UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

MODELING AND ANALYSIS OF ELECTRICITY AUCTIONS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
MEMORIA PARA OPTAR AL TITULO DE
INGENIERO CIVIL MATEMÁTICO

DANIEL ESTEBAN PEREDA HERRERA

PROFESOR GUÍA:
RENÉ ALEJANDRO JOFRÉ CÁCERES

MIEMBROS DE LA COMISIÓN:
LUCE BROTCORNE -
HÉCTOR RAMIREZ CABRERA

Este trabajo ha sido parcialmente financiado por CMM Conicyt PIA AFB170001

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO Y GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
POR: DANIEL ESTEBAN PEREDA HERRERA
FECHA: 2019
PROF. GUÍA: RENÉ ALEJANDRO JOFRÉ CÁCERES

MODELING AND ANALYSIS OF ELECTRICITY AUCTIONS

Este trabajo consiste principalmente en desarrollar y analizar algoritmos, para encontrar las estrategias óptimas en mercados de electricidad modelados de manera realista.

En el Capítulo 1, se presenta el modelo general y el problema de optimización a estudiar junto con resultados teóricos previos que prueban la existencia del óptimo. Se resuelve este problema para tamaños pequeños y medianos en el caso en el que las funciones de costo son lineales por pedazos y cuadráticas utilizando algoritmos desarrollados en esta tesis, los cuales se presentan y prueban su correctitud en este mismo capítulo. Éstos se basan en explotar la forma en la cual se asignan las cantidades óptimas dependiendo de la demanda y estrategias de cada jugador. Se muestran resultados para ambos tipos de funciones de costo y se hace un análisis de sensibilidad.

En el Capítulo 2, se presenta un problema de optimización alternativo basado en un enfoque moderno, el cual, simplifica el problema al suponer que los generadores obtienen información sobre sus rivales luego de haber jugado, de manera que una empresa puede asignar probabilidades a los escenarios posibles de sus competidores y optimizar su pago esperado. Se muestra una heurística basada en un método de penalización para resolver el problema en el caso linear por partes y se prueba que es un esquema de penalización exacto. Además, se dan ideas de como aplicar heurísticas similares a otros casos. En este capítulo, se comparan ambos enfoques y se muestra que si las probabilidades asignadas a los distintos escenarios son cercanas a las del equilibrio en estrategias mixtas, entonces los valores óptimos obtenidos en ambas formulaciones son cercanos, con diferencias del orden del 0.001%, de manera que al utilizar información pública del mercado, juegos anteriores y resultados para tamaños pequeños - medianos, se pueden extrapolar las probabilidades y resolver el problema para tamaños mayores, para los cuales, no era posible utilizar el enfoque del cálculo de equilibrios de Nash en estrategias mixtas.

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO Y GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
POR: DANIEL ESTEBAN PEREDA HERRERA
FECHA: 2019
PROF. GUÍA: RENÉ ALEJANDRO JOFRÉ CÁCERES

MODELING AND ANALYSIS OF ELECTRICITY AUCTIONS

This work consists mainly in developing and analyzing algorithms, to find the optimal strategies in realistically modeled electricity markets.

In Chapter 1, the general model and the optimization problem to be studied are presented together with previous theoretical results that prove the existence of the optimum. This problem is solved for small and medium sizes in the case where the cost functions are piecewise linear and quadratic using algorithms developed in this thesis, which are presented and prove their correctness in this chapter. These are based on exploiting the way in which the optimal amounts are allocated depending on the demand and strategies of each player. Results are shown for both types of cost functions and a sensitivity analysis is performed.

In Chapter 2, an alternative optimization problem is presented based on a modern approach, which simplifies the problem by assuming that the generators obtain information about their rivals after playing, so that a company can assign probabilities to the scenarios possible from your competitors and optimize your expected payment. A heuristic is shown based on a penalty method to solve the problem in the piecesise linear case and it is proved that it is an exact penalty scheme. In addition, ideas are given on how to apply similar heuristics to other cases. In this chapter, both approaches are compared and it is shown that if the probabilities assigned to the different scenarios are close to those of equilibrium in mixed strategies, then the optimal values obtained in both formulations are close, with differences of the order of 0.001% , so that by using public market information, previous games and results for small - medium sizes, It's possible to extrapolate the odds and solve the problem for larger sizes, for which it was not possible to use the Nash equilibrium approach.

