixed point algorithm as if some benevolent agents situated at each node of the network were exchanging information. They collectively try to minimize the total cost and, to do so, they communicate their current marginal costs. This marginal cost is the minimum of their local marginal cost and the marginal cost of importation from the adjacent nodes. At each iteration, the agents compute how much they are going to produce based on their current marginal cost. Then they update their marginal cost based on the information they just received and transmit this marginal cost to the adjacent nodes. We point out that the information used by each agent is local.

7.4.4 Decreasing Rate

We derive in this section an estimate for the decreasing rate. We denote $\alpha = \max_{(e,e') \in E^2} r_e / r_{e'}$. We have the following bound:

Lemma 7.25. *For any $(i, i', k, \lambda_{-i}) \in E \times [0, N] \times [c_*, c^*]^{n-1}$,*

$$\partial_{\lambda_i} g_{i'}^k(\lambda_{-i}) \geq \frac{1}{N\alpha}(\frac{c_*}{c^*})^5. \tag{7.64}$$

Proof. We combine (7.40) with (7.34) and (7.35). □

Lemma 7.26. *Since $(\lambda_i^k)_{k \in \mathbb{N}}$ is non-increasing for all $i \in I$, there is a finite number of k for which at least one coordinate λ_i^k satisfies*

$$\lambda_i^k > c_i^q \quad \textit{and} \quad \lambda_i^{k+1} \leq c_i^q \tag{7.65}$$

or

$$\lambda_i^k = c_i^q \quad \textit{and} \quad \lambda_i^{k+1} < c_i^q. \tag{7.66}$$

We denote by $\mathcal{K}$ this set. Let $(k_1, k_2) \in \mathbb{N}^2$ such that $[k_1 - 1, k_2 + 1] \cap \mathcal{K} = \emptyset$. Then for $k \in [k_1, k_2]$ and $i \in I$ such that $\lambda_i^{k-1} \neq \lambda_i^k$

$$\lambda_i^k - \lambda_i^{k+1} \geq \frac{1}{N\alpha}(\frac{c_*}{c^*})^5 \max_{i' \in V(i)} (\lambda_{i'}^{k-1} - \lambda_{i'}^k) \tag{7.67}$$

Proof. By definition of λ^k, $\lambda_i^k - \lambda_i^{k+1} = \Lambda^i(\lambda_{-i}^{k-1}) - \Lambda^i(\lambda_{-i}^k)$. By construction, there exists $j \in [0, N-1]$ such that $\Lambda^i(\lambda_{-i}^{k-1}) = g_i^j(\lambda_{-i}^{k-1})$ and $\Lambda^i(\lambda_{-i}^k) = g_i^j(\lambda_{-i}^k)$. Then by monotony of g, $g_i^j(\lambda_{-i}^k) - g_i^j(\lambda_{-i}^{k-1})$ is lower bounded by

$$|\partial_{\lambda_{i'}} g_i^j|_\infty (\lambda_{i'}^{k-1} - \lambda_{i'}^k), \tag{7.68}$$

	Fixed Point	**CVX**
cost	83.2	83.195
time (s)	2.03	30.23

	Fixed Point	**CVX**
cost	4971.4	4971.4
time (s)	28.39	35.23

Table 7.2 – Results for a linear (a) and piecewise linear (b) instances of the problem solved with the fixed point algorithm and CVX.

for $i' \in V(i)$. We then take the $i' \in V(i)$ that maximizes $(\lambda_{i'}^{k-1} - \lambda_{i'}^{k})$ and use the previous lemma to get the result. □

7.4.5 Algorithm Implementation

We implemented this algorithm in Matlab. We use a dichotomy to compute the g_i^k. Note that for linear cost the analysis is similar. We define $g_i(\lambda_{-i})$ as the unique x such that $f_i(x, \lambda_{-i}) = 0$ and $x \geq 0$ and define Λ such that

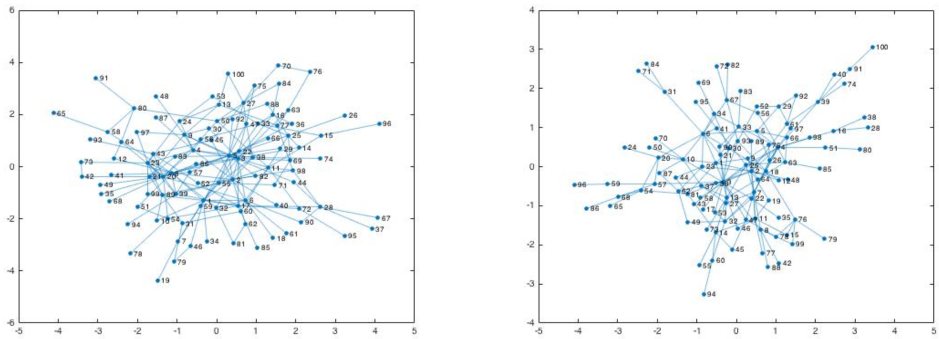
$$\Lambda_i(\lambda) = \min(c_i, g_i(\lambda_{-i})) \tag{7.69}$$

We perform some numerical comparisons with CVX, a package for specifying and solving convex programs [41, 42] for both linear and piecewise linear production cost functions. We generate a graph with 100 nodes connected randomly. To generate the graph, we use a Barabasi-Albert model [11] to ensure some scaling properties. The experiment is performed on a personal laptop (OSX, 4 Go,1.3 GHz Intel Core i5). The networks randomly generated to test the implementations are displayed in Figures 7.1a and 7.1b, and the results are summarized in Table 7.2.

Both CVX and the fixed point algorithm find the optimal value. The linear version of the fixed point algorithm is about ten times faster than the CVX resolution. Note that the algorithm could be distributed, since at each iteration, the computation at each node only depends on the values of the previous iteration. In addition, instead of computing the iterates of Λ at each step, we could use intermediate steps were we follow a decreasing direction $\Lambda(\lambda^k) - \lambda^k$ and choose $h > 0$ such that $\lambda_h^k = h(\Lambda(\lambda^k) - \lambda^k) + \lambda^k$ satisfies $\Lambda(\lambda_h^k) \leq \lambda_h^k$, which is easier to check than computing the g_i. This makes the algorithm similar to more standard descent based approaches (see [15]).

7.5 Two-agent allocation problem

We propose another algorithm for the piecewise linear allocation problem when the network is limited to two agents. This section is motivated by the need for efficient (both in speed and precision) allocation algorithms to numerically compute Bayesian Nash equilibria of the standard setting. Indeed, the natural next step of this work would be to proceed with numerical benchmarks, by comparing the Bayesian Nash equilibrium of the standard



(a) The network generated to test the linear implementation of the algorithm

(b) The network generated to test the generic implementation of the algorithm

setting and the solution of the optimal mechanism. In general the numerical search of such equilibrium requires to solve the allocation problems many times. A second motivation to present this piece of work here is the complementary insight it gives on the structure of the allocation problem.

7.5.1 First order condition

The allocation problem with two agents of slopes c_i^j and demand vector d_i is

Problem 5.

$$\begin{aligned}
\underset{q_i^j, h}{\textit{minimize}} \quad & \sum_{j\in J} c_1^j q_1^j + c_2^j q_2^j \\
\textit{subject to} \quad & \sum_j q_1^j - h \geq \frac{r}{2}h^2 + d_1 \\
& \sum_j q_2^j + h \geq \frac{r}{2}h^2 + d_2 \\
& q_i^j \leq \bar{q} \\
& q_i^j \geq 0 \\
& h \in \mathbb{R}.
\end{aligned} \tag{7.70}$$

We assume that N is big enough so that each agent could supply the whole amount without producing more than $\bar{q}N$. This is not a restrictive assumption as we could put very high marginal cost to model some capacity constraints. We denote for $h \in \mathbb{R}$ and $j \in J$

$$q_1(h) = d_1 + r\frac{h^2}{2} + h, \qquad q_2(h) = d_2 + r\frac{h^2}{2} - h, \tag{7.71}$$

$$\phi_i^j(h) = \min((q_i(h) - (j-1)\bar{q})^+, \bar{q}). \tag{7.72}$$

In order to reduce to an unconstrained problem, we assume that the constraints on the positiveness of the production are not bounding (else we can already conclude). This can be checked numerically by computing the gradients at $q_i(h) = 0$. Since the nodal constraints are bounding, we reformulate the problem:

$$\underset{h\in\mathbb{R}}{\text{minimize}} \quad C(h) = \sum_{j\in J} c_1^j \phi_1^j(h) + c_2^j \phi_2^j(h). \tag{7.73}$$

By definition of ϕ for $(i, j, h) \in I \times J \times \mathbb{R}$,

$$\phi_i^j(h) = \begin{cases} 0 & \text{if } q_i(h) \le (j-1)\bar{q} \\ q_i(h) - (j-1)\bar{q} & \text{if } q_i(h) \in [j-1, j]\bar{q} \\ \bar{q} & \text{if } q_i(h) \ge j\bar{q}. \end{cases} \tag{7.74}$$

So we can express the derivative of ϕ_i^j:

$$\partial\phi_i^j(h) = \begin{cases} 0 & \text{if } q_i(h) < (j-1)\bar{q} \\ rh + (-1)^{i+1} & \text{if } q_i(h) \in]j-1, j[\bar{q} \\ 0 & \text{if } q_i(h) > j\bar{q}. \end{cases} \tag{7.75}$$

The function C is convex, the expression of its subdifferential $\partial C(h)$ follows from (7.75):

$$\begin{cases} c_1^{j_1}(rh+1) + c_2^{j_2}(rh-1) & \text{if } q_i(h) \in](j_i-1), j_i[\bar{q} \\ [c_1^{j_1}, c_1^{j_1+1}](rh+1) + c_2^{j_2}(rh-1) & \text{if } q_2(h) \in](j_2-1), j_2[\bar{q} \text{ and } q_1(h) = j_1\bar{q} \\ c_1^{j_1}(rh+1) + [c_2^{j_2}, c_2^{j_2+1}](rh-1) & \text{if } q_1(h) \in](j_1-1), j_1[\bar{q} \text{ and } q_2(h) = j_2\bar{q} \\ [c_1^{j_1}, c_1^{j_1+1}](rh+1) + [c_2^{j_2}, c_2^{j_2+1}](rh-1) & \text{if } q_i(h) = j_i\bar{q}. \end{cases}$$

By the fifth assumption, we eliminate the last possibility. We denote

$$g(u) = \frac{1-u}{1+u}, \tag{7.76}$$

so that $0 \in \partial C(h)$ is equivalent to

$$\begin{cases} g(rh) = c_1^{j_1}/c_2^{j_2} & \text{if } q_i(h) \in]j_i-1, j_i[\bar{q} \\ g(rh) \in [\frac{c_1^{j_1}}{c_2^{j_2}}, \frac{c_1^{j_1+1}}{c_2^{j_2}}] & \text{if } q_2(h) \in]j_2-1, j_2[\bar{q} \text{ and } q_1(h) = j_1\bar{q} \\ g(rh) \in [\frac{c_1^{j_1}}{c_2^{j_2+1}}, \frac{c_1^{j_1}}{c_2^{j_2}}] & \text{if } q_1(h) \in]j_1-1, j_1[\bar{q} \text{ and } q_2(h) = j_2\bar{q} \end{cases} \tag{7.77}$$

We denote

$$q_1^{-1}(x) = -\frac{1}{r} + \sqrt{\frac{1}{r^2} - \frac{2}{r}(d_1 - x)} \quad \text{and} \quad q_2^{-1}(x) = \frac{1}{r} - \sqrt{\frac{1}{r^2} - \frac{2}{r}(d_2 - x)}, \tag{7.78}$$

and

$$j_i(h) = \lceil \frac{q_i(h)}{\bar{q}} \rceil. \tag{7.79}$$

By (7.77), $0 \in \partial C(h)$ is equivalent to one of those propositions being true:

$$\begin{cases} \exists j_1, j_2 \quad q_i(h) \in]j_i - 1, j_i[\bar{q} & \text{and } h = g(\frac{c_1^{j_1}}{c_2^{j_2}})/r \\ \exists j_1, \quad g(rh) \in [\frac{c_1^{j_1}}{c_2^{j_2(h)}}, \frac{c_1^{j_1+1}}{c_2^{j_2(h)}}] & \text{and } h = q_1^{-1}(j_1\bar{q}) \\ \exists j_2, \quad g(rh) \in [\frac{c_1^{j_1(h)}}{c_2^{j_2+1}}, \frac{c_1^{j_1(h)}}{c_2^{j_2}}] & \text{and } h = q_2^{-1}(j_2\bar{q}). \end{cases} \tag{7.80}$$

We then use the fact that g is idempotent: $g(u) = x \Leftrightarrow g(x) = u$. We obtain:

$$0 \in \partial C(h) \Leftrightarrow \begin{cases} \exists j_1, j_2 \quad h \in q_i^{-1}(](j_i - 1), j_i[\bar{q}) & \text{and } h = g(\frac{c_1^{j_1}}{c_2^{j_2}})/r \\ \exists j_1, \quad rh \in [g(\frac{c_1^{j_1+1}}{c_2^{j_2(h)}}), g(\frac{c_1^{j_1}}{c_2^{j_2(h)}})] & \text{and } h = q_1^{-1}(j_1\bar{q}) \\ \exists j_2, \quad rh \in [g(\frac{c_1^{j_1(h)}}{c_2^{j_2}}), g(\frac{c_1^{j_1(h)}}{c_2^{j_2+1}})] & \text{and } h = q_2^{-1}(j_2\bar{q}). \end{cases} \tag{7.81}$$

We denote, for $(i, j) \in I \times J$ and $(j_1, j_2) \in J^2$:

$$a_i^j = q_i^{-1}(j\bar{q}) \text{ and } b_{j_1,j_2} = g(c_1^{j_1}/c_2^{j_2})/r. \tag{7.82}$$

Those two quantities only depend on the problem data. We point out that a_i^j corresponds to the value of h when we set $q_i = j\bar{q}$ and b_{j_1,j_2} corresponds to the optimal value of h when $q_i \in](j_i - 1), j_i[\bar{q}$. We sum up with the following Lemma:

Lemma 7.27. *There exist $(j_1, j_2) \in J^2$ such that one of those propositions is true:*

$$b_{j_1,j_2} \in]a_i^{j_1-1}, a_i^{j_1}[\cap]a_i^{j_2}, a_i^{j_2-1}[\tag{7.83}$$

$$a_1^{j_1} \in [b_{j_1+1, j_2(a_1^{j_1})}, b_{j_1, j_2(a_1^{j_1})}] \tag{7.84}$$

$$a_2^{j_2} \in [b_{j_1(a_2^{j_2}), j_2}, b_{j_1(a_2^{j_2}), j_2+1}]. \tag{7.85}$$

Then the optimal value of h is respectively b_{j_1,j_2}, $a_1^{j_1}$ and $a_2^{j_2}$.

7.5.2 Algorithm

We denote by c_i^- the copy of the vector c_i with the first coordinate removed, and q_i the total production of agent i. We denote by $q_1(d, c_1, c_2)$ and $q_2(d, c_1, c_2)$ the optimal production allocation when the demand is d at both node and the cost vectors are c_1 and c_2.

Lemma 7.28. *If* $q_1(d, \mathbf{c}_1, \mathbf{c}_2) \geq \bar{q}$ *and* $q_2(d, \mathbf{c}_1, \mathbf{c}_2) \geq \bar{q}$*, then*

$$q_1(d, \mathbf{c}_1, \mathbf{c}_2) = q_1(d - \bar{q}, \mathbf{c}_1^-, \mathbf{c}_2^-) + \bar{q} \quad \text{and} \quad q_2(d, \mathbf{c}_1, \mathbf{c}_2) = q_2(d - \bar{q}, \mathbf{c}_1^-, \mathbf{c}_2^-)) + \bar{q}. \tag{7.86}$$

Proof. Fix $q_i^1 = \bar{q}$, the resulting optimization problem is equivalent to $P(d - \bar{q}, \mathbf{c}_1^-, \mathbf{c}_2^-)$. □

We set

$$F(\lambda_1, \lambda_2) = d + \frac{1}{r}\frac{\lambda_2 - \lambda_1}{\lambda_1 + \lambda_2} + \frac{1}{2r}\{\frac{\lambda_2 - \lambda_1}{\lambda_1 + \lambda_2}\}^2, \tag{7.87}$$

which is the 2-agent equivalent of F_i. We already know that if (λ_1, λ_2) are the solution of the dual, then $q_1 = F(\lambda_1, \lambda_2)$ and $q_2 = F(\lambda_2, \lambda_1)$. The main result of this part is:

Theorem 7.29. *If* $c_1^1 < c_2^1$ *and the second and third propositions of Lemma 7.27 are not satisfied, then let* k *be the smallest element of* $J \cap \{0\}$ *such that*

$$F(c_1^{k+1}, c_2^1) \leq k\bar{q} \text{ (A)} \quad \text{or} \quad F(c_2^1, c_1^k) > \bar{q} \text{ (B)} \tag{7.88}$$

then

- *if (B), then* $q_1(d, \mathbf{c}_1, \mathbf{c}_2) = q_1(d - \bar{q}, \mathbf{c}_1^-, \mathbf{c}_2^-) + \bar{q}$ *and* $q_2(d, \mathbf{c}_1, \mathbf{c}_2) = q_2(d - \bar{q}, \mathbf{c}_1^-, \mathbf{c}_2^-)) + \bar{q}$.
- *else,* $q_1 = F(c_1^k, c_2^1)$ *and* $q_2 = F(c_2^1, c_1^k)$

Proof. If (B), then we show that $q_2 \geq \bar{q}$. Indeed, if we assume $q_2 < \bar{q}$, then since we have eliminated the corner solution cases $\lambda_2 = c_2^1$. If we assume in addition that $q_1 < (k-1)\bar{q}$, then $\lambda_1 < c_1^k$, then $q_1 = F(\lambda_1, \lambda_2) = F(\lambda_1, c_2^1) > F(c_1^k, c_2^1) > (k-1)\bar{q}$ because of the definition of k, which is absurd. So if $q_2 < \bar{q}$ then necessarily $q_1 > (k-1)\bar{q}$ (The case $q_1 = (k-1)\bar{q}$ is a corner solution case that has been eliminated by hypothesis). So $\lambda_1 > c_1^k$ so by (B) $q_2 = F(\lambda_2, \lambda_1) > F(c_2^1, c_1^k) > \bar{q}$ which is in contradiction with the assumption. So if (B), then $q_2 > \bar{q}$, and since $c_1^1 < c_2^1$, $q_1 > \bar{q}$.