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia, la cual me ha apoyado desde pequeño. A mi mamá, papá, hermanos, abuelos y tías. Agradezco su apoyo incondicional, los valores y todo lo que me han enseñado.

En segundo lugar, me gustaría agradecer a mis amigos, especialmente Rodrigo Zelada y Pablo Arratia por ser unos payasos y pasar re buenos momentos, a Felipe Matus y Diego Marchant por el apoyo y darme trabajo todos los semestres, a Reidmen Aróstica y Felipe Atenas por todas las horas que pasamos estudiando y ayudándonos con los ramos, además de las conversaciones interesantes que teníamos. A Sofía Pontigo y Beatriz Zenteno, que siempre me apoyaron y pasamos buenos momentos sobre todos los primeros años de universidad. A Martín Rapaport, Vicente Ocqueteau, Manuel Suil, Juan D'Etigny por las buenas conversaciones y bromas y en general, a toda mi generación del dim.

Dentro de la gente del dim también quiero agradecer a las y los funcionarios, a Natacha Astromujoff, Oscar Mori y especialmente a Eterin y Karen por tener siempre buena disposición y ayudarme con lo que fuese necesario.

Quiero agradecer también a mi profesor guía, Alejandro Jofré por no solo por su constante apoyo y preocupación durante este trabajo, si no también por su comprensión, paciencia y confianza. Agradezco su dedicación y entusiasmo por intentar impulsarme a ser mejor estudiante. También a mi profesora guía durante mi estadia en Inria, Luce Brotcorne por su dedicación, preocupación y por ayudarme harto en adaptarme y lograr vivir bien en Francia.

Por último, pero no menos importante, quiero agradecer a Héctor Ramírez por aceptar ser parte de esta comisión y por ayudarme con mi postulación al doctorado.

Contents

Introduction

During the past twenty years many countries have liberalized their electricity sector. Reform began in the 1980's when the Chilean goverment introduced new legislation that privatized the majority of the electrical generators. However, it was the structual reform to the electricity industry introduced in Britain (1990) that became the most paradigmatic. These institutional reforms introduced markets as a recurring term in the modern literature of the electricity sector and have raised fundamental questions to economists, operation researches, engineers, and mathematicians. Even more recently, the massive entry of renewable energies has increased the number of questions without answers in energy markets there are few results in both the theoretical and numerical fields. Therefore his study can be a great contribution to a country like Chile where these energy sources increase every year and is one of the main motivations of this work.

In this thesis an electricity spot market is modeled and it is provided a game theorical and a scenario approach framework for its study.

In the first part the model is presented following the paper from A. Jofré and J.Escobar [13]. While not considering any specific actual market design, that paper focuses on a spot market which functions similarly to markets in Britain, New Zealand, the Us, Spain and Colombia. In the model, firms bid functions representing their cost functions. In principe, these functions may or may not reflect actual cost but must belong to a set of functions exogenously defined. Then, given the information revealed by producers during the auction process, a central agent runs a minimum cost program that respects physical network constraints. Firms are dispatched according to solutions for the minimum cost program and are paid the marginal cost of energy at their nodes.

Then a game theory approach is studied in the game among producers,i.e, a nash equilibrium outcome for a strategic form game among producers that fully foresee the consequences of their actions is studied. In particular, firms are aware that in order to get revenues as high as possible they can manipulate the minimum cost program during the auction process.

For the numerical results of the first chapter, quadratic and piecewise linear bids are studied when there are few number of players with a few numbers of steps in the discretization in the cost functions.