Else, by definition, (A) is true. Note that $q_1 = F(c_1^k, c_2^1)$ and $q_2 = F(c_2^1, c_1^k)$ solve the linear problem with $c_1 = c_1^k$ and $c_2 = c_2^1$ and it is admissible. So by convexity, this is the solution. □

Combining this result with the previous subsection, we can build an algorithm that first checks that we do not have a corner solution, and then recursively computes the solution.

7.6 Conclusion

We have shown how to characterize and compute the mechanism design. In addition, the allocation problem for the optimal and the standard mechanism are the same. We have proposed an algorithm based on a fixed point to solve the allocation problem. This work raises some questions. Can we weaken the Assumptions used in this work? Can we estimate the social benefit of using such mechanism? How to build numerical benchmarks to compare the optimal mechanism and the standard setting? How to implement the optimal mechanism in practice? Which real markets enter in the framework described in §7.3.6?

7.7 Proof of Lemma 7.7

Proof. By definition

$$\begin{aligned}
X(a^1\dots a^{k-1},b,a^{k+1}\dots a^N)-X(a^1\dots a^{k-1},c,a^{k+1}\dots a^N)=\\
V(a^1\dots b\dots a^N)-V(a^1\dots c\dots a^N)+\\
\sum_{j\neq k}a^j[Q^j(a^1\dots b\dots a^N)-Q^j(a^1\dots c\dots a^N)]\\
+bQ^k(a^1\dots b\dots a^N)-cQ^k(a^1\dots c\dots a^N)\\
=\int_b^c Q^k(a^1\dots s\dots a^N)\mathrm{d}s+\sum_{j\neq k}a^j[Q^j(a^1\dots b\dots a^N)-Q^j(a^1\dots c\dots a^N)]\\
+bQ^k(a^1\dots b\dots a^N)-cQ^k(a^1\dots c\dots a^N).
\end{aligned}$$

We use (H1) for the last equality. Then we apply a telescopic formula

$$\begin{aligned}
X(a)-X(b)=X(a^1\dots a^N)-X(b^1,a^2\dots a^N)+\\
X(b^1,a^2\dots a^N)-X(b^1,b^2\dots a^N)+\dots\\
+X(b^1\dots b^{N^1},a^N)-X(b^1\dots b^N)\\
=\sum_{k=1}^N\Big(\int_{a^k}^{b^k}Q^k(b^1\dots s\dots a^N)\mathrm{d}s\Big)+\\
\sum_{k=1}^N\sum_{j<k}b^j[Q^j(b^1\dots b^{k-1},a^k,a^{k+1}\dots a^N)-Q^j(b^1\dots b^{k-1},b^k,a^{k+1}\dots a^N)]\\
+\sum_{k=1}^N\sum_{j>k}a^j[Q^j(b^1\dots b^{k-1},a^k,a^{k+1}\dots a^N)-Q^j(b^1\dots b^{k-1},b^k,a^{k+1}\dots a^N)]\\
+\sum_{k=1}^N a^kQ^k(b^1\dots b^{k-1},a^k,a^{k+1}\dots a^N)-b^kQ^k(b^1\dots b^{k-1}\dots b^k,a^{k+1}\dots a^N)
\end{aligned}$$

Reordering the last three terms, we get

$$\begin{aligned}
\sum_{j=1}^N\sum_{k>j}b^j[Q^j(b^1\dots b^{k-1},a^k,a^{k+1}\dots a^N)-Q^j(b^1\dots b^{k-1},b^k,a^{k+1}\dots a^N)]\\
+\sum_{j=1}^N\sum_{k<j}a^j[Q^j(b^1\dots b^{k-1},a^k,a^{k+1}\dots a^N)-Q^j(b^1\dots b^{k-1},b^k,a^{k+1}\dots a^N)]
\end{aligned}$$

$$+\sum_{j=1}^{N} a^j Q^j(b^1 \dots b^j - 1, a^j, a^{j+1} \dots a^N) - b^j Q^j(b^1 \dots b^{j-1} \dots b^j, a^{j+1} \dots a^N)$$

$$= \sum_{j=1}^{N} \{ b^j \sum_{k>j} [Q^j(b^1 \dots b^{k-1}, a^k, a^{k+1} \dots a^N) - Q^j(b^1 \dots b^{k-1}, b^k, a^{k+1} \dots a^N)]$$
$$+ a^j Q^j(b^1 \dots b^{j-1}, a^j, a^{j+1} \dots a^N) - b^j Q^j(b^1 \dots b^{j-1} \dots b^j, a^{j+1} \dots a^N) +$$
$$a^j \sum_{k<j} [Q^j(b^1 \dots b^{k-1}, a^k, a^{k+1} \dots a^N) - Q^j(b^1 \dots b^{k-1}, b^k, a^{k+1} \dots a^N)] \}$$
$$= \sum_{j}^{N} a^j Q^j(a^1 \dots a^N) - b^j Q^j(b^1 \dots b^N)$$

We end up with

$$X(a) - X(b) = \sum_{j=1}^{N} (a^j Q^j(a) - b^j Q^j(b) + \int_{a^j}^{b^j} Q^j(b^1 \dots b^{j-1}, t, a^{j+1} \dots a^N) \mathrm{d}t) \tag{7.89}$$

$\square$

7.8 On $\mathcal{S}$ and the regularity of q

Remember that the set $\mathcal{S}$ corresponds to the points of transition between the two possibilities defined by the first order condition (7.50):

$$\mathcal{S} = \{ c \in \boldsymbol{C}^n, q_i(c) = j\bar{q} \text{ and } \lambda_i(c) = c_{j'} \text{ for some } i \in I, j \in J, j' \in \{j, j+1\} \}. \tag{7.90}$$

Our first goal is to show that $\mathcal{S}$ is a finite union of sets of zero measure (Lemmas 7.30 and 7.32). To do so, we apply the implicit functions theorem. From this we deduce the regularity of q (proof of Lemma 7.16). For any I_A, I_B partition of I, and $I_C \subset I_B$ not empty, $j \in J^I$ and $j' \in J^I$ such that for all i, $j' \in \{j_i, j_i + 1\}$, we denote by $S(I_A, I_B, I_C, j, j')$ the set

$$\left\{ c \in \boldsymbol{C}^n \text{ such that for any } i \in I \begin{cases} i \in I_A \Rightarrow & \lambda_i(c) = c_i^{j'_i} \text{ and } q_i(c) \notin \mathbb{N}\bar{q} \\ i \in I_B \Rightarrow & q_i(c) = j_i \bar{q} \\ i \in I_C \Rightarrow & \lambda_i(c) = c_i^{j'_i} \end{cases} \right\}. \tag{7.91}$$

For an element c of such set, we denote by M the matrix

$$M(c) = \left(\frac{\partial F_i(\lambda(c))}{\partial \lambda_j} \right)_{(i,j) \in I_B}. \tag{7.92}$$

We need to study the invertibility of M to apply the implicit functions theorem (Lemma 7.31). Note that the function S is defined on a finite set. We use the image of S to show that the measure of $\mathcal{S}$ with respect to the Lebesgue measure is zero. We first show in the next lemma that $\mathcal{S}$ is included in the finite union of the $S(I_A, I_B, I_C, j, j')$ family. Then we will show that each element of this family has a measure equal to zero.

Lemma 7.30. $\mathcal{S} \subseteq \cup S(I_A, I_B, I_C, j, j')$

Proof. Take $c \in \mathcal{S}$, then by definition of $\mathcal{S}$, there exist $i \in I$, $j \in J$ and $j' \in \{j, j+1\}$ such that $q_i(c) = j\bar{q}$ and $\lambda_i(c) = c_{j'}$, so I_C is not empty. By Lemma 7.14, for all $i \in I$, i is in I_A or I_B. So we have a set $S(I_A, I_B, I_C, j, j')$ such that c is in this set, so $\mathcal{S}$ is included in the union of those sets. □

Lemma 7.31. *For any $c \in \mathcal{C}^n$ the matrix $M(c)$ is invertible.*

Proof. Assume that there are some coefficients α_i such that $\sum_i \alpha_i M_i = 0$ where M_i is the ith column of M. Then by (7.34) and (7.35), the ith row of this relation writes:

$$\alpha_i \sum_{j \in V(i)} \frac{\lambda_j^2}{r_{i,j}(\lambda_i + \lambda_j)^3} = \sum_{j \in V(i), j \in I_B} \frac{\alpha_j \lambda_i \lambda_j}{r_{i,j}(\lambda_i + \lambda_j)^3}. \tag{7.93}$$

We denote $b_{i,j} = \frac{\lambda_j^2 \lambda_i}{r_{i,j}(\lambda_i+\lambda_j)^3}$ and $a_i = \frac{\alpha_i}{\lambda_i}$. Then (7.93) is equivalent to

$$a_i = \sum_{j \in V(i), j \in I_B} a_j \frac{b_{i,j}}{\sum_{k \in V(i)} b_{i,k}} \tag{7.94}$$

Considering the biggest a_i, we get that all a_i are equal by convexity, and so either all are equal to zero or

$$\sum_{j \in V(i)} b_{i,j} = \sum_{j \in V(i), j \in I_B} b_{i,j} \tag{7.95}$$

which is not the case since I_A is not empty by the fifth assumption. □

Next we show that $S(I_A, I_B, I_C, j, j')$ has a zero Lebesgue measure.

Lemma 7.32. *For any I_A, I_B partition of I, and $I_C \subset I_B$ not empty, $j \in J^I$ and $j' \in J^I$ such that for all i, $j' \in \{j, j+1\}$, the measure of the set $S(I_A, I_B, I_C, j, j')$ is zero.*

Proof. We assume in the market description that it is not possible to produce a multiple $\bar{q}$ at each node and satisfy exactly the nodal constraints (fifth assumption). Therefore it is not possible that $I_B = I$, so I_A is not empty. By definition of $S_{I_A, I_B, I_C, j, j'}$, for all $i \in I_B$,

$$F_i(c_{I_A}^{j'}, \lambda_{I_B}(c)) = q_i(c) = j_i \bar{q}, \tag{7.96}$$

which is a system of equations in λ_{I_B} parametrized by $c_{I_A}^{j'}$. Let $c \in \boldsymbol{C}$ such that the system is satisfied, by Lemma 7.31, we can apply the implicit function theorem, so there is a ball around c in which $S(I_A, I_B, I_C, j, j')$ is included in a smooth surface. By compacity of $\boldsymbol{C}$, we can choose a sequence dense in $S(I_A, I_B, I_C, j, j')$. We apply the result to each element of this sequence. By density, $S(I_A, I_B, I_C, j, j')$ is a countable union of smooth surfaces. Therefore the measure of $S(I_A, I_B, I_C, j, j')$ is zero. □

A direct consequence of Lemma 7.32 and Lemma 7.30 is

Lemma 7.33. *The measure of $\mathcal{S}$ is zero.*

We proceed with the proof of Lemma 7.16.

of lemma 7.16. Let $c = (c_1 \dots c_n) \in \boldsymbol{C}^n \backslash \mathcal{S}$. Let us show that q is infinitely differentiable at c. We consider the two assertions:

$$A_i = \text{"}\exists k_i, \quad F_i(\lambda(c)) \in]k_i - 1, k_i[\bar{q} \quad \text{and} \quad \lambda_i = c_i^k\text{"}$$

$$B_i = \text{"}\exists k_i, \quad F_i(\lambda(c)) = k_i \bar{q} \quad \text{and} \quad \lambda_i \in]c_i^k, c_i^{k+1}[\text{"}$$

By Lemma 7.14 and by defintion of $\mathcal{S}$, for any $i \in I$ either A_i or B_i is true, but never both. We denote by I_A (resp. I_B) the set of elements of I for which A_i (resp. I_B) is true. If A_i is true for all i then there is a neighborhood V of c such that for any element $\tilde{c}$ of V, $F_i(\tilde{c}) \in]k_i - 1, k_i[\bar{q}$, therefore on V, $\lambda(\tilde{c}) = \tilde{c}$.

Else I_B is not empty and by definition of B_i

$$\forall i \in I_B \quad F_i(\lambda_{I_A}, \lambda_{I_B}) = \bar{q} j_i, \tag{7.97}$$

which we can see as an equation in λ_{I_B} parametrized by λ_{I_A}. This equation is satisfied at $\lambda(c)$. If we denote by M the matrix

$$M = \left(\frac{\partial F_i(\lambda(c))}{\partial \lambda_j} \right)_{(i,j) \in I_B}, \tag{7.98}$$

then M is invertible (see lemma 7.31), the implicit function theorem applies and there exists a function λ_{I_B} so that in a neighborhood V of c, for all $i \in I_B$, we have $F_i(\lambda_{I_A}, \lambda_{I_B}(\lambda_{I_A})) = \bar{q} k_i$. Moreover, since F_i is C^∞ on $[c_*, c^*]^n$, λ_{I_B} is C^∞ on V. Then if $\tilde{c} \in V$, $(\tilde{c}, \lambda_{I_B}(\tilde{c}))$ checks the first order condition so by uniqueness $c_{I_A}, \lambda_{I_B}(\tilde{c})$ is the dual solution, and so, $q_i = F_i(\lambda_{I_B}(\tilde{c}), \tilde{c})$ for all $i \in I$ on V, so q_i is C^∞ at c. This concludes the proof of the first part of the lemma.

The continuity of q comes from Berge maximum principle (see Theorem 9.17 in [85]) in a convex setting. □

The next lemma is an important component for the proof of Theorem 7.10.

Lemma 7.34. *Let $i \in I$ and $c_i \in C_i$, then the Lebesgue measure of the set*

$$\mathcal{S}_i(c_i) = \{c_{-i} \in C_{-i}, (c_i, c_{-i}) \in \mathcal{S}\} \tag{7.99}$$

is zero.

Proof. Using Lemma 7.30, $\mathcal{S}_i(c_i) \subseteq \{c_{-i} \in C_{-i}, (c_i, c_{-i}) \in \cup S(I_A, I_B, I_C, j, j')\}$. So let $c_{-i} \in \mathcal{S}_i(c_i)$, I_A, I_B a partition of I, and $I_C \subseteq I_B$ not empty, and j, j' such that $(c_i, c_{-i}) \in S(I_A, I_B, I_C, j, j')$. There are three possible cases:

- $i \in I_A$ then as explained in the proof of Lemma 7.32, $S(I_A, I_B, I_C, j, j')$ is locally a surface parametrized by c_i so by projection over an hyperplane of the type $c_i = x$ it also a surface in $\boldsymbol{C}_{-i}$.

- $i \in I_B \backslash I_C$ locally, q is independant of c_i so if $S(I_A, I_B, I_C, j, j') \cap (c_i, \mathcal{S}_i(c_i))$ is of strictly positive measure, then $S(I_A, I_B, I_C, j, j')$ has also a strictly positive measure in $\boldsymbol{C}^n$, since this is not true, $S(I_A, I_B, I_C, j, j') \cap (c_i, \mathcal{S}_i(c_i))$ is of zero measure in the neighborhood.

- Else $i \in I_C$, which is the tricky part. First by definition of I_C, for any element c of $S(I_A, I_B, I_C, j, j')$, $q_i(c) = j_i \bar{q}$ and $\lambda_i(c) = c_i^{j'_i}$. Without loss of generality, we assume $j'_i = j_i$, the other case can be treated similarly. Then we make the observation that we do not modify the c_{-i} of $S(I_A, I_B, I_C, j, j')$ if we set $c_i^{j+1} = c_i^j$. Since we are interested in $S(I_A, I_B, I_C, j, j') \cap (c_i, \mathcal{S}_i(c_i))$, we can assume without loss of generality that $c_i^{j+1} = c_i^j$. Then we have reduced to the case $i \in I_A$.

We conclude as in the proof of Lemma 7.32. □

Chapter 8

On a class of bidding games whose dynamics converges to the unique pure equilibrium

This is a joint work with Alejandro Jofré

Contents

We introduce a class of Bayesian bidding games for which we prove that the set of pure Nash equilibria is a (non-empty) sublattice and we give a sufficient condition for uniqueness that is often verified in the context of markets with inelastic demand. We propose a dynamics that converges to the equilibrium set and derive a scheme to compute the extreme Nash equilibria. This scheme is an alternative to the more standard best reply dynamics and fictitious play. It shows better converging properties on the electricity auction instance that motivated its introduction. We apply this framework to the wholesale electricity procurement auctions that motivated this study.

8.1 Introduction

The interaction between firms (or individuals) competing on price to maximize their profits in an imperfect information environment constitutes a Bayesian game. Very often most of the private information concerns the production costs of the firms. Such situations occur for instance in procurement auctions, commodity markets and oligopolies. It is then natural to ask about the Nash equilibria of such kind of games. Do we have a guaranty of existence? Is the equilibrium unique? Do we have an algorithm to compute it? Those are generically hard questions when the classical results of game theory are not applicable. In this work we identify a class of games for which some of those questions can be answered using basic tools of lattice theory.

An essential observation is that a bid increase by one firm is very often an incitation for all other firms to increase their bids. The exploitation of such monotonic behaviors, summarized in the notion of strategic complementarity, is central in the study of many pricing games (and more generally in Bayesian games) and explains the intensive use of lattice theory in a literature that has many interesting results of pure Nash equilibria existence. As opposed to what is required in most of game theory literature, those results usually do not need payoff functions to be quasiconcave. Those existence results, depending on the fixed point theorem over which they are constructed, differ by the underlying assumptions and the additional information they provide on the equilibria set.

What follows builds on a rich literature. We now briefly review some of its major achievements.

Optimization on lattices The development of this work was strongly inspired by a book from Topkis [87] in which the author surveys some fundamental results of monotone comparative statics, in particular [86, 68, 84, 33]. Monotone comparative statics concerns settings where a parametrized collection of optimal decisions are monotone in the parameter.

Bayesian games and equilibrium existence The class we introduce belongs to the large class of Bayesian games, and more specifically has some strategic complementarity properties.

In [91] Vives proves the existence of a pure Nash equilibrium for Bayesian games with general action and type spaces in action with a Tarsky fixed-point theorem on lattices. More precisely, the equilibrium set is a non-empty complete lattice. Payoffs need to be supermodular.

In [5], Athey shows the existence of a monotone pure strategy equilibrium for finite Bayesian games satisfying a single-crossing condition. Actions and types should be one dimensional. The demonstration relies on Kakutani's fixed-point theorem. Athey generalizes the result to a continuum action set when the payoffs are continuous.

In [65], McAdams extends Athey results to a multidimensional setting. Like in [5] the extension to continuum action sets is obtained by taking the limit of finite approximations equilibria. McAdams, like Athey, uses a single-crossing condition, combined with a quasisupermodularity condition, and then applies the Glicksberg's fixed point theorem.

In [80], Reny and Zamir show that first price, one unit auctions with affiliated types and interdependent values have a pure equilibrium; they use Athey's [5] result for their demonstration. They combine a result by [69] with Athey [5] approach of limit of finite action grids.

Vives surveys in [92] the use of monotone comparative static tools for games with complementarities. He points out some interesting properties of games with strategic complementaries that are still valid in our framework: general strategic spaces, existence of pure Nash equilibria, specific structure of the set of equilibria, existence of a Best reply dynamics algorithm to compute the extremal equilibria. In the seventh section of the article, he discusses some aspects specific to Bayesian games.

In [90], Van Zandt and Vives give a constructive proof of the existence of a pure Nash equilibrium for Bayesian games satisfying a strategic complementarity condition. The action and type sets can be infinite dimensional. The payoffs need to be supermodular and satisfy some increasing difference property. In addition, they show that one can compute such equilibria by best-reply iteration. We propose a different approach that turns out to be more stable on the numerical benchmark we use in this study.

In [79] Philip J. Reny generalizes the results of Athey [5] and McAdams [65] on the existence of monotone pure strategy equilibria in generic Bayesian games. While Athey and McAdams proofs rely on the convexity of the best reply sets, Reny uses a fixed point theorem that relies on the notion of contractibility. He shows that the result applies when the payoff function is weakly quasi supermodular and satisfies a weak single crossing condition, and concludes that his result is strictly more general than [5] and [65]. In particular, the result can be applied when type and action sets are infinite dimensional. The payoff functions need to be continuous in the actions.

Equilibrium computation, best reply dynamics and fictitious play Many approaches to compute an equilibrium consist in mimicking the behaviors of the players when the game is repeated sequentially. As time goes on, each player takes a decision based on the previous iterations of the game. Basically, an approach is characterized by the *memory* of the players (do they remember a joint probability of actions, marginals…)

and the way they *choose* the next action. The two historical approaches are the Cournot's tâtonnements and Brown's fictitious play [23]. In the standard Cournot's tâtonnements (or Best Reply dynamics), actions are taken as best replies against the last actions of the other players. In fictitious play, the last action is modified *in the direction* of the Best Reply against the average of the other players past actions. There are no general results of fictitious play convergence for games with complementarities. Yet one may consult [14, 13, 66, 67]. Vives discusses in [92] the tâtonnements in a context very close to ours. The convergence is derived from some monotonicity properties but, as he points out, convergence cannot be ensured for an arbitrary starting point. Observe that most of these results are related to matrix games. We propose an alternative approach to fictitious play and Best Reply dynamics for a class of Bayesian games to which the electricity market introduced in [38] belongs.
In the next section, we introduce the game and present our main results. We illustrate those results on an example in §8.3. §8.4, §8.5 and §8.6 are dedicated to the proofs of the three main results: the existence of a Nash equilibrium, the uniqueness of the equilibrium and a convergence of a tâtonnement dynamics to the equilibrium. We discuss some possible extensions in the conclusion.

8.2 Game presentation and main results

8.2.1 Definitions

Definition 8.1 (Least Upper Bound ($\vee$) and Greatest Lower Bound ($\wedge$) , see [87]). *Let $\mathbb{X}$ be a partially ordered set, $\mathbb{X}' \subseteq \mathbb{X}$. We say that $\bar{x} \in \mathbb{X}$ is the least upper bound of $\mathbb{X}'$ when $\forall (x, x') \in \mathbb{X} \times \mathbb{X}' \quad \bar{x} \le x \iff x' \le x$. We say that $\underline{x} \in \mathbb{X}$ is the greatest lower bound of $\mathbb{X}'$ when $\forall (x, x') \in \mathbb{X} \times \mathbb{X}' \quad x \le \underline{x} \iff x \le x'$. For $(x, y) \in \mathbb{X}^2$, we denote by $x \wedge y$ and $x \vee y$ the greatest lower bound and the least upper bound of the pair $\{x, y\}$.*

Definition 8.2 (Lattice, Sublattice, Complete Lattice, see [87]). *A partially ordered set $\mathbb{X}$ is a lattice iff it contains a least upper bound and a greatest lower bound for each pair of its elements. A subset $\mathbb{X}' \subseteq \mathbb{X}$ is a sublattice if it contains a least upper bound and a greatest lower bound for each pair of its elements. A lattice in which each nonempty subset has a greatest lower bound and a least upper bound is complete.*

We will use the notion of increasing function in lattice, which differs from the usual definition. Observe that one may also encounter the term isotone in the literature.

Definition 8.3 (Increasing). *We say that a function f from two ordered sets is* increasing *if for all $x \le y$, $f(x) \le f(y)$.*

8.2.2 Game Presentation

Notations

The game $(\mathbb{I}, \mathbb{T}, \mathbb{B}, \Sigma, K, p)$ consists in a set of *players* $\mathbb{I} = 1 \ldots n$, $n \in \mathbb{N}$. For each player, there is a set of *types* $\mathbb{T}^i$ and a set of *bids* (or *action*) $\mathbb{B}^i$. Types and bids are included in a compact $[c_*, c^*]$ (where $c_* > 0$) such that $\mathbb{T} \subset \mathbb{B}$. The *strategies* are applications from $\mathbb{T}^i$ to $\mathbb{B}^i$. For each player, we denote by Σ^i his strategy set. For any $i \in \mathbb{I}$, the *demand response* K^i is a function from the bid set $\mathbb{B}$ to $[0, K^+]$, where $K^+ > 0$. We generically denote by σ^i the elements of the strategy set Σ^i, c^i the elements of $\mathbb{T}^i$ (because it can be interpreted as a production *cost* and we want to avoid any confusion with the time variable), and b^i the bids, elements of $\mathbb{B}^i$. We use the standard notation of game theory $-i$ to refer to all but player i. Last but not least, p^i is a probability density of support $\mathbb{T}^{-i}$.

For a strategy profile $\sigma^{-i} = (\sigma^1 \ldots \sigma^n)$, the expected ex-ante payoff Π^i of player i of type c^i bidding $b^i = \sigma^i[c^i]$ is

$$\Pi^i_{\sigma^{-i}}(b^i, c^i) = \int_{c^{-i} \in T^{-i}} \pi^i(b^i, c^i, \sigma^{-i}[c^{-i}]) p^{-i}(c^{-i}) \mathrm{d}c^{-i}, \tag{8.1}$$

with the *payoff* π^i defined for $(b^i, c^i, \sigma^{-i}, c^{-i}) \in \mathbb{B}^i \times \mathbb{T}^i \times \Sigma^{-i} \times \mathbb{T}^{-i}$ by

$$\pi^i(b^i, c^i, \sigma^{-i}[c^{-i}]) = (b^i - c^i) K^i(b^i, \sigma^{-i}(c^{-i})). \tag{8.2}$$

In this expression, K^i can be interpreted as the quantity i is asked to provide for a *marginal price profile* b^i when the other players bid the marginal prices $\sigma^{-i}[c^{-i}]$. So the integrand is the profit of this player if he has a marginal production cost c^i. The kernel K^i corresponds to the market (or auctioneer) response to the bids. We assume the Kernel to be continuous. In what follows we assume $\pi^i(b^i, c^i, \sigma^{-i}[c^{-i}])$ Lebesgue measurable with respect to c^{-i} for all $(b^i, c^i, \sigma^{-i}) \in \mathbb{B}^i \times \mathbb{T}^i \times \Sigma^{-i}$.

Definition 8.4 (Best Reply). *We denote by BR^i the Best Reply set-valued mapping from Σ^{-i} to the subsets of Σ^i such that for any $\sigma^{-i} \in \Sigma^{-i}$,*

$$BR^i(\sigma^{-i}) = \{\beta \in \Sigma^i : \quad \forall (c, \sigma) \in \mathbb{T}^i \times \Sigma^i, \quad \Pi^i_{\sigma^{-i}}(\beta[c^i], c^i) \geq \Pi^i_{\sigma^{-i}}(\sigma^i[c^i], c^i)\} \tag{8.3}$$

Definition 8.5 (Pure Nash Equilibrium). *A strategy profile $\sigma \in \Sigma$ is a Pure Nash Equilibrium if for any $i \in \mathbb{I}$, $\hat{\sigma}^i \in \Sigma^i$ and $c^i \in \mathbb{T}^i$*

$$\Pi^i_{\sigma^{-i}}(\sigma^i[c^i], c^i) \geq \Pi^i_{\sigma^{-i}}(\hat{\sigma}^i[c^i], c^i) \tag{8.4}$$

We use the partial order $\leq_{\Sigma^i}$ on Σ^i defined by

$$\forall (\sigma_1, \sigma_2) \in \Sigma^i, \quad (\sigma_1 \leq_{\Sigma^i} \sigma_2) \quad \text{iff} \quad (\forall c \in \mathbb{T}^i, \quad \sigma_1(c) \leq \sigma_2(c)). \tag{8.5}$$

We denote by $\leq_\Sigma$ the induced product order on Σ.

Assumption 5 (Kernel Monotonicity). *For any $i \in \mathbb{I}$ $K^i(b^i, b^{-i})$, is increasing in b^{-i} and decreasing in b^i.*

The *Kernel Monotonicity* assumption corresponds to the fact that the bidding occurs in a competitive setting, and the demand tends to go to the cheapest bidder.

Assumption 6 (Strict Increasing Differences). *For any $i \in \mathbb{I}$, $c \in \mathbb{T}^i$, set $\pi^i_c = \pi^i(., c, .)$. Then π^i_c satisfies the Strict Increasing Differences Property:*

$$\forall (b_1, b_2, b_1^{-i}, b_2^{-i}) \in \mathbb{B}^i \times \mathbb{B}^i \times \mathbb{B}^{-i} \times \mathbb{B}^{-i} \text{ such that } b_1 \leq b_2 \text{ and } b_1^{-i} \leq b_2^{-i} : \tag{8.6}$$

$$\pi^i_c(b_2, b_1^{-i}) - \pi^i_c(b_1, b_1^{-i}) < \pi^i_c(b_2, b_2^{-i}) - \pi^i_c(b_1, b_2^{-i}). \tag{8.7}$$

8.2.3 Main results

Theorem 8.6 (Existence of a Pure Nash Equilibrium). *The set of pure Nash equilibra is a nonempty complete lattice.*

Theorem 8.7 (Uniqueness Sufficient Condition). *If*

- *for any $\alpha > 0$, $(x, y) \in \mathbb{B}$,*
$$K(\alpha x, \alpha y) = K(x, y), \tag{8.8}$$

- *for any $\sigma \in \Sigma$ equilibrium strategy profile, $i \in \mathbb{I}$, and $(c_1, c_2) \in [c_*, c^*]\times]0, c^*]$ such that $c_1 > c_2$*

$$\beta_1 > \beta_2, \forall (\beta_1 \beta_2) \in \arg\max_{b \in \mathbb{B}^i} \Pi^i_{c_1}(b, \sigma^{-i}) \times \arg\max_{b \in \mathbb{B}^i} \Pi^i_{c_2}(b, \sigma^{-i}) \tag{8.9}$$

- *for any $\sigma \in \Sigma$ equilibrium strategy profile, $i \in \mathbb{I}$, $\inf BR^i(\sigma^{-i})(c)$ and $\sup BR^i(\sigma^{-i})(c)$ are continuous*

- *for any $(\sigma_1, \sigma_2) \in \Sigma$ equilibrium strategy profile for all $(i, c) \in \mathbb{I} \times \mathbb{T}$,*

$$\sigma_2^i(c^*) \frac{\sigma_1^i(c)}{\sigma_2^i(c)} \leq b^* \tag{8.10}$$

then the set of pure Nash equilibria is a singleton.

Relation (8.8) means that if every player multiplies his bid by the same constant, then the resulting allocation does not change. This should be satisfied with inelastic demand.

Theorem 8.8 (Converging dynamics). *Assume:*

- *$b \in \mathbb{B}^i \to \Pi^i_{\sigma^{-i}}(b, c)$ is C^2 and $b \to \partial_b \Pi^i_{\sigma^{-i}}(b, c)$ is uniformly Lipschitz for all $(c, \sigma) \in \mathbb{T}^i \times \Sigma^{-i}$.*

- $\sigma_*(\mathbb{T})$ *is included in* $]b_*, b^*[^{\mathbb{I}}$, *where* σ_* *is the smallest equilibrium's strategy profile.*

- *For any* $(\sigma^{-i}, c) \in \Sigma^i \times \mathbb{T}^i$, $\Pi^i_{\sigma^{-i}}(b, c)$ *is concave in* b.

Then the solution to the system of differential equations

$$\forall (i,c) \in \mathbb{I} \times \mathbb{T}^i \quad \partial_t \sigma^i(c,t) = \partial_b \Pi^i_{\sigma(,t)^{-i}}(\sigma^i(c,t), c) \tag{8.11}$$

$$\sigma^i(c,0) = c \tag{8.12}$$

converges to the smallest equilibrium strategy profile σ_* *as* t *goes to* $+\infty$.

8.3 Application

Consider a simple geographical electricity market with two nodes (Node 1 and Node 2). The nodes are connected by a line through which electricity can be sent. There is a known (inelastic) demand d at each node. We assume marginal prices to be constant, within a compact $[b_*, b^*]$. We consider that there is one producer at each node, namely a_1 and a_2. An independent operator has to allocate the production to meet the demand at each node and to minimize the total cost paid to the producers. When a quantity h of electricity is sent through the line, rh^2 is lost in the process (Joule effect). The players of the Bayesian game are the two producers, who want to maximize their expected profit (we say *expected* because they do not know the other player production cost). Solving the independent operator problem, we get

$$K^i(b^i, b^{-i}) = \begin{cases} F(b^i, b^{-i}) & \text{if } F(b^i, b^{-i}) \geq 0 \text{ and } F(b^{-i}, b^i) \geq 0 \\ 0 & \text{if } F(b^i, b^{-i}) < 0 \\ \bar{q} & \text{if } F(b^{-i}, b^i) < 0, \end{cases} \tag{8.13}$$

where

$$F(x,y) = d + \frac{1}{2r}\left(\frac{x-y}{x+y}\right)^2 - \frac{1}{r}\frac{x-y}{x+y} \quad \text{and} \quad \bar{q} = 2\frac{1-\sqrt{1-2dr}}{r}. \tag{8.14}$$

Therefore Assumption 5 is satisfied. Observe that the increasing difference property is not satisfied (see Picture 8.1) everywhere. In the following, we assume that $F(b^*, b_*) \geq 0$. Therefore the corner solutions are not to be considered and $K^i(b^i, b^{-i}) = F(b^i, b^{-i})$. Moreover, we assume $b^* < 2b_*$. Then the payoff writes $\pi^i_c(b^i, b^{-i}) = (b^i - c)K^i(b^i, b^{-i})$. Therefore

$$\partial_{xy}\pi^i_c(x,y) = \frac{4y}{r(x+y)^4}\left(x(2y-x) + c(2x-y)\right) > 0 \tag{8.15}$$

Therefore the strict increasing differences condition 6 is satisfied. So Theorem 8.6 applies.

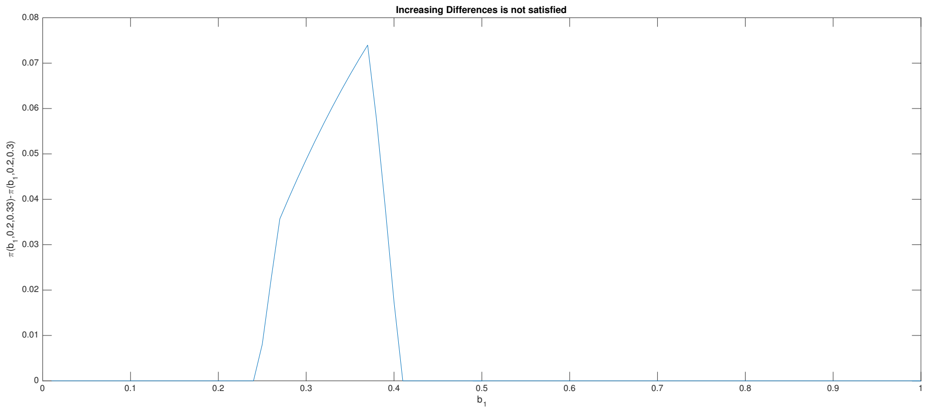


Figure 8.1 – The increasing difference property is not satisfied, we use something weaker

Next it is clear that the scaling property (8.8) of Theorem 8.7 is satisfied. Using

$$\partial_{xc}\pi_c^i(x,y) = \frac{4y^2}{r(x+y)^4} > 0 \tag{8.16}$$

we show as in [90] that (8.9) is satisfied.

Observe that

$$\partial_{xx}\pi_c^i(x,y) = \frac{4y}{r(x+y)^4}\,(x-3c-2y) < 0 \tag{8.17}$$

Therefore π is strictly concave with respect to its first variable, and by integration, so is Π. Therefore by Berge theorem, the best reply is continuous in c. To show that (8.10), we first observe that in full information, symmetric setting, if the cost is c for both players, then the best reply is $\frac{c}{1-2rd}$. We combine this observation with the monotonicity of the best reply with respect to the type and the opponent strategy to conclude that any optimal bid b should satisfy

$$b \in [\frac{c_*}{1-2rd}, \frac{c^*}{1-2rd}] \tag{8.18}$$

Therefore condition (8.10) is satisfied for b^* large enough and Theorem 8.7 applies.

We have already checked that all conditions to apply Theorem 8.8 were satisfied.

8.4 Existence of a Nash Equilibrium

8.4.1 General Preliminary Results

Definition 8.9 (Strict Single crossing property, see [87]). *Let $\mathbb{X}$, $\mathbb{Y}$ and $\mathbb{Z}$ be partially ordered set, let $f(x,z)$ be a function of a subset $\mathbb{S}$ of $\mathbb{X}\times\mathbb{Z}$ into $\mathbb{Y}$, then $f(x,z)$ satisfies the strict single crossing property in (x,z) on $\mathbb{S}$ if for all x_1 and x_2 in $\mathbb{X}$ and z_1, z_2 in $\mathbb{Z}$ with $x_1 < x_2$, $z_1 < z_2$ and $(x_1,x_2)\times(z_1,z_2)$ being a subset of $\mathbb{S}$, $f(x_1,z_1) \le f(x_2,z_1)$ implies*

$f(x_1, z_2) < f(x_2, z_2)$.