The quadractic case is similar to the one from the papers from D. Aussel, P. Bendotti and M. Pištěk [4] and [5] they study quadratic bids $a_i q_i + b_i q_i^2$ with $a_i > 0$ and $b_i \geq 0$, characterize the equilibria depending on a different values of the demand and the best response of a producers, that is, the optimal bids maximizing his profit. The main differences with these works are: First the demand in our model is unknown, although the demand probability distribution is known by the ISO, while in the paper the demand is deterministic and it is known both by the ISO and by the players. Second, they only considered the case when there are no resistance losses. Third they did not provide any algorithms from the problem.

The piecewise linear case is similar to the paper from M. Fampa , L.A. Barroso, D. Candal and L. Simonetti [14] here they only consider the case when the bids are linear functions and there are not resistance losses, although they have 3 different models, the one they solve is the one similar to ours and provide a heuristic to find the optimum of the bilevel problem for a fixed demand an generator on the simplified model. The ideas of this paper are considered to extend them to the case of piecewise linear bids and stochastic demand.

Chapter 1

Nash Equilibrium Approach

1.1 The Model

This section is based on the paper from A. Jofré, J. Escobar [13]. We describe the model and the main result of the paper is presented.

There is a network that consists of a set of nodes $\{1, ..., N\}$ and a set of edges. $\{1, ..., E\} \subseteq \{1, ..., N\} \times \{1, ..., N\}$. $G \subseteq \{1, ..., N\}$ is the set of nodes where there is an electricity producer. We also consider a central agent that can set production plans while respecting some network constraints.

Transactions are organized by means of an auction, which takes place as follows. First, firms submit simultaneously functions $c = (c_n)_{n \in G}$, which must belong to a set exogenously given. Second, a vector of demands $\mathrm{d} = (\mathrm{d}_n)_{n=1}^{N}$, where $\mathrm{d}_n \geq 0$ is realized. Third, after observing the vectors of bids c and demands d the central agent runs a minimum cost program subject to a number of network constraints. Fourth, firms produce as mandated by the minimum cost program and are paid marginal cost of electricity at their nodes. Finally, payoffs accrue.

1.1.1 The Dispatch Program

This subsection details the minimum cost program. Roughly speaking, after observing the bids $c = (c_n)_{n \in N}$ and the state of the demand d, the central agent minimizes the total cost of production:

$$\sum_{n \in G} c_n(q_n)$$

Subject to the technological and physical constraints. These constraints are specified below.

NODAL BALANCES. At each node, avaible power must satisfy nodal demand. Due to thermal considerations, there are power flow losses in the transmission lines. A good approximation for the losses is a quadratic function of the flow. Indeed, if the flow over $e \in E$ is f_e, the loss is given by $r_e f_e^2$, where $r_e \geq 0$ is the line resistance. Assuming that losses are split between the nodes associated to each line, the nodal power balances are:

$$\sum_{e \in K_n} \frac{r_e}{2} f_e^2 + d_n \leq q_n + \sum_{e \in K_n} f_e sgn(e, n), \quad n \in G \tag{1.1}$$

$$\sum_{e \in K_n} \frac{r_e}{2} f_e^2 + d_n \leq \sum_{e \in K_n} f_e sgn(e, n), \quad n \notin G \tag{1.2}$$

Where K_n is the set of transmission lines connecting node n and $sgn(e, n)$ is equal to 1 or -1 depending on the orientation of the graph and whenever $e = (n, m)$, $sgn(e, n) = -sgn(e, m)$. We also denote $K = \cup_{n \in G} K_n$. The left hand side of (1) is half the sum of all the losses related to node n plus nodal demand d_n. The right hand side of (1) is the production of generator n plus the sum of effective flows. The interpretation of (2) is similar, but for nodes $n \notin G$ there is no local producer at the local demand must be satisfied with external production.

GENERATION CONSTRAINTS. Each generator has a nonempty production set:

$$q_n \in [0, \bar{q}_n] \tag{1.3}$$

Where $\bar{q}_n \geq 0$.