Lemma 8.10. *The application* $(b, \sigma^{-i}) \to \Pi^i_{\sigma^{-i}}(b, c)$ *satisfies the Strict Single Crossing Property.*

Proof. Let $b_1 < b_2 \in \mathbb{B}$ and $a_1^{-i} < a_2^{-i} \in \mathbb{A}^{-i}$ such that $\Pi^i_c(b_1, \sigma_1^{-i}) \leq \Pi^i_c(b_2, \sigma_1^{-i})$. By increasing differences, for any $c^{-i} \in \mathbb{T}^{-i}$, we have

$$\pi^i_c(b_2, \sigma_1^{-i}(c^{-i})) - \pi^i_c(b_1, \sigma_1^{-i}(c^{-i})) < \pi^i_c(b_2, \sigma_2^{-i}(c^{-i})) - \pi^i_c(b_1, \sigma_2^{-i}(c^{-i})), \tag{8.19}$$

so multiplying by $p^{-i}(c^{-i})$, and integrating, we get

$$0 \leq \Pi^i_c(b_2, \sigma_1^{-i}) - \Pi^i_c(b_1, \sigma_1^{-i}) < \Pi^i_c(b_2, \sigma_2^{-i}) - \Pi^i_c(b_1, \sigma_2^{-i}), \tag{8.20}$$

where the first inequality comes from the hypothesis. □

Definition 8.11 (Quasisupermodularity, see [87]). *Let $\mathbb{X}$ be a lattice, $\mathbb{Z}$ a partially ordered set and f a function from $\mathbb{X}$ to $\mathbb{Z}$, then we say that f is quasisupermodular if for all x_1 and x_2 from X, $f(x_1 \wedge x_2) \leq f(x_1)$ implies $f(x_2) \leq f(x_1 \vee x_2)$ and $f(x_1 \wedge x_2) < f(x_1)$ implies $f(x_2) < f(x_1 \vee x_2)$.*

Lemma 8.12 (Quasi Supermodularity). *For any $i \in \mathbb{I}$, $c \in \mathbb{T}$, $\sigma^{-i} \in \Sigma^{-i}$, $b \to \Pi^i_{\sigma^{-i}}(c, b)$ is quasisupermodular.*

Proof. Trivial since we are in a monodimensional setting. □

8.4.2 Existence

We will need the following result:

Theorem 8.13 (Increasing Optimal Strategies (see [87]) page 83). *Suppose that $\mathbb{X}$ is a lattice, $\mathbb{Z}$ is a partially ordered set, $\mathbb{S}_z$ is a subset of $\mathbb{X}$ for each z in $\mathbb{Z}$, $\mathbb{S}_z$ is increasing in z on $\mathbb{Z}$, $f(x, z)$ is quasisupermodular in x on X for each z in $\mathbb{Z}$, and $f(x, z)$ satisfies the strict single crossing property in (x, z) on $\mathbb{X} \times \mathbb{Z}$. If z_1 and z_2 are in $\mathbb{Z}$, $z_1 < z_2$, x_1 is in $\arg\max_{x \in \mathbb{S}_{z_1}} f(x, z_1)$ and x_2 is in $\arg\max_{x \in \mathbb{S}_{z_2}} f(x, z_2)$, then $x_1 \leq x_2$. Hence if one picks any x_z in $\arg\max_{x \in \mathbb{S}_z}$ for each z in $\mathbb{Z}$ with argmax non empty, then x_z is increasing in z on $\{z : z \in \mathbb{Z}, \quad \arg\max_{x \in \mathbb{S}_z} f(x, z) \quad \text{non empty}\ \}$.*

Definition 8.14 (Induced Set ordering). *Let $\mathbb{X}$ be a lattice, $\mathbb{X}_1$ and $\mathbb{X}_2$ two non empty subsets of $\mathbb{X}$. We say that $\mathbb{X}_1 \sqsubseteq \mathbb{X}_2$ iff for any $(x_1, x_2) \in \mathbb{X}_1 \times \mathbb{X}_2$, $x_1 \wedge x_2 \in \mathbb{X}_1$ and $x_1 \vee x_2 \in \mathbb{X}_2$.*

Combining Lemma 8.10, Lemma 8.12 with Theorem 8.13, we get:

Lemma 8.15. *For any $i \in \mathbb{I}$, for any $(c, \sigma_1^{-i}, \sigma_2^{-i}) \in \mathbb{T}^i \times \Sigma_2^{-i}$, such that $\sigma_1^{-i} < \sigma_2^{-i}$, for any $(\beta_1, \beta_2) \in BR^i(\sigma_1^{-i}) \times BR^i(\sigma_2^{-i})$*

$$\beta_1(c) \leq \beta_2(c) \tag{8.21}$$

In particular, BR^i is increasing in the induced set ordering on Σ^i. In addition, for any $\sigma^{-i} \in \Sigma^{-i}$, $BR^i(\sigma^{-i})$ is a complete sublattice.

Proof. By continuity and compactness, for any $c \in \mathbb{T}^i$, $\arg\max_b \Pi_c^i(b, \sigma^{-i})$ is nonempty. Since Π_c^i is quasisupermodular (Lemma 8.12) and satisfies the strict single crossing property (Lemma 8.10). Therefore by Theorem 8.13,(8.21) is satisfied for any c. So any selection of BR^i is increasing in σ^{-i}. Therefore BR^i in increasing in the induced set ordering. Since Π is continous in b (by continuity of the integrand) and supermodular in b (since b is monodimensionnal), by corollary 2.7.1 of [87], $BR^i(\sigma^{-i})[c]$ is a complete sublatice of $\mathbb{B}^i$, therefore $BR^i(\sigma^{-i})$ is a complete sublatice of Σ^i. □

Theorem 8.16 (Tarsky fixed point, see [87] Theorem 2.5.1)**.** *Suppose that $\mathbb{X}$ is a non empty complete lattice, $Y(x)$ is an increasing correspondence (in the induced set ordering) from $\mathbb{X}$ to the set of the non empty complete sublattices of $\mathbb{X}$. Then the set of fixed points of Y is a nonempty complete sublattice.*

Proof of Theorem 8.6. With Lemma 8.15, $Y = (BR^1(\sigma^{-1}) \dots BR^n(\sigma^{-n}))$ is increasing in the induced set ordering on Σ. Since Σ is a non empty complete lattice (this comes from the definition of $\mathbb{B}$), Theorem 8.16 ensures that the set of fixed points is a nonempty complete sublattice. The elements of this set satisfy the definition of a Nash equilibrium. □

On the example: multinodal case Observe that the reasoning could be extended to a multinodal, non symmetric setting. One needs to use the strict increasing difference property with respect to the neighboring nodes.

8.5 Uniqueness Sufficient Condition

Proof of Theorem 8.7. Assume that the equilibria set is not a singleton. Then since it is a complete lattice, there exists a biggest and a smallest equilibria in the set. We denote by $\bar{\sigma} \in \Sigma$ and $\underline{\sigma} \in \Sigma$ the strategy profiles of those equilibria. The ratio

$$\frac{\bar{\sigma}^i[c]}{\underline{\sigma}^i[c]} \tag{8.22}$$

is bounded by b^*/b_* and therefore admits a supremum $\alpha > 1$. By compactness of $\mathbb{T}$ and continuity of the extremal equilibrium strategies (hypothesis), there exist $i^* \in \mathbb{I}$ and

$c^* \in \mathbb{T}^{i^*}$ such that

$$\frac{\bar{\sigma}^{i^*}[c^*]}{\underline{\sigma}^{i^*}[c^*]} = \alpha. \tag{8.23}$$

For any strategy profile $\sigma \in \Sigma$ we denote by $\alpha\sigma$ the strategy profile defined by

$$(\alpha\sigma)^i(c) = \alpha(\sigma^i(c)) \tag{8.24}$$

for any $i \in \mathbb{I}$ and $c \in \mathbb{T}^i$.

By (8.10) $\alpha\underline{\sigma}$ belongs to Σ. Now observe that for $i \in \mathbb{I}$, $c \in \mathbb{T}$,

$$BR^i(\alpha\underline{\sigma}^{-i})[c] = \arg\max_{b \in \mathbb{B}^i} \mathbb{E}_{c^{-i}}(b-c)K^i(b, \alpha\underline{\sigma}^{-i}[c^{-i}]) \tag{8.25}$$

$$= \arg\max_{b \in \mathbb{B}^i} \mathbb{E}_{c^{-i}}(b-c)K^i(b/\alpha, \underline{\sigma}^{-i}[c^{-i}]) \tag{8.26}$$

$$= \arg\max_{b \in \mathbb{B}^i} \mathbb{E}_{c^{-i}}(b/\alpha - c/\alpha)K^i(b/\alpha, \underline{\sigma}^{-i}[c^{-i}]) \tag{8.27}$$

$$= \alpha \arg\max_{u, \alpha u \in \mathbb{B}^i} \mathbb{E}_{c^{-i}}(u - c/\alpha)K^i(u, \underline{\sigma}^{-i}[c^{-i}]) \tag{8.28}$$

where we applied the definition of BR, the scaling relation (8.8) and a change of variable. Now combining the last computation with (8.9), we have

$$BR^i(\alpha\underline{\sigma}^{-i})[c] < \alpha\underline{\sigma}^i[c] \tag{8.29}$$

The inequality should be understood in the sense that any element of the LHS set is smaller than the RHS. Combining the definition of an equilibrium, Lemma 8.15 on the monotonicity of the best replies, and the last relation, we get

$$\bar{\sigma}^i[c] \in BR^i(\bar{\sigma}^{-i})[c] \leq BR^i(\alpha\underline{\sigma}^{-i})[c] < \alpha\underline{\sigma}^i[c] \tag{8.30}$$

which is not coherent with the definition of α. We conclude that the equilibrium is unique. □

8.6 A dynamics that converges to the smallest Nash Equilibrium

8.6.1 Proof of theorem 8.8

Proof. First we need to show that (8.11) has a solution. Let $T > 0$ and $\mathbb{D}_T$ the set of measurable functions σ from $[0, T]$ to Σ. On $\mathbb{D}_T$ we consider the operator ϕ such that, for any $\sigma \in \mathbb{D}_T$, $\phi_\sigma \in \mathbb{D}_T$ and for any $(i, c, t) \in \mathbb{I} \times \mathbb{T} \times [0, T]$

$$[\phi_\sigma(t)]^i(c) = c + \int_0^t \partial_b \Pi^i_{[\sigma(s)]^{-i}}([\sigma(s)]^i(c), c)\mathrm{d}s. \tag{8.31}$$

Observe that for T small enough, $\mathbb{D}_T$ is stable by ϕ, which is a contracting operator, and, moreover, $\mathbb{D}_T$ associated with the $||_\infty$ is a closed subset of a Banach space. Therefore we can apply Picard fixed point theorem and denote by $\tilde{\sigma}_t$ the associated fixed point. Iterating the reasoning we can extend the flow $\tilde{\sigma}_t$ as long as it stays strictly below b^*. Using the single crossing property of Π (which is conserved by integration from π), the fact that Π is convex and that the dynamics cancel at any equilibrium point, we see that $\tilde{\sigma}_t$ increases and stays below σ_*. Therefore the flow is defined for all $t \geq 0$, is bounded and increasing and therefore converges as t goes to $+\infty$ to a stationary point. Using the convexity of Π, we deduce that the stationary point is an equilibrium, therefore the flow converges to σ_*. □

8.6.2 Remarks

Observe that we could build the same kind of scheme to compute σ^*. We introduce this scheme because it showed better converging properties on our example than the Best Reply iterations. Our explanation is that the Best Reply dynamics requires a discretization of $\mathbb{B}$ and $\mathbb{T}$, while an Euler scheme of the differential equation only requires a discretization of $\mathbb{T}$ (and the time). We display in Figures 8.2 and 8.3 a numerical experiment with the Best Reply iterations and the continuous time dynamics. Even when the hypotheses of Theorem 8.8 are not satisfied, the scheme displays good converging properties.

8.7 Conclusion and possible extension

We have identified a class of Bayesian games for which we showed that there exists a unique pure Nash equilibrium to which a simple dynamics converges. Numerical experiments seem to indicate that those results could be reinvestigated with weaker assumptions.

In addition, some extensions should be considered. In particular the use of affiliated types, atoms in the distribution and general bidding functions (not only constant marginal rates).

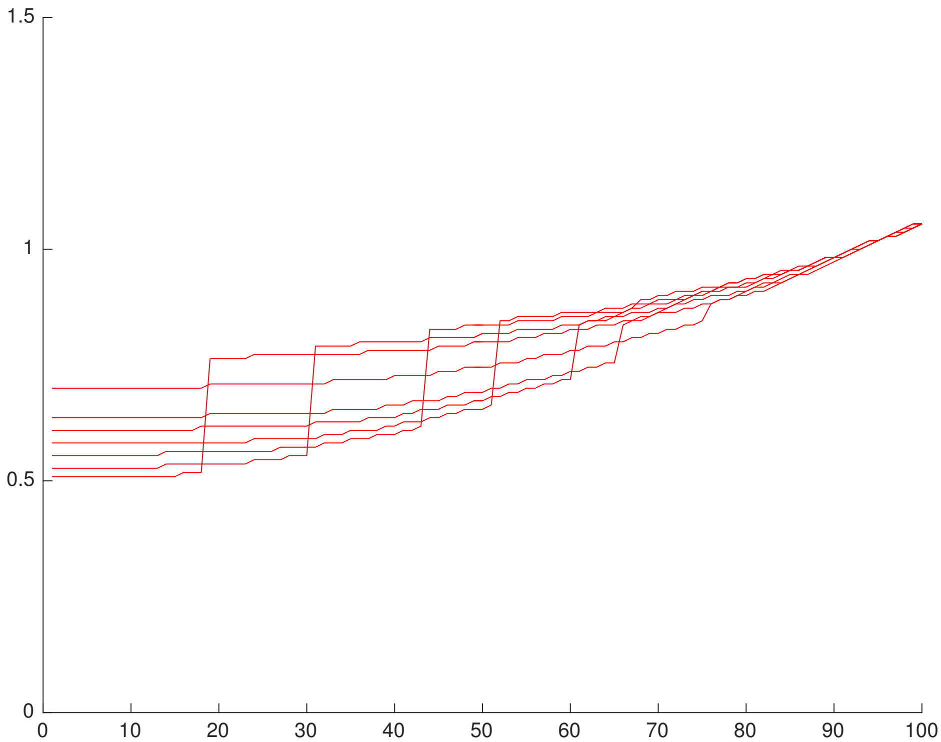


Figure 8.2 – The iterated best replies algorithm does not converge

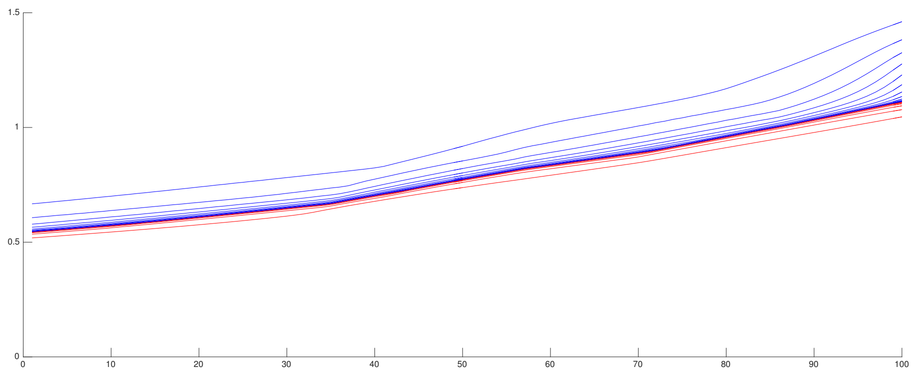


Figure 8.3 – The differential approach converges when we start above or below the equilibrium strategy profil

Appendix A

BocopHJB 1.1.0 – A collection of examples

We provide some toy problems solved with BocopHJB.

A.1 Car with obstacle

We consider a very simplified car model. The **state** (x, y) describes the coordinates of a pointlike car in $\mathbb{R}^2$. The **controls** u and θ are the velocity and the direction of the car. The **dynamics** is

$$\dot{x} = u \cos\theta$$

$$\dot{y} = u \sin\theta.$$

Our **criteria** is to reach a prescribed final position as fast as possible, therefore

$$\min \int_0^T f(x(t), y(t))dt$$

where $f(x, y)$ is 0 if we are close enough to $(x_f, y_f) = (0.2, 0.75)$, and 1 else. We define the part of the space where the car can go in order to illustrate the use of **state contraints** with BocopHJB:

$$\{(x, y) \in [0, 1]^2 : x < 0.5, y < (0.75 - 0.5x), y > (0.25 + 0.5x)\}$$

is a forbidden space. Figure A.1 shows the value function we get, with the simulated trajectory corresponding to the initial conditions $(x_0, y_0) = (0.2, 0.2)$.

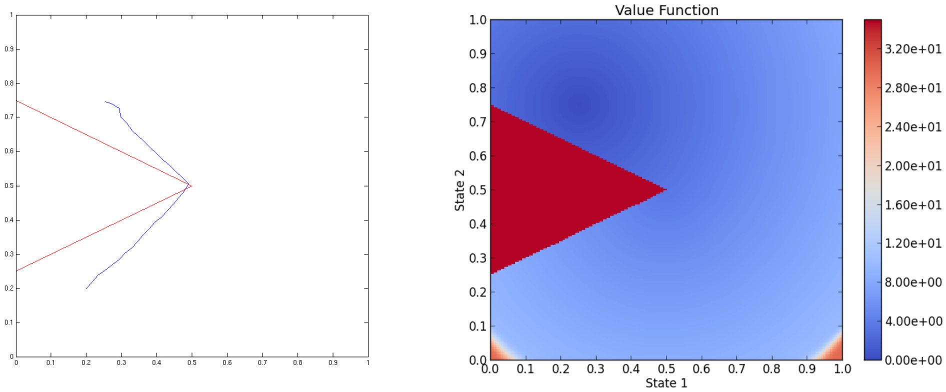


Figure A.1 – Value Function and simulated trajectory for car problem

A.2 Thermostat

We use a very simple thermostat model to illustrate the mode switching. The **state** x represents the temperature in the room. There is no **control**, but only two modes for the thermostat (on or off). The **dynamics** of the thermostat is

$$\dot{x} = +10 \quad \text{when the thermostat is on}$$

$$\dot{x} = -10 \quad \text{when the thermostat is off.}$$

We have to pay a constant cost of 1 per-unit of time when the thermostat is on. We set an additional cost of 10 per-unit of time when the temperature goes below 50. We also have a switching cost of 1 when turning the thermostat on.

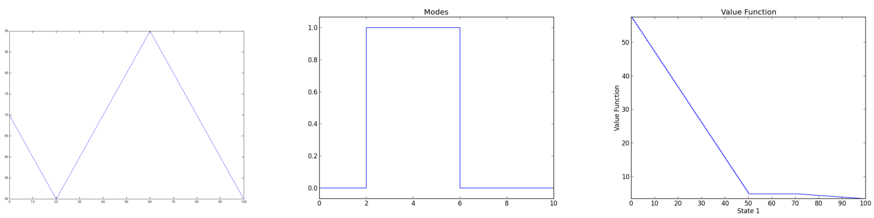


Figure A.2 – Simulated trajectory, modes and value function for the thermostat problem

A.3 Mouse in a maze

This maze illustrates the use of both discrete switches and continuous controls. A mouse tries to get out of a maze. This mouse has a "bomberman" control space. The state can be described with the variable $(x, y) \in \mathbb{R}^2$ which defines the position of the unlucky pointlike mouse. The mouse has 4 modes modeling its direction: north, east, west, south. In addition to the direction modes, the mouse has a control variable for its velocity, which is positive and upper-bounded. We consider a running cost of 10 per unit of time in the maze, and each change of direction costs 1 as a switching cost. The mouse starts at the red square while the exit of the maze is at the green square. The optimal trajectory is displayed in Fig. A.3.

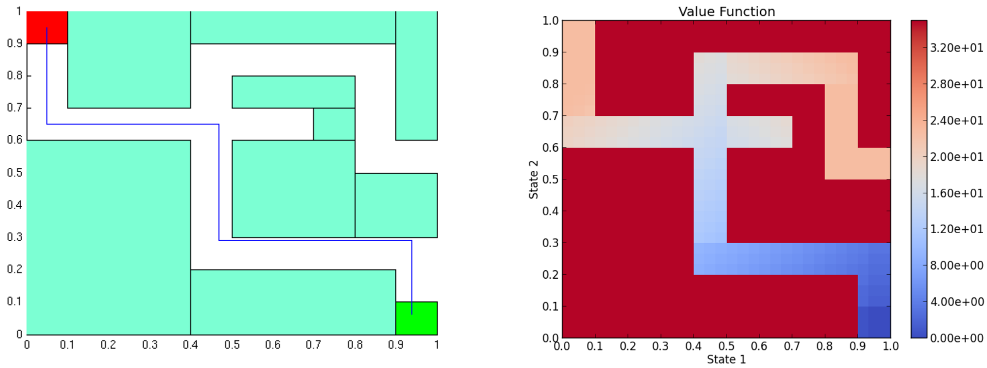


Figure A.3 – The mouse & maze trajectory, and the corresponding value function

A.4 Call option

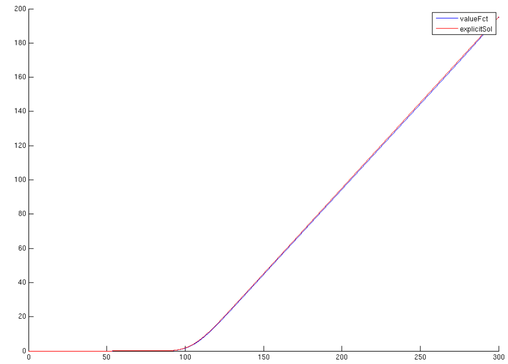
We use the Black-Scholes model as an example of a stochastic problem without control variables. We compute the price of a European call option, with S the price of a stock as the **state** variable. In Black-Scholes model, S follows the dynamics

$$dS = S(\mu dt + \sigma dW)$$

and the payoff is given by $g(S) = (S - K)^+$ where K is the strike. The interest rate is r. We solve Black-Scholes equation to compute the value of the option. We show on Fig. A.4 the results for $r = 0.01$, $\sigma = 0.02$, $K = 105$, $t = 0$, $T = 20$, $S_0 = 100$. We check that the value function is very close to the solution given by the Black-Scholes formula.

Remark: we recall that for a call option the Black-Scholes formula gives the solution

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)}$$
$$d_1 = \frac{1}{\sigma\sqrt{(T-t)}}(\ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t))$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$



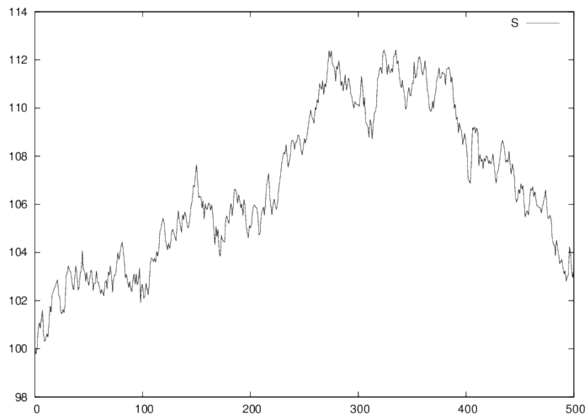


Figure A.4 – The price of a call option computed withBocopHJB and explicit solution (above), an example of simulated trajectory (below)

A.5 Portfolio allocation

As an example of a stochastic control problem, we consider the Merton portfolio allocation problem in finite horizon, for which the solution is known (see for instance [76]). The portfolio consists in a risky asset whose value S follows $dS = S(\mu dt + \sigma dW)$ and a non-risky asset whose value S_0 follows $dS_0 = S_0 r dt$. The portfolio is invested in the risky asset with proportion α, and the value of the portfolio X is the state variable with dynamics

$$dX_t = \frac{X_t \alpha_t}{S_t} dS_t + \frac{X_t(1-\alpha_t)}{S_t^0} dS_t^0 = X_t(\alpha_t \mu + (1-\alpha_t)r)dt + X_t \alpha_t \sigma dW_t.$$

We want to solve the utility maximization problem $V(x) = \sup_\alpha \mathrm{E}(\mathrm{U}(\mathrm{X}_T^{x,\alpha}))$, where U is the CRRA utility function defined by $U(x) = \frac{x^p}{p}$. The solution is given by

$$V(x) = e^{\rho T} U(x), \quad \text{with } \rho = \frac{(\mu - r)^2}{2\sigma^2} \frac{p}{1-p} + rp,$$

and the optimal control is constant, equal to $\hat{\alpha} = \frac{\mu - r}{\sigma^2(1-p)}$. The results for $p = 0.5$ and other parameters as in [76] are displayed in figure A.5.

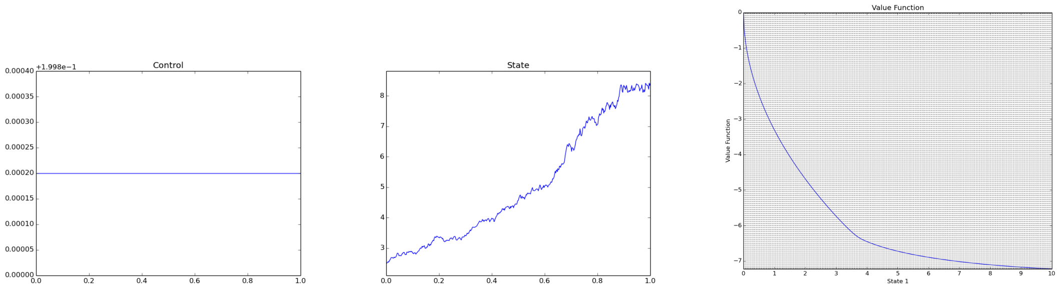


Figure A.5 – Control and state of the simulated trajectory. Value function at $t = 0$

A.6 Oscillations

We consider the optimal control problem

$$\begin{aligned} &\min \int_0^1 y^2 - u^2 \\ &\dot{y} = u \\ &u \in [-1, 1] \end{aligned}$$

The infimum is -1 (consider $u_n(t) = 1$ if $t \in [\frac{2k}{2^n}, \frac{2k+1}{2^n}]$, -1 else).

The Hamiltonian is $H = y^2 - u^2 + up$. A minimizing control u^* is either -1 or 1. The intuition is that the control has to oscillate very quickly between -1 and 1 to obtain the optimal value.

Consider that the control is randomized at any time, with probability α for $u = 1$. We can formulate the relaxed problem

$$\begin{aligned} &\min \int_0^1 y^2 - 1 \\ &\dot{y} = \mathbb{E}(u) = \alpha 1 + (1 - \alpha)(-1) = 2\alpha - 1 \\ &\alpha \in [0, 1] \end{aligned}$$

The optimal solution for the relaxed problem is given by $\alpha = 0.5$. Therefore when solving numericaly the original problem, we expect the simulated trajectory to present a sequence of very fast oscillations.

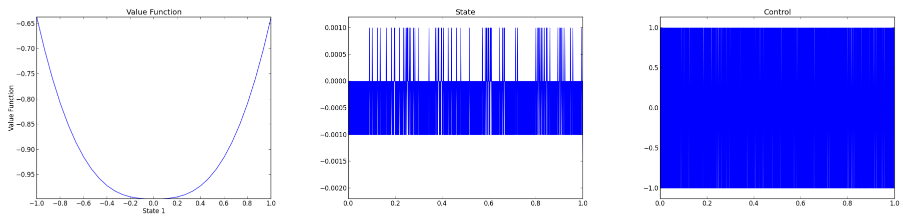


Figure A.6 – On the left the value function of the problem. The control and the state oscillate oscillations

Appendix B

Simplified microgrid model for BocopHJB

We display here the most important samples from the code we used for the battery aging study.

```
 double u = control[0];
 double P_diesel = fmax(0e0,u);
running_cost =  0.5* pow(P_diesel,2)  ;
   int k = round(time/0.25e0);
   double P_solar = SolarPower[k];
   double P_load = LoadPower[k];
   double P_bat = - P_diesel - P_solar + P_load  ;
   double P_out = fmax(0e0,P_bat);
   double U_bat = 0.25;
   double Ah_bat = 5e3;
   double alpha = Alpha[0]  ;
   double I_out = P_out  / U_bat;
   double SOC = state[0];
   double SF = (-4e0*SOC*SOC + 5e0) / 5e0;
   running_cost += alpha/ (Ah_bat) *  SF * I_out;
```

Listing B.1 – runningCostp

```
        double SOC_i = StartingPoint[0];
        double SOC = final_state[0];
        if(SOC>SOC_i){
          final_cost = (SOC-SOC_i)*1e3;
        }
        else{
          final_cost = 1e5;
        }
```

Listing B.2 – finalCost

```
    double rho_i = 0.95;
    double rho_o = 0.95;
    double age = Age[0];
    double capacity_bat = 100;
    // control
    double P_diesel = control[0];
    // data for solar power and power load
    int k = round(time/0.25e0);
    double P_solar = SolarPower[k];
    double P_load = LoadPower[k];
    // battery power split (charge / discharge)
    double P_bat = - P_diesel - P_solar + P_load;
    double P_out = fmax(0e0,P_bat);
    double P_in = -fmin(0e0,P_bat);
    // aging
    rho_i = rho_i * ( 1000-age )/1000;
    // dynamics
    state_dynamics[0] = (P_in*rho_i - P_out/rho_o ) / capacity_bat;
```

Listing B.3 – dynamics

```
# Initial and final time :
time.initial double 0
time.final double  24
# Dimensions :
state.dimension integer 1
control.dimension integer 1
constant.dimension integer 0
brownian.dimension integer 0
# Control :
discretization.control.type string uniform
combination.control.type string uniform
# Time discretization :
discretization.time integer  24
# Grid type :
grid.type string uniform
# Interpolation :
interpolation.inner string linear
interpolation.outer string user_function
# Switching mode :
switching.mode integer 1
# Simulation :
simulation.type string from_computed_sol
simulation.noise string none
simulation.starting.mode string user_function
solution.file string valueFunction
```

Listing B.4 – problemHJB.def

Appendix C

BocopHJB User Guide

THE OPTIMAL CONTROL SOLVER

BocopHJB 1.0.0 – User Guide

Frédéric Bonnans[*] Daphné Giorgi[†] Benjamin Heymann[*]
Pierre Martinon[*] Olivier Tissot[*]

August 26, 2015

[*]Inria Saclay and CMAP Ecole Polytechnique
[†]LPMA, Sorbonne University, Paris

Contents

1 BocopHJB overview

1.1 Key features

- Global optimization for both deterministic and stochastic optimal control problems.
- Handles switching between discrete modes of the system.
- Stopping time problems can be solved using switchings.
- Built-in simulation module to recompute optimal strategies.
- Supports advanced rules to define the discrete control set.
- Parallel execution with OpenMP.
- Matlab / Python scripts to read value function and simulated trajectories.

1.2 Algorithm

The original Bocop package implements a local optimization method. The optimal control problem is approximated by a finite dimensional optimization problem (NLP) using a time discretization (the direct transcription approach). The NLP problem is solved by the well known software Ipopt, using sparse exact derivatives computed by Adol-C.

The second package BocopHJB implements a global optimization method. Similarly to the Dynamic Programming approach, the optimal control problem is solved in two steps. First we solve the Hamilton-Jacobi-Bellman equation satisfied by the value fonction of the problem. Then we simulate the optimal trajectory from any chosen initial condition. The computational effort is essentially taken by the first step, whose result, the value fonction, can be stored for subsequent trajectory simulations.

1.3 Workflow

BocopHJB package contains core files and problem files. Core files implements the HJB solver and are problem independent. Each problem is defined by a set of c/c++ files and text files located in the problem folder. Solving an optimal control problem with BocopHJB involves the following steps:

1. Problem Definition
   Define the optimal control problem by completing the problem files. This files typically define the dimension, functions, and discretization (time, state and control) of the problem.

2. Build and Run
   The build step will create the `bocophjp` executable. Running the executable will, depending on the options set in `problemHJB.def`, compute the value fonction and/or simulate an optimal trajectory.

3. Visualization
   You can use provided python scripts in order to load and visualize the results of the solution and simulation files. Note that plotting the value fonction is not always available since it is a function of n variables, where n is the state dimension.

BocopHJB package includes a folder `examples/` with several sample problems to illustrate the features of the toolbox. These examples are described in more details in the document 'A collection of examples'

2 Example: the mouse & maze problem

2.1 Problem description

To test the use of both several switching possibilities and controls, we designed the following maze problem. A mouse trapped in a maze tries to get out. This mouse has a "bomberman typed" control space. The state can be described by the variable $(x, y) \in \mathbb{R}^2$ describing the position of the unlucky punctual mouse. The mouse has 4 modes modeling its direction: north, east, west, south. In addition to the direction modes, the mouse has a control variable for its velocity, which is positive and upper-bounded. We consider a running cost of 10 per unit of time in the maze, and each change of direction costs 1 as a switching cost. The mouse starts at the red square while the exit of the maze is at the green square. The optimal trajectory is shown on Fig. 1.

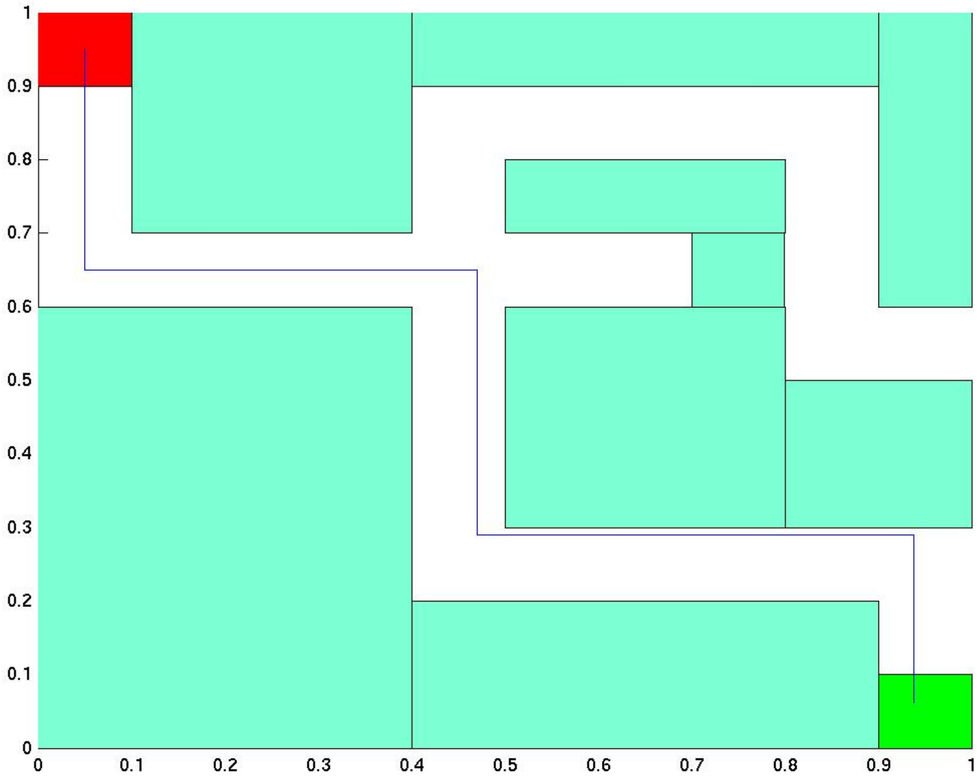


Figure 1: The Maze and the mouse trajectory according to BocopHJB

You can run this test and display the results with the following commands in terminal. Locally from the problem folder (`examples/maze/`):

```
> ./build
> ./bocophjb
```

Or from the root of the package:

```
> sh bocop build examples/maze
> sh bocop run examples/maze
```

2.2 Files for the mouse & maze problem

2.2.1 Definition files

`problemHJB.def`, `stateDisc/state.grid`, `controlDisc/control.grid`, `controlDisc/control.combination`.

```
# This file defines all dimensions and parameters
# values for your problem :

# Initial and final time :
time.initial double 0
time.final double 3

# Dimensions :
state.dimension integer 2
control.dimension integer 1
constant.dimension integer 0
brownian.dimension integer 0

# Control :
discretization.control.type string uniform
combination.control.type string uniform

# Time discretization :
discretization.time integer 50

# Grid type :
grid.type string uniform

# Interpolation :
# Inner : linear ; other
# Outer : final value ; projection ; user function
interpolation.inner string linear
interpolation.outer string user_function

# Switching mode :
switching.mode integer 4

# Names :
state.0 string x1
state.1 string x2
control.0 string u

# Simulation :
simulation.type string from_computed_sol
simulation.noise string none

solution.file string valueFunction.sol
```

```
# Discretization of the state :
discretization.state.0 integer 30
discretization.state.1 integer 30

# Minimum of the state grid :
minimum.state.0 double 0
minimum.state.1 double 0

# Maximum of the state grid :
maximum.state.0 double 1
maximum.state.1 double 1
```

```
# Discretization :
discretization.control.0 integer 11

# Minimum of the control grid :
minimum.control.0 double 0

# Maximum of the control grid :
maximum.control.0 double 1
```