TRANSMISSION CONSTRAINTS Each transmission line $e \in E$ has a maximum capacity: $\underline{f}_e \leq f_e \leq \bar{f}_e$. Where $\underline{f}_e \leq 0 \leq \bar{f}_e$. More generally, we considerer the constraints

$$f \in F \tag{1.4}$$

where $F \subseteq R^E$ is a convex compact set. This formulatios is general enough to include Kirchhoff's voltage law constraints and several other power network constraints.

Given a vector of demands $d = (d_n)_{n=1}^N$, we define :

$$\Omega(d) = \big\{(f, q) \in \mathbb{R}^E \times \mathbb{R}^G : (f, q) \text{ satisfies } (1.1), ..., (1.4)\big\}$$

Set of feasible plans which turns out to be compact convex set. thus, the central agent solves the following dispatch program:

$$min \left\{ \sum_{n \in G} c_n(q_n) : (f, q) \in \Omega(d) \right\} \tag{1.5}$$

We denote its optimal value $OPT(c, d)$, and define the set:

$$Q(c, \mathrm{d}) = \big\{q \in \mathbb{R}^G : f \in \mathbb{R}^E, (f, q) \text{ is a solution of } (1.5)\big\}$$

of optimally generated quantities $q = (q_n)_{n \in G}$.

Nodal prices are set as shadow values associated to the nodal power balances. That is, the price at each node is a dual variable on the power balance constraint at this node.

1.1.2 The Bidders

Now we focus our attention on the interaction among producers. Broadly speaking, this game consists of each firm independently manipulating the dispatch program (5) (and so quantities $q(c, \mathrm{d}) \in Q(c, \mathrm{d})$ and prices $\alpha(c, \mathrm{d})$) in order to obtaint revenues as high as possible.

Suppose that firm n produces q_n and is paid a price p_n per each unit produced, its payoff is $u_n(p_n, q_n)$ where: $u_n : \mathbb{R}^2 \to \mathbb{R}$ is a continuous function. While $u_n(p_n, q_n) = p_n q_n - \hat{c}_n(q_n)$ (where $\hat{c}_n$ is the actual cost function) is the most important case in practice, at this stage we keep the model as general as possible. For $p, q \in \mathbb{R}^{|G|}$ we also define u by: $u(p, q) = \big(u_1(p_1, q_1), ..., u_{|G|}(p_{|G|}, q_{|G|})\big)$.

At the beginning of the game, firms bid simultaneously their cost of production functions $c = (c_n)_{n=1}^N$ to the central agent. the bid of firm n, c_n may or may not reflect its actual cost. Indeed, if a firm finds profitable lying in respect to its actual cost $\hat{c}_n$, then it will do it. We, however, restrict the set of feasible decisions to firm n. Thus, firm n must choose a function belonging to a nonempty set of fuctions S_n that is exogenously defined. We assume S_n only contains functions c_n that are convex real-valued (thus continuous), non-decreasing and, just for simplicity, such that $c_n(0) = 0$.

We note that firms bid functions that represent their cost functions. In contrast, in actual markets firms bid supply functions, functions representing their marginal costs. Under the assumption $c_n(0) = 0$ for all $c_n \in S_n$, there is no strategic difference between bidding cost functions and supply functions. So, for the sake of brevity, we assume firm bid functions representing their cost functions.

When submitting its function c_n, firm n does not know the demand vector d. However, each firm knows the probability distribution P of d and (in particular) its support D. All aspects of the game are commond knowledge

Definition 1.1 *Noncooperative Equilibrium*

Let Γ be the described model. A noncooperative equilibrium (henceforth equilibrium) of the game Γ is a 3-Tuple $(q, \lambda, (\bar{m}_n)_{n \in G})$ such that:

- q is a selection from $Q(\cdot, \cdot)$, so a solution of the dispatch program.
- λ is a selection from $\Lambda(\cdot, \cdot)$, so a Lagrange multiplier of the dispatch program.
- $\bar{m} = (\bar{m}_n)_{n \in G}$ is a non-trivial mixed strategy Nash equilibrium of the normal form

game among producers $\bar{\Gamma}(\lambda, q) = (S_n, V_n)_{n\in G}$, where each generator chooses a strategy c_n belonging to the set of functions S_n and obtains a payoff given by the expected profit:

$$V_n(c_n, c_{-n}) = \mathbb{E}[u_n\left(\lambda_n(c,\cdot), q_n(c,\cdot)\right)] = \int u_n\left(\lambda_n(c,\mathrm{d}), q_n(c,\mathrm{d})\right) \mathrm{d}P(\mathrm{d}), \quad c \in S$$

In this context, the selections λ and q *supports* the measure $\bar{m}$ as a first stage Nash equilibrium.