2.2.2 Source files

`dynamicsHJB.cpp`

```
/**
  * Drift function which describes the deterministic part of the dynamics.
  */
#include "header_drift"
{
    double u1 = control[0];

    switch(mode)
    {
    case 0 : // UP
        state_dynamics[0] = 0.0;
        state_dynamics[1] = u1;
        break;
    case 1 : // DOWN
        state_dynamics[0] =  0.0;
        state_dynamics[1] = -u1;
        break;
    case 2 : // LEFT
        state_dynamics[0] = -u1;
        state_dynamics[1] =  0.0;
        break;
    case 3 : // RIGHT
        state_dynamics[0] = u1;
        state_dynamics[1] = 0.0;
        break;
    }

}

/**
  * Volatility function which describes the stochastic part of the dynamics.
  */
#include "header_volatility"
```

```
{
// This function is unused since the problem is deterministic.
```

costFunctions.cpp

```
/**
  * Running cost for the computation of the criterion.
  */
#include "header_runningCost"
{
    double x1 = state[0];
    double x2 = state[1];

    if ( (x1 > 0.9) && (x2 < 0.1) )
        running_cost = 0;
    else
        running_cost = 10;
}

/**
  * Final cost for the computation of the criterion.
  */
#include "header_finalCost"
{
    final_cost = 0;
}

/**
  * Switching cost for the computation of the criterion.
  */
#include "header_switchingCost"
{

    if (current_mode == next_mode)
        switching_cost = 0;
    else
        switching_cost = 1;
}
```

constraints.cpp

```
/**
  *  User function used to check if a state is admissible or not.
  */
#include "header_checkAdmissibleState"
{

// We use state contraints to describe  the maze (position of walls).
    double x = state[0];
    double y = state[1];

    if( (x>1) ||  (x<0) || (y<0) || (y>1) ){return false;}

    if( (x<0.4) && (y<0.6) ){return false;}
    if( (x>=0.1) && (x<0.4) && (y>=0.7) ){return false;}
    if( (x>=0.4) && (x<0.9) && (y<0.2) ){return false;}

    if( (x>=0.4) && (x<0.9) && (y>=0.9) ){return false;}
    if( (x>=0.9) && (y>=0.6) ){return false;}


    if( (x>=0.5) && (x<0.8) && (y>=0.7) && (y<0.8) ){return false;}
    if( (x>=0.5) && (x<0.8) && (y>=0.3) && (y<0.6) ){return false;}
    if( (x>=0.7) && (x<0.8) && (y>=0.6) && (y<0.7) ){return false;}

    if( (x>=0.8) && (y>=0.3) && (y<0.5) ){return false;}

    return true;
}

/**
  *  User function used to check if a combination of controls and a state is admissible or not.
  */
#include "header_checkAdmissibleControlState"
{
    return true;
}
```

simulation.cpp

```
/**
  *\fn void simulationStartingPoint(std::vector<double>& starting_point)
  * User function to define the starting point of the simulation.
  */
#include "header_simulationStartingPoint"
{
  starting_point[0] = 0.05;
  starting_point[1] = 0.95;
```

```
}

/**
  *\fn void simulationStartingMode(int& starting_mode)
  * User function to define the starting mode of the simulation.
  */
#include "header_simulationStartingMode"
{
  starting_mode = 0;
}
```

optionalFunctions.cpp

```
/**
  * User function to compute the value of the value function for the points outside the grid.
  */
#include "header_userOutOfGridValueFunction"
{
// we return a huge value to prevent exit from the grid
    result = 10000;
}

/**
  *  User function used to define the discretized controls.
  */
#include "header_userControlDiscretization"
{
//unused function for this example (see control.discretization in problemHJB.def)
    return 0;
}

/**
  *  User function used to compute the combinations of controls.
  *  Each line of the resulting matrix is a combination of controls (u_0,..., u_p).
  */
#include "header_userControlCombination"
{
//unused function for this example (see control.discretization in problemHJB.def)

    return vector< vector<double> >();
}

/**
  *  User function used to compute the combinations of controls when it depends of state.
  *  Each line of the resulting matrix is a combination of controls (u_0,..., u_p).
  */
#include "header_userControlCombinationStateDependent"
{
//unused function for this example (see control.discretization in problemHJB.def)

    return vector< vector<double> >();
}
```

3 Algorithm description

3.1 Stochastic optimal control problem

Let y_t be a stochastic process described by

$$\begin{cases} dy_t = f(t, u_t, y_t)dt + \sigma(t, u_t, y_t)dW_t \\ y_0 = x \end{cases} \tag{1}$$

where the *control* $u_t \in U$ and $t \in [0, \infty[$, W_t is a standard Brownian motion and the *drift* f and the *volatility* σ are Lipschitz and bounded.

We define $\mathcal{U}$ the set of mappings with value in U adapted to the filtration generated by the Brownian motion (which means that we can take $u(t)$ as a function of the past history of the Brownian). We want to solve the stochastic optimal control problem

$$\min_{u \in \mathcal{U}} \mathbb{E} \left(\int_{t_0}^{T} \ell(t, u_s, y_s)ds + \phi(y_T) \right) \tag{2}$$

where ℓ is the *running cost* and ϕ the *final cost.*

Remark: Our framework includes additional state and control constraints of the form $g(t, u(t), y(t)) \leq 0$. *It also handles switchings between several modes, which allows in particular to solve stopping time problems, on/off state of plants, etc.*

3.2 Dynamic Programming Principle

We define the value function $V(x, t)$ such that

$$V(x, t) := \min_{u \in \mathcal{U}} \mathbb{E} \left(\int_{t}^{T} \ell(t, u_s, y_s)ds + \phi(y_T) \,\middle|\, y_t = x \right)$$

and

$$V(x, T) = \phi(x)$$

Let us take $\tau \in (t_0, T)$. We can write

$$V(y_0, t_0) = \min_{u \in \mathcal{U}} \mathbb{E}_{t_0} \left(\int_{t_0}^{\tau} \ell(t, u_s, y_s)ds + \int_{\tau}^{T} \ell(t, u_s, y_s)ds + \phi(y_T) \right)$$

which leads to the dynamic programming equation

$$V(y_0, t_0) = \min_{u \in \mathcal{U}} \mathbb{E}_{t_0} \left(\int_{t_0}^{\tau} \ell(t, u_s, y_s)ds + V(y_\tau, \tau) \right) \tag{3}$$

We can discretize on time the stochastic process (using for instance an Euler scheme), so that we have y^{k+1} as a function of y^k, σ^k, u^k. Let $t_k = h_0 k$ with $t_N = T$. The discretized problem is

$$\min_{u_k \in \mathcal{U}} \mathbb{E} \left(h_0 \sum_{k=0}^{N-1} \ell(t_k, u^k, y^k) + \phi(y^N) \right)$$

where we set $y^k = y(t_k)$ and $u^k = u(t_k)$. The value function is defined as

$$V^k(x) := \min_{u \in \mathcal{U}} \mathbb{E} \left(h_0 \sum_{j=0}^{N-1} \ell(t_i, u^j, y^j) + \phi(y^N) \,\middle|\, y^k = x \right)$$

which leads to

$$V^k(x) := \min_{u \in U} \mathbb{E}_x \left(h_0 l(t_k, u, x) + V^{k+1}(y^{k+1}) \right) \tag{4}$$

with final condition

$$V^N(x) = \phi(x) \tag{5}$$

We can extend this reasoning to cases where the dynamics and the cost functions depend of a mode : a diesel engine for example which can be turned off or on. If we denote M the number of modes, with a subscript i (or j) the functions corresponding to the mode i and c_{ij} the switching cost from mode i to mode j (assuming that $c_{ii} = 0$), this leads to

$$V_i^k(x) = \min_{j \in \{0,\dots,M\}} \left(c_{ij} + \min_{u \in U} \left\{ h_0 \ell_j(t_k, u, x) + \mathbb{E}_x \left[V_j^{k+1}\left(y^{k+1}\right) \right] \right\} \right) \tag{6}$$

The algorithm used to compute the Value function at t_k is the following

Algorithm 1 Compute V^k

Require: $0 \le k \le N$
  for $x \in$ Grid **do**
    if $k = N$ **then**
      $V^N(x) = \phi(x)$
    else
      for $i \in \{0, \dots, M\}$ **do**
        $\widetilde{V}_i^k(x) = \min_{u \in U} \left(h_0 \ell_j(t_k, u, x) + \mathbb{E}_x \left[V_j^{k+1}\left(y^{k+1}\right) \right] \right)$
      end for
      for $i \in \{0, \dots, M\}$ **do**
        $V_i^k(x) = \min_{j \in \{0,\dots,M\}} \left(c_{ij} + \widetilde{V}_j^k(x) \right)$
      end for
    end if
  end for

This algorithm is independent of the way of calculating $\mathbb{E}_x \left[V_j^{k+1}\left(y^{k+1}\right) \right]$. A classical method is to use an interpolation on the grid of V^{k+1} and an Euler scheme for the dynamics: this is the semi-Lagrangian method, as used in BocopHJB.

3.3 Semi Lagrangian scheme

3.3.1 Time discretization

Remark: in the following we drop the argument t^k in functions f, l for clarity.

In the deterministic case, we naturally discretize the dynamics:

$$y^{k+1} = y^k + h_0 f(u^k, y^k) \tag{7}$$

In the stochastic case, remembering that a Brownian motion has independent increments following a Normal law, $W(t_{k+1}) - W(t_k) \sim \sqrt{h_0}\mathcal{N}(0,1)$, we obtain

$$y^{k+1} = y^k + h_0 f(u^k, y^k) + \sqrt{h_0}\sigma(u^k, y^k)\mathcal{N}(0,1) \tag{8}$$

According to [3], $\mathcal{N}(0,1)$ can be replaced by any law with the same first two moments. We use a binary choice and obtain

$$\begin{aligned} y^{k+1} &\simeq y^k + h_0 f(u^k, y^k) + \alpha\sqrt{h_0}e\sigma_{\mathcal{X}}(u^k, y^k) \\ \mathbb{P}(e=1) &= \mathbb{P}(e=-1) = \frac{1}{2} \end{aligned} \tag{9}$$

where $\mathcal{X}$ follows an uniform distribution on $\{1,\dots,q\}$ and we have to choose α such that the expected value and the variance of this approximated process correspond to the ones of the original process in (8). Since the normal distribution and the random variable e are centered, and e and $\mathcal{X}$ are independent, the expected value is the same for any α. The variance in (8) is $h_0\sigma\sigma^T$. The variance in (9) writes

$$\mathbb{E}\left(\alpha\sqrt{h_0}e\sigma_{\mathcal{X}}(\alpha\sqrt{h_0}e\sigma_{\mathcal{X}})^T\right) = \alpha^2 h_0 \mathbb{E}\left(e^2\sigma_{\mathcal{X}}\sigma_{\mathcal{X}}^T\right) = \alpha^2 h_0 \frac{1}{q}\sum_{s=1}^{q}\sigma_s\sigma_s^T = \frac{\alpha^2}{q}h_0\sigma\sigma^T$$

therefore we have $\alpha = \sqrt{q}$. Plugging (9) in (3) we obtain

$$V^k(x) = \min_{u\in U}\left(h_0\ell(u,x) + \frac{1}{2q}\sum_{s=1}^{q}V^{k+1}\left(x + h_0 f(u,x) \pm \sqrt{qh_0}\sigma_s(u,x)\right)\right). \tag{10}$$

3.3.2 Space discretization

We know the value of V at the points of the grid, and we want to interpolate at the point y. We choose the coefficients $\alpha_i \in [0,1]$ such that $y_j = (1-\alpha_j)x_{i_j} + \alpha_j x_{i_j+1}$. We interpolate the value function at the point y as follows (see [2]):

$$V^{k+1}(y) = \sum_{(k_1,\dots,k_n)\in\{0,1\}^n}\left[\prod_{j=1}^{n}(1-\alpha_j)^{1-k_j}\alpha_j^{k_j}\right]V^{k+1}(x_{i_1+k_1},\dots,x_{i_n+k_n})$$

where the sum is made on the 2^n elements of $\{0,1\}^n$.

When a point doesn't belong to the grid we cannot interpolate the value function at this point. A typical choice is to take the value of the nearest point of the grid. Depending on the problem, another sensible choice can be to take the final cost.

3.3.3 Control discretization

The minimizer of (10) is approximated by discretizing the control set U.

3.3.4 Simulation

BocopHJB includes a built-in module to simulate the optimal strategies provided by the dynamic programming algorithm. At each time step, the optimal control is taken as the minimizer of (4) over the discrete control set.

4 Description of problem files

In BocopHJB a problem is defined by the following files:

- a set of **(C/C++)** files:
  - *constraints.cpp* for the constraints of the problem (state and/or control-state)
  - *costFunctions.cpp* for the running, final and switching cost functions
  - *dynamicsHJB.cpp* for the drift and volatility
  - *simulation.cpp* for the initial conditions of the simulated trajectory
  - *optionalFunctions.cpp* for several optional functions see 4.5

- a set of text files:
  - *problemHJB.def* for general definition and settings
  - *stateDisc/* folder for state discretization
  - *controlDisc/* folder for control discretization

4.1 Definition file: problemHJB.def

This file defines the dimensions and names for the variables, as well as several general parameters. Note that the ordering of the lines in this file does not matter. Blank lines can be used for more clarity, as well as comments beginning by #. We recommend renaming every variable and control, however this is not mandatory. The line format is the following: `keyword type value`, where the keywords are listed below and the type can be integer, double or string.

- Initial and final time
  - `time.initial`: initial time t_0
  - `time.final`: final time t_f

- Dimensions
  - `state.dimension`: dimension of state variables y
  - `control.dimension`: dimension of control variables u
  - `constants.dimension`: number of numerical constants
  - `brownian.dimension`: dimension of brownian motion W

- Control discretization
  - `discretization.control.type`: discretization for each component of the control, can be "uniform" (automatic), "user_function" (see 4.5.2), or "user_file". The values for the i-th control component are in the files `controlDisc/control.i.disc` and must be filled manually if option is set to "user_file".
  - `combination.control.type`: how to build the discretized control set. It can be "uniform" (automatic), "user_function"(see 4.5.2), or "user_file". The control set is written in the file `controlDisc/control.combination`, one element per row. As above, the file must be filled manually if option is set to "user_file", in which case "discretization.control.type" is ignored.

- Time discretization
  - `discretization.time`: number of time steps

- Grid type
  - `grid.type`: type of state grid, for now the only available option is "uniform".

- Interpolation
  - `interpolation.inner`: type of interpolation for the points inside the grid, for now the only available option is "linear_interpolation".
  - `interpolation.outer`: type of interpolation for the points outside the grid, can be "final_value" for the final value, "projection" for the projection on the nearest point of the grid, or "user_function" (see 4.5.3) for a specific function coded by the user.

- Switching mode
  - `switching.modes`: number of modes among which the system can switch. Set to 1 if there are no switchings.

- Simulation
  - `simulation.directory`: the name of an existing directory inside the problem directory where the simulation results will be saved. The simulated trajectory consist in the files *simulatedTrajectory.[times,states,controls, modes]* that contain the values for $(t, x(t), u(t))$ and the mode.

  - `simulation.type`: can be "none" (only compute the value function, no trajectory simulation), "from_computed_sol" (first compute the value function, then simulate the optimal trajectory from the given initial conditions), or "from_sol_file" (read a previously computed value function file then simulate the optimal trajectory).

  - `simulation.noise`: type of noise (i.e. realization of the Brownian for the simulation), can be "none", "gaussian", or "user_function" (see 4.5.5). This parameter has no effect for deterministic problem with `brownian.dimension` set to 0.

  - `simulation.starting.mode`: set the initial mode for the simulation; "auto" picks the initial mode i_0 giving the lowest value of $V(t_0, x_0, i_0)$, "user_function" lets the user set explicitly the initial mode i_0 in simulation.cpp (see 4.4).

- Names
  - `state.i`: name of component i of y
  - `control.i`: name of component i of u

- Constants
  - `constant.i`: name and value of i^{th} constant, the name replaces the type for constants (ex: `constant.0 c0 1.0`)

- Solution file
  - `solution.file`: name of the solution file (default "valueFunction.sol")

- Output frequency
  - `timestep.output.frequency`: frequency of the displayed output (in the terminal), can be 0 for no output at all, 1 to output every time step, or n (with n an integer less than the number of time step) to output only the time steps which are multiple of n.

4.2 State discretization file: folder stateDisc/

This file *state.grid* gives, for each component of the state, the lower and upper bounds and the number of discretization steps (uniformly spread). For instance, 10 steps in $[0, 1]$ give the discretized set $\{0, 0.1, \ldots, 1\}$.
- `discretization.state.i`: number of discretization steps for component i
- `minimum.state.i`: lower bound for component i
- `maximum.state.i`: upper bound for component i

4.3 Control discretization file: folder controlDisc/

The files to be completed depend on the options "discretization.control.type" and "control.combination.type".

- **Discretized control set:** if "control.combination.type" is set to
  - "uniform", the control set will be built automatically by taking the values from each control component (see below).
  - "user_function" or "user_function_statedependent": the control set will be built by the corresponding user function (see 4.5.2).
  - "user_file": complete the file *control.combination*, each row containing a m-tuple where m is the dimension of the control space.

- **IF CONTROL.COMBINATION.TYPE=UNIFORM.**
  Individual control components: if "discretization.control.type" is set to
  - "uniform": complete the file *control.grid* with a syntax similar to *state.grid*. Individual files *control.i.disc* will be written automatically.
  - "user_function": the control component will be discretized by the corresponding user function (see 4.5.2).
  - "user_file": complete the individual files *control.i.disc* for each component of the control. Each file contains the set of discretized values for the corresponding component.

Example: Assume we have a problem with a two-dimensional control u with $u_0 \in \{0, 1\}$ and $u_1 \in \{0, 1\}$. Setting control.combination.type to "uniform" gives the discretized control set $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$. If we want to impose the constraint $u_0 \geq u_1$, we can define directly the control set with *control.combination.type* set to *user_file*, and write the file `controlDisc/control.combination` as follows

```
0 0
1 0
1 1
```

4.4 Basic Functions for the optimal control problem

The user has to write the functions which define the problem: the drift and the (optional) volatility to describe the dynamics, the running cost, the final cost and the (optional) switching cost to describe the criterion to optimize; if there are constraints, the functions to check the admissibility of the states and the controls; and some other optional functions, if the user wants to give its own functions to discretize the single controls, to make the control combinations, or to interpolate inside and/or outside the grid.

The dynamics functions f and σ are in dynamicsHJB.cpp:

```
// Drift function which describes the deterministic part of the dynamics.
void drift(const double& initial_time,
           const double& final_time,
           const double& time,
           const vector<double>& control,
           const vector<double>& state,
           const int mode,
           const int dim_constant,
           const double* constants,
           vector<double>&  state_dynamics)
```

```
// Volatility function which describes the stochastic part of the dynamics.
void volatility(const double& initial_time,
                const double& final_time,
                const double& time,
                const vector<double>& control,
                const vector<double>& state,
                const int mode,
                const int dim_constant,
                const double* constants,
                vector<double>&  volatility_dynamics)
```

Cost functions are in costFunctions.cpp:

```
// Running cost for the computation of the criterion.
void runningCost(const double& initial_time,
                 const double& final_time,
                 const double& time,
                 const vector<double>& control,
                 const vector<double>& state,
                 const int mode,
                 const int dim_constant,
                 const double* constants,
                 double& running_cost)
```

```
// Final cost for the computation of the criterion.
void finalCost(const double& initial_time,
               const double& final_time,
               const vector<double>& state,
               const int mode,
               const int dim_constant,
               const double* constants,
               double& final_cost)
```

For the simulation step, one has to set the initial state and mode in simulation.cpp. Modes are numbered from 0 to NbModes-1.

```
// Starting point definition.
void simulationStartingPoint(vector<double>& starting_point)
```

```
// Starting mode definition.
void simulationStartingMode(int& starting_mode)
```

4.5 More advanced features

In this part we describe some optional more advanced functions.

4.5.1 State and/or control constraints

State and control admissibility functions are in constraints.cpp:

```
// User function used to check if a combination of controls is admissible or not.
bool checkAdmissibleControl(const vector<double> control,
                            const int dim_constant,
                            const double* constants)
```

```
// User function used to check if a state is admissible or not.
bool checkAdmissibleState(const double initial_time,
                          const double final_time,
                          const double time,
                          const vector<double> state,
                          const int mode,
                          const int dim_constant,
                          const double* constants)
```

```
// User function used to check if a combination of controls and a state is admissible or not.
bool checkAdmissibleControlState(const double initial_time,
                                 const double final_time,
                                 const double time,
                                 const vector<double> control,
                                 const vector<double> state,
                                 const int mode,
                                 const int dim_constant,
                                 const double* constants)
```

4.5.2 Non uniform control discretization

Control discretization functions are in optionalFunctions.cpp. :

This function allows to define explicitly the discretized values taken by each component of the control.

```
// User function used to define the discretized controls.
// The user has to fill the values of m_discretizedControl[i][j] with i=0,...,m_dimControl
// and j=0,...,m_discretizedControl[i].size()
int userControlDiscretization()
```

This function allows to define explicitly the elements of the discret control set. Each element is an m-tuple, where m is the dimension of the control space. It can be used in particular to enforce some constraints on the control.

```
// User function to compute the combinations of controls.
// Each line of the resulting matrix is a combination of controls (u_0,..., u_p)
vector< vector<double> > userControlCombination(const int dim_constant,
                                                const double* constants)
```

The next function is similar but also take into account the state variables.

```
// User function to compute the combinations of controls when it depends on state.
// Each line of the resulting matrix is a combination of controls (u_0,..., u_p)
vector< vector<double> > userControlCombinationStateDependent(const double initial_time,
                                                              const double final_time,
                                                              const double time,
                                                              const vector<double> state,
                                                              const int mode,
                                                              const int dim_constant,
                                                              const double* constants)
```

4.5.3 Out of grid evaluation

Interpolation of the value function when it is out of the grid is in optionalFunctions.cpp:

```
// User function to compute the value of the value function for the points outside the grid.
void userOutOfGridValueFunction(const double initial_time,
                        const double final_time,
                        const double time,
                        const vector<double>& state,
                        const int dim_constant,
                        const double* constants,
                        double& result)
```

4.5.4 Switching modes

If the system has several modes (set in problemHJB.def) we must define the cost of switching from one mode to another. Modes are numbered from 0 to NbModes-1.

```
// Switching cost for the computation of the criterion.
void switchingCost(const int initial_mode,
                   const int final_mode,
                   const int dim_constant,
                   const double* constants,
                   double& switching_cost)
```

4.5.5 Brownian realization for the simulation

If simulation.noise is set to user_function, user_noise() in optionalFunctions.cpp defines the Brownian realization used in the simulation.

```
// User function to compute the noise for the simulation.
std::vector<double> user_noise()
```

References

[1] Kristian Debrabant and Espen Jakobsen. Semi-lagrangian schemes for linear and fully non-linear diffusion equations. *Mathematics of Computation*, 82(283):1433–1462, 2013.

[2] Maurizio Falcone and Roberto Ferretti. *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*. SIAM, 2013.

[3] Harold Kushner and Paul G Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2013.

[4] Huyên Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media, 2009.

A Install notes (INSTALL file)

```
*************************
BOCOP HJB INSTALL NOTES
*************************

----------------------------------------
LINUX

In the following, <BOCOPHJB> is the directory in which you have extracted
the package. Please make sure that there are no blanks or spaces in the
path name to this folder.


A. PREREQUISITES

BocopHJB requires the compiler g++ and CMake.
Please install them if necessary (using yum, apt-get or the system tools).


B. HOW TO LAUNCH BOCOPHJB

First we recommend that you compile and run a test case. To do so you can call
the following commands from <BOCOPHJB>:

> ./bocop build examples/maze
> ./bocop run examples/maze

To define a new problem you can call the following command:

> ./bocop create_problem PROBLEM_NAME

Once you have completed the input files located in <BOCOPHJB>/problems/PROBLEM_NAME
as described in the documentation. You have to compile (build) and run BocopHJB:

> ./bocop build problems/PROBLEM_NAME
> ./bocop run problems/PROBLEM_NAME

If you want to visualize the simulation results you can call the following command:

> ./bocop visualize -s -d problems/PROBLEM_NAME

To visualize the value function you can call the following commands:

> ./bocop visualize -v -d problems/PROBLEM_NAME -m MODE_VALUE -t TIME_VALUE
```

```
Example:
> ./bocop visualize -v -d <BOCOPHJB>/examples/maze -m 0 -t 0

NB: you can use the -h option to print an help message.


----------------------------------------
MAC OS

In the following, <BOCOPHJB> is the directory in which you have extracted
the package. Please make sure that there are no blanks or spaces in the
path name to this folder.


A. PREREQUISITES

BocopHJB requires Xcode and CMake.
Please install them if necessary according to the following guideline.

A.1 XCODE

Download and install Xcode from the appstore. Please note that you have to accept
Xcode license in order to use the C++ compiler.

A.2 CMAKE

1) Get cmake from internet, put it in /Applications

2) Check that the file 'cmake', 'ccmake' are in the directory
/Applications/CMake.app/Contents/bin/

3) Open a terminal and create symbolic links to /usr/bin as follows:
     sudo ln -s /Applications/CMake.app/Contents/bin/ccmake /usr/bin/ccmake
     sudo ln -s /Applications/CMake.app/Contents/bin/ccmake /usr/bin/cmake

4) Check the result by typing in terminal
     which cmake
     which ccmake

The answers should be
/usr/bin/cmake
/usr/bin/ccmake


B. HOW TO LAUNCH BOCOPHJB
```

First we recommend that you compile and run a test case. To do so you can call the following commands from <BOCOPHJB>:

```
> ./bocop build examples/maze
> ./bocop run examples/maze
```

To define a new problem you can call the following command:

```
> ./bocop create_problem PROBLEM_NAME
```

Once you have completed the input files located in <BOCOPHJB>/problems/PROBLEM_NAME as described in the documentation. You have to compile (build) and run BocopHJB:

```
> ./bocop build problems/PROBLEM_NAME
> ./bocop run problems/PROBLEM_NAME
```

If you want to visualize the simulation results you can call the following command:

```
> ./bocop visualize -s -d problems/PROBLEM_NAME
```

To visualize the value function you can call the following commands:

```
> ./bocop visualize -v -d problems/PROBLEM_NAME -m MODE_VALUE -t TIME_VALUE
```

Example:

```
> ./visualize_solution -d <BOCOPHJB>/examples/maze -m 0 -t 0
```

NB: you can use the -h option to print an help message.

WINDOWS

In the following, <BOCOPHJB> is the directory in which you have extracted the package. Please make sure that there are no blanks or spaces in the path name to this folder.

WARNING : BocopHJB must be installed in a directory without any blanks or spaces, in particular not in Program Files !

A. PREREQUISITES

BocopHJB requires MinGW and CMake to run on Windows.

A.1 MINGW

Due to some incompatibilities with the latest MinGW version, we
recommend that you use the provided full MinGW archive, available on the
Download page of bocop.org.
Simply extract the archive to a location without spaces in its
name (for instance C:\, but NOT C:\Program Files\).

In the following, <MINGW> is the installation target directory
(for example C:\MinGW which is the preferred one).

* Change the Path environment variable, as explained here :

- Right-click on your "My Computer" icon and select "Properties".
- Click on the "Advanced" tab, then on the "Environment Variables" button.
- Click on the PATH entry and edit it.
- Scroll to the BEGINNING of the string and add the directories for your MinGW:

```
<MinGW>\msys\1.0\bin;<MinGW>\bin;
```

Note: we recommend to put the two directories for MinGW at the beginning of the
PATH to avoid the confusion with other versions of files such as sed.exe or
libtools that may be present in your system folders. Such files can be installed
by other applications, and may not be compatible with the building process in
Bocop.

A.2 CMAKE

The building process requires CMake. You can download the installer here:
http://www.cmake.org/cmake/resources/software.html

During the installation process choose the option to add the CMake path
in the Path environment variable.

When this is done please reboot your computer to update the Path
environment variable.
IMPORTANT:
CMake under Windows assumes building with Visual Studio by default.
Since we currently use MinGW instead, we have to add the option
-G "MSYS Makefiles" as stated below.

A.3 PYTHON

Visualization process need Python 2.7 installed.

B. HOW TO LAUNCH BOCOPHJB

First we recommend that you compile and run a test case. To do so you can call the following commands from <BOCOPHJB>:

```
> sh bocop build examples/maze
> sh bocop run examples/maze
```

/!\ Please note that the check need python installed to be passed /!\

To define a new problem you can call the following command:

```
> sh bocop create_problem PROBLEM_NAME
```

Once you have completed the input files located in <BOCOPHJB>/problems/PROBLEM_NAME as described in the documentation. You have to compile (build) and run BocopHJB:

```
> sh bocop build problems/PROBLEM_NAME
> sh bocop run problems/PROBLEM_NAME
```

If you want to visualize the simulation results you can call the following command:

```
> sh bocop visualize -s -d problems/PROBLEM_NAME
```

To visualize the value function you can call the following commands:

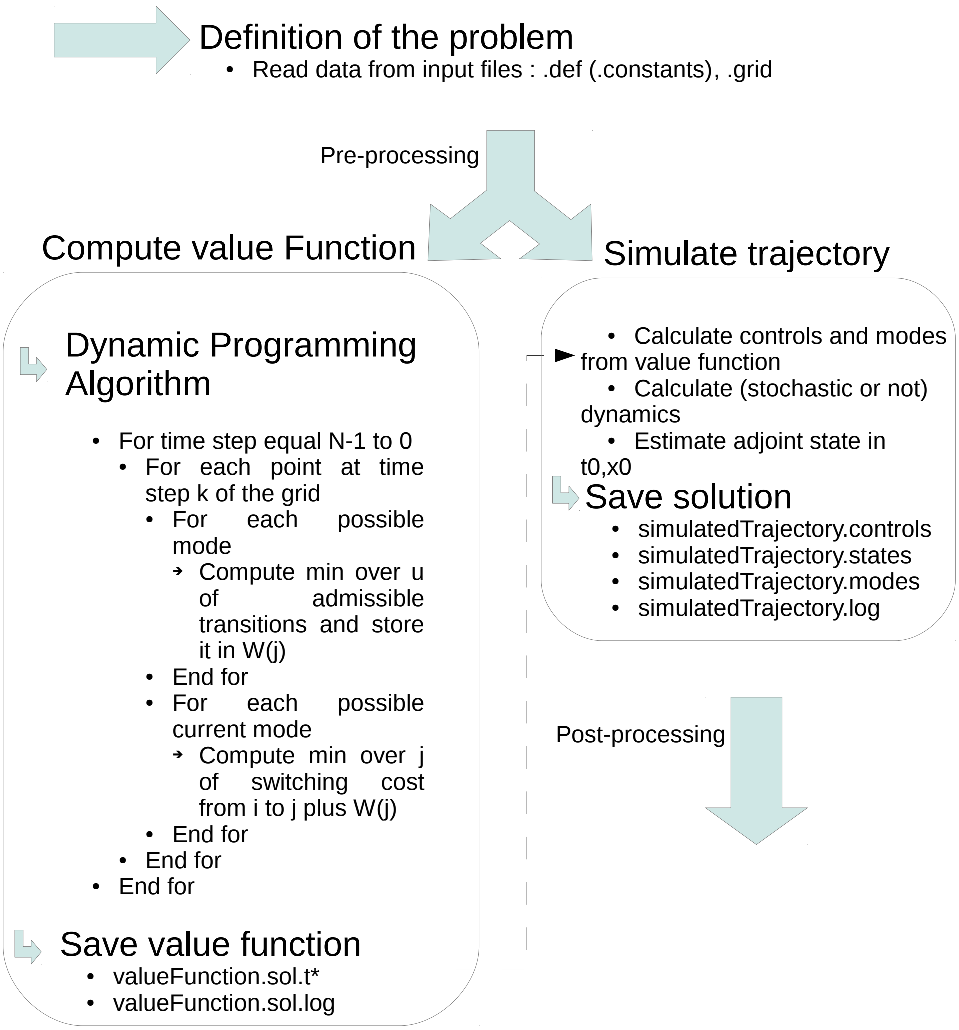
```
> sh bocop visualize -v -d problems/PROBLEM_NAME -m MODE_VALUE -t TIME_VALUE
```

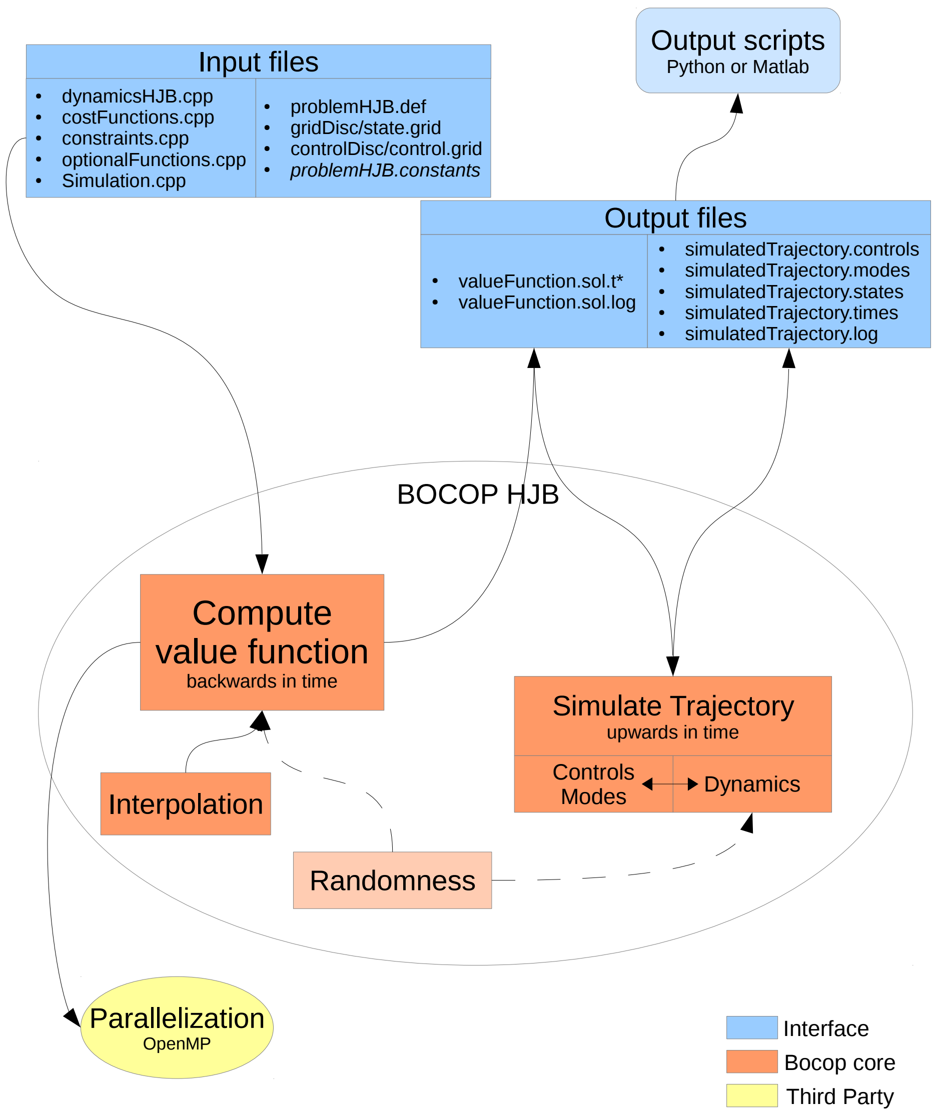
Example:

```
> sh bocop visualize -v -d <BOCOPHJB>/examples/maze -m 0 -t 0
```

NB: you can use the -h option to print an help message.

B Code structure




Output scripts
Python or Matlab
Input files
dynamicsHJB.cpp
costFunctions.cpp
constraints.cpp
optionalFunctions.cpp
Simulation.cpp
problemHJB.def
gridDisc/state.grid
controlDisc/control.grid
problemHJB.constants
Output files
valueFunction.sol.t*
valueFunction.sol.log
simulatedTrajectory.controls
simulatedTrajectory.modes
simulatedTrajectory.states
simulatedTrajectory.times
simulatedTrajectory.log
BOCOP HJB
Compute value function
backwards in time
Simulate Trajectory
upwards in time
Controls
Modes
Dynamics
Interpolation
Randomness
Parallelization
OpenMP
Interface
Bocop core
Third Party

Bibliography

[1] E. J. Anderson, *On the existence of supply function equilibria*, Mathematical Programming, 140 (2013), pp. 323–349.

[2] E. J. Anderson, P. Holmberg, and A. B. Philpott, *Mixed strategies in discriminatory divisible-good auctions*, The RAND Journal of Economics, 44 (2013), pp. 1–32.

[3] E. J. Anderson and X. Hu, *Finding supply function equilibria with asymmetric firms*, Operations Research, 56 (2008), pp. 697–711.

[4] E. J. Anderson and A. B. Philpott, *Optimal offer construction in electricity markets*, Mathematics of operations research, 27 (2002), pp. 82–100.

[5] S. Athey, *Single crossing properties and the existence of pure strategy equilibria in games of incomplete information*, Econometrica, 69 (2001), pp. 861–889.

[6] J.-P. Aubin and I. Ekeland, *Estimates of the duality gap in nonconvex optimization*, Mathematics of Operations Research, 1 (1976), pp. 225–245.

[7] R. J. Aumann and S. Hart, *Handbook of game theory with economic applications*, vol. 2, Elsevier, 1992.

[8] D. Aussel, R. Correa, and M. Marechal, *Electricity spot market with transmission losses*, Management, 9 (2013), pp. 275–290.

[9] H. Babazadeh, W. Gao, Z. Wu, and Y. Li, *Optimal energy management of wind power generation system in islanded microgrid system*, in North American Power Symposium (NAPS), 2013, IEEE, 2013.

[10] M. Bagnoli and T. Bergstrom, *Log-concave probability and its applications*, Economic theory, 26 (2005), pp. 445–469.

[11] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks*, science, 286 (1999), pp. 509–512.

[12] R. Bellman, *Dynamic Programming*, Princeton Univ. Press, Princeton, New Jersey, 1957.

[13] U. Berger, *Learning in games with strategic complementarities revisited*, Journal of Economic Theory, 143 (2008), pp. 292–301.

[14] U. Berger et al., *The convergence of fictitious play in games with strategic complementarities: A comment*, MPRA Paper, (2009).

[15] D. Bertsekas, *Convex optimization algorithms*, Athena Scientific Belmont, 2015.

[16] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2010.

[17] F. Bonnans, *Quelques aspects numériques de la commande optimale stochastique*, 2009.

[18] F. Bonnans, D. Giorgi, B. Heymann, P. Martinon, and O. Tissot, *Bocophjb 1.0. 1–user guide*, tech. rep., 2015.

[19] F. Bonnans, P. Martinon, D. Giorgi, V. Grélard, B. Heymann, L. Jinyan, S. Maindrault, and O. Tissot, *Bocop - a collection of examples*, tech. rep., 2016.

[20] F. Bonnans and P. Rouchon, *Commande et optimisation de systemes dynamiques*, Editions Ecole Polytechnique, 2005.

[21] S. Borenstein, J. Bushnell, and S. Stoft, *The competitive effects of transmission capacity in a deregulated electricity industry*, tech. rep., National Bureau of Economic Research, 1997.

[22] H. Borhan, M. A. Rotea, and D. Viassolo, *Optimization-based power management of a wind farm with battery storage*, Wind Energy, 16 (2013), pp. 1197–1211.

[23] G. W. Brown, *Some notes on computation of games solutions*, tech. rep., DTIC Document, 1949.

[24] A. E. Bryson, Jr. and Y. C. Ho, *Applied optimal control*, Hemisphere Publishing Corp., 1975.

[25] F. Camilli and M. Falcone, *An approximation scheme for the optimal control of diffusion processes*, RAIRO-Modélisation mathématique et analyse numérique, 29 (1995), pp. 97–122.

[26] F. Chaplais, *Averaging and deterministic optimal control*, SIAM Journal on Control and Optimization, 25 (1987), pp. 767–780.

[27] C. M. Colson, M. H. Nehrir, and C. Wang, *Ant colony optimization for microgrid multi-objective power management*, in 2009 IEEE/PES Power Systems Conference and Exposition, PSCE 2009, 2009.

[28] S. Conti, R. Nicolosi, and S. A. Rizzo, *Optimal dispatching of distributed generators in an MV autonomous micro-grid to minimize operating costs and emissions*, in IEEE International Symposium on Industrial Electronics, 2010.

[29] J. R. Correa and N. Figueroa, *On the planners loss due to lack of information in bayesian mechanism design*, in Algorithmic Game Theory, Springer, 2009, pp. 72–84.

[30] M. G. Crandall, H. Ishii, and P.-L. Lions, *Users guide to viscosity solutions of second order partial differential equations*, Bulletin of the American Mathematical Society, 27 (1992), pp. 1–67.

[31] K. Debrabant and E. Jakobsen, *Semi-lagrangian schemes for linear and fully non-linear diffusion equations*, Mathematics of Computation, 82 (2013), pp. 1433–1462.

[32] A. Eajal, E. F. El-Saadany, Y. Elrayani, and K. Ponnambalam, *Two-stage stochastic power generation scheduling in microgrids*, in Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on, IEEE, 2014, pp. 1–6.

[33] A. S. Edlin and C. Shannon, *Strict monotonicity in comparative statics*, Journal of Economic Theory, 81 (1998), pp. 201–219.

[34] J. F. Escobar and A. Jofré, *Equilibrium analysis for a network market model*, in Robust Optimization-Directed Design, Springer, 2006, pp. 63–72.

[35] ——, *Monopolistic competition in electricity networks with resistance losses*, Economic theory, 44 (2010), pp. 101–121.

[36] ——, *Equilibrium analysis of electricity auctions*, Department of Economics Stanford University, (2014).

[37] M. Falcone and R. Ferretti, *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.

[38] N. Figueroa, A. Jofré, and B. Heymann, *Cost-minimizing regulations for a wholesale electricity market*, (2015).

[39] F. Garcia and C. Bordons, *Optimal economic dispatch for renewable energy microgrids with hybrid storage using Model Predictive Control*, in IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, 2013.

[40] R. Gibbons, *Game theory for applied economists*, Princeton University Press, 1992.

[41] M. Grant and S. Boyd, *Graph implementations for nonsmooth convex programs*, in Recent Advances in Learning and Control, Springer-Verlag Limited, 2008, pp. 95–110.

[42] ——, *CVX: Matlab software for disciplined convex programming, version 2.1*, Mar. 2014.

[43] D. Guasch and S. Silvestre, *Dynamic battery model for photovoltaic applications*, Progress in Photovoltaics: Research and applications, 11 (2003), pp. 193–206.

[44] P. Haessig, *Dimensionnement et gestion d'un stockage d'énergie pour l'atténuation des incertitudes de production éolienne*, PhD thesis, Cachan, Ecole normale supérieure, 2014.

[45] P. Haessig, H. Ben Ahmed, and B. Multon, *Energy storage control with aging limitation*, in PowerTech, 2015 IEEE Eindhoven, IEEE, 2015, pp. 1–6.

[46] P. Haessig, B. Multon, H. Ben Ahmed, S. Lascaud, and L. Jamy, *Aging-aware nas battery model in a stochastic wind-storage simulation framework*, in PowerTech (POWERTECH), 2013 IEEE Grenoble, IEEE, 2013, pp. 1–6.

[47] N. Hatziargyriou, G. Contaxis, M. Matos, J. Lopes, G. Kariniotakis, D. Mayer, J. Halliday, G. Dutton, P. Dokopoulos, A. Bakirtzis, J. Stefanakis, A. Gigantidou, P. O'Donnell, D. McCoy, M. Fernandes, J. Cotrim, and A. Figueira, *Energy management and control of island power systems with increased penetration from renewable sources*, 2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309), 1 (2002).

[48] B. Heymann, F. Bonnans, J Frédéric Bonnans Silva, and G. Jimenez, *A stochastic continuous time model for microgrid energy management.* 2016.

[49] B. Heymann, J. F. Bonnans, P. Martinon, F. Silva, F. Lanas, and G. Jimenez, *Continuous optimal control approaches to microgrid energy management*, (2015).

[50] B. Heymann and A. Jofré, *Mechanism design and auctions for electricity network.* 2016.

[51] A. Hooshmand, B. Asghari, and R. Sharma, *A novel cost-aware multi-objective energy management method for microgrids*, in Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES, 2013.

[52] X. Hu and D. Ralph, *Using epecs to model bilevel games in restructured electricity markets with locational prices*, Operations research, 55 (2007), pp. 809–827.

[53] C.-C. Huang, M.-J. Chen, Y.-T. Liao, and C.-N. Lu, *DC microgrid operation planning*, in Renewable Energy Research and Applications (ICRERA), 2012 International Conference on, IEEE, 2012.

[54] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit*, Control and Optimization, (1965).

[55] Y. Ji, J. Wang, S. Yan, W. Gao, and H. Li, *Optimal microgrid energy management integrating intermittent renewable energy and stochastic load*, 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, (2015), pp. pp. 334–338.

[56] H. Kanchev, B. Francois, and V. Lazarov, *Unit commitment by dynamic programming for microgrid operational planning optimization and emission reduction*, in Electrical Machines and Power Electronics and 2011 Electromotion Joint Conference (ACEMP), 2011 International Aegean Conference on, IEEE, 2011.

[57] M. A. Katsoulakis, *Viscosity solutions of second order fully nonlinear elliptic equations with state constraints*, Indiana University Mathematics Journal, 43 (1994), pp. 493–520.

[58] H. J. Khasawneh and M. S. Illindala, *Battery cycle life balancing in a microgrid through flexible distribution of energy and storage resources*, Journal of Power Sources, 261 (2014).

[59] P. D. Klemperer and M. A. Meyer, *Supply function equilibria in oligopoly under uncertainty*, Econometrica: Journal of the Econometric Society, (1989), pp. 1243–1277.

[60] M. Koller, T. Borsche, A. Ulbig, and G. Andersson, *Defining a degradation cost function for optimal control of a battery energy storage system*, in PowerTech (POWERTECH), 2013 IEEE Grenoble, IEEE, 2013, pp. 1–6.

[61] J.-J. Laffont and D. Martimort, *The theory of incentives: the principal-agent model*, Princeton university press, 2009.

[62] J.-J. Laffont and J. Tirole, *The dynamics of incentive contracts*, Econometrica: Journal of the Econometric Society, (1988), pp. 1153–1175.

[63] H. P. Li, C. Z. Zang, P. Zeng, H. B. Yu, and Z. W. Li, *Scenario-based stochastic programming strategy for microgrid energy scheduling considering uncertainties*, Applied Mechanics and Materials, 672-674 (2014), pp. 1322–1328.

[64] H. Liu, Y. Wu, C. Qian, and X. Liu, *The application of dynamic programming in the stand-alone micro-grid optimal operation*, in Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific, IEEE, 2012.

[65] D. McAdams, *Isotone equilibrium in games of incomplete information*, Econometrica, 71 (2003), pp. 1191–1214.

[66] P. Milgrom and J. Roberts, *Rationalizability, learning, and equilibrium in games with strategic complementarities*, Econometrica: Journal of the Econometric Society, (1990), pp. 1255–1277.

[67] ——, *Adaptive and sophisticated learning in normal form games*, Games and economic Behavior, 3 (1991), pp. 82–100.

[68] ——, *Comparing equilibria*, The American Economic Review, (1994), pp. 441–459.

[69] P. R. Milgrom and R. J. Weber, *A theory of auctions and competitive bidding*, Econometrica: Journal of the Econometric Society, (1982), pp. 1089–1122.

[70] R. B. Myerson, *Optimal auction design*, Mathematics of operations research, 6 (1981), pp. 58–73.

[71] M. Y. Nguyen, Y. T. Yoon, and N. H. Choi, *Dynamic programming formulation of micro-grid operation with heat and electricity constraints*, in Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009, IEEE, 2009.

[72] N. Nisan, *Introduction to mechanism design (for computer scientists)*, Algorithmic game theory, 209 (2007), p. 242.

[73] D. E. Olivares, J. D. Lara, C. A. Cañizares, and M. Kazerani, *Stochastic-predictive energy management system for isolated microgrids*, Smart Grid, IEEE Transactions on, 6 (2015), pp. 2681–2693.

[74] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou, *Trends in microgrid control*, IEEE Transactions on Smart Grid, 5 (2014).

[75] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, and D. Saez, *A microgrid energy management system based on the rolling horizon strategy*, IEEE Transactions on Smart Grid, 4 (2013), pp. 996–1006.

[76] H. Pham, *Continuous-time stochastic control and optimization with financial applications*, vol. 61, Springer Science & Business Media, 2009.

[77] L. Pontryagin, V. Boltyanski, R. Gamkrelidze, and E. Michtchenko, *The Mathematical Theory of Optimal Processes*, Wiley Interscience, New York, 1962.

[78] S. A. Pourmousavi, R. K. Sharma, and B. Asghari, *A framework for real-time power management of a grid-tied microgrid to extend battery lifetime and reduce cost of energy*, in 2012 IEEE PES Innovative Smart Grid Technologies, ISGT 2012, 2012.

[79] P. J. Reny, *On the existence of monotone pure-strategy equilibria in bayesian games*, Econometrica, 79 (2011), pp. 499–553.

[80] P. J. Reny and S. Zamir, *On the existence of pure strategy monotone equilibria in asymmetric first-price auctions*, Econometrica, 72 (2004), pp. 1105–1125.

[81] Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, *Optimal power flow management for grid connected PV systems with batteries*, Sustainable Energy, IEEE Transactions on, 2 (2011), pp. 309–320.

[82] T. Rockafellar, *Fundamental of optimization.*

[83] D. U. Sauer and H. Wenzl, *Comparison of different approaches for lifetime prediction of electrochemical systems—using lead-acid batteries as example*, Journal of Power Sources, 176 (2008), pp. 534–546.

[84] C. Shannon, *Weak and strong monotone comparative statics*, Economic Theory, 5 (1995), pp. 209–227.

[85] R. K. Sundaram, *A first course in optimization theory*, Cambridge university press, 1996.

[86] D. M. Topkis, *Minimizing a submodular function on a lattice*, Operations research, 26 (1978), pp. 305–321.

[87] ——, *Supermodularity and complementarity*, Princeton University Press, 1998.

[88] N. Touzi, *Stochastic control and application to finance*, Scuola Normale Superiore, Pisa. Special Research Semester on Financial Mathematics, (2002).

[89] U.S. Department of Energy, *The smart grid: An introduction*, 2010.

[90] T. Van Zandt and X. Vives, *Monotone equilibria in bayesian games of strategic complementarities*, Journal of Economic Theory, 134 (2007), pp. 339–360.

[91] X. Vives, *Nash equilibrium with strategic complementarities*, Journal of Mathematical Economics, 19 (1990), pp. 305–321.

[92] ——, *Complementarities and games: New developments*, Journal of Economic Literature, 43 (2005), pp. 437–479.

[93] A. Waechter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming, 106 (2006).

[94] A. Zakariazadeh, S. Jadid, and P. Siano, *Smart microgrid energy and reserve scheduling with demand response using stochastic optimization*, International Journal of Electrical Power & Energy Systems, 63 (2014), pp. 523–533.

[95] Y. Zhang and Q.-S. Jia, *Optimal operation for energy-efficient building micro-grid*, in Control Conference (CCC), 2015 34th Chinese, IEEE, 2015.

[96] B. Zhao, X. Zhang, J. Chen, C. Wang, and L. Guo, *Operation optimization of standalone microgrids considering lifetime characteristics of battery energy storage system*, Sustainable Energy, IEEE Transactions on, 4 (2013), pp. 934–943.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Contributions mathématiques pour la régulation et l'optimisation de la production d'électricité

Mots clefs : électricité, micro réseau, marchés en réseaux, optimisation, vieillissement, mécanismes d'incitation.

Résumé : Nous présentons notre contribution sur la régulation et l'optimisation de la production d'électricité.
La première partie concerne l'optimisation de la gestion d'un micro réseau. Nous formulons le programme de gestion comme un problème de commande optimale en temps continu, puis nous résolvons ce problème par programmation dynamique à l'aide d'un solveur développé dans ce but : BocopHJB. Nous montrons que ce type de formulation peut s'étendre à une modélisation stochastique. Nous terminons cette partie par l'algorithme de poids adaptatifs, qui permet une gestion de la batterie du micro réseau intégrant le vieillissement de celle-ci. L'algorithme exploite la structure à deux échelles de temps du problème de commande.
La seconde partie concerne des modèles de marchés en réseaux, et en particulier ceux de l'électricité. Nous introduisons un mécanisme d'incitation permettant de diminuer le pouvoir de marché des producteurs d'énergie, au profit du consommateur. Nous étudions quelques propriétés mathématiques des problèmes d'optimisation rencontrés par les agents du marché (producteurs et régulateur). Le dernier chapitre étudie l'existence et l'unicité des équilibres de Nash en stratégies pures d'une classe de jeux Bayésiens à laquelle certains modèles de marchés en réseaux se rattachent. Pour certains cas, un algorithme de calcul d'équilibre est proposé. Une annexe rassemble une documentation sur le solveur numérique BocopHJB.

Title : Mathematical contributions for the optimization and regulation of electricity production

Keywords : electricity, microgrid, network markets, optimization, aging, mechanism design, auctions.

Abstract : We present our contribution on the optimization and regulation of electricity production.
The first part deals with a microgrid Energy Management System (EMS). We formulate the EMS program as a continuous time optimal control problem and then solve this problem by dynamic programming using BocopHJB, a solver developed for this application. We show that an extension of this formulation to a stochastic setting is possible. The last section of this part introduces the adaptative weights dynamic programming algorithm, an algorithm for optimization problems with different time scales. We use the algorithm to integrate the battery aging in the EMS.
The second part is dedicated to network markets, and in particular wholesale electricity markets. We introduce a mechanism to deal with the market power exercised by electricity producers, and thus increase the consumer welfare. Then we study some mathematical properties of the agents' optimization problems (producers and system operator). In the last chapter, we present some pure Nash equilibrium existence and uniqueness results for a class of Bayesian games to which some networks markets belong. In addition we introduce an algorithm to compute the equilibrium for some specific cases. We provide additional information on BocopHJB (the numerical solver developed and used in the first part of the thesis) in the appendix.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France