Therefore the problem can be written as:

$$(P) = \begin{cases} \max & \displaystyle\sum_{n\in G} \mathbb{E}_n[u_n(\lambda_n(c,\cdot), q_n(c,\cdot))] \\ s.t \quad (\lambda, q) \in ISO(c,P) = & \begin{cases} \min & \displaystyle\sum_{n\in G} c_n(q_n) \\ s.t & \displaystyle\sum_{\mathrm{e}\in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq q_n + \sum_{\mathrm{e}\in K_n} f_\mathrm{e} sgn(\mathrm{e},n), \quad n \in G \\ & \displaystyle\sum_{\mathrm{e}\in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq \sum_{\mathrm{e}\in K_n} f_\mathrm{e} sgn(\mathrm{e},n), \quad n \notin G \\ & q_n \in [0, \bar{q}_n] \\ & f \in F \end{cases} \end{cases}$$

These types of problems are known as Bilevel problems and they are of great importance in different areas of engineering.

1.2 Existence of Noncooperative Equilibrium

Consider the following assumptions.

`ASSUMPTION 1` There exists p^* such that for all $c_n \in S_n$, all $q_n \in \mathbb{R}$, and all $x \in \partial c_n(q_n)$, $|x| \leq p^*$. Put differently, for all $c_n \in S_n$ and all $q_n \in \mathbb{R}$, the derivative

$$c_n^+(q_n) := \lim_{h \searrow 0} \frac{c_n(q_n + h) - c_n(q_n)}{h} \leq p^*$$

`ASSUMPTION 2` For all $\mathrm{d} \in D$, there exists $\delta_{\mathrm{d}} > 0$ such that $\forall \hat{\mathrm{d}} \in B(\mathrm{d}, \delta_{\mathrm{d}}), \quad \Omega(\hat{\mathrm{d}}) \neq \emptyset$. (In particular, the dispatch program is feasible.)

`ASSUMPTION 3` One of the two assertions holds:

1. P is atomless;
2. For all convex compact sets $M, N \subseteq \mathbb{R}^G$, the set $u(M \times N) \subseteq \mathbb{R}^G$ is convex.

`ASSUMPTION 4` For all $n \in G$, S_n is closed under pointwise convergence.

Assumption 1 is reasonable. For example, p^* can be set as a regulated price cap. (This mitigation measure is common in power market designs; see Wilson [26]). On the other hand, note that D must be compact. In fact. D is closed (by definition) and for all $\mathrm{d} \in D$, $\Omega(\mathrm{d})$ is nonempty and, as a consequence,

$$0 \leq \sum_{n=1}^{N} \mathrm{d}_n \leq \sum_{n \in G} \bar{q}_n$$

Where the last inequality comes from the nodal balance:

$$\sum_{n=1}^{N} \mathrm{d}_n + \sum_{\mathrm{e} \in E} \frac{r_{\mathrm{e}}}{2} f_2^2 \leq \sum_{n \in G} q_n \leq \sum_{n \in g} \bar{q}_n$$

Assumption 2 together with compactness of D implies the existence of $\delta > 0$ such that for all $\mathrm{d} \in D$,

$$\forall \hat{\mathrm{d}} \in B(\mathrm{d}, \delta), \Omega(\hat{\mathrm{d}}) \neq \emptyset$$

Indeed, δ is a Lebesgue number associated to the open covering $(B(\mathrm{d}, \delta_{\mathrm{d}}) : \mathrm{d} \in D)$. This δ in a sense, reflect how tight the network is and it is called network slackness.

Theorem 1.2 *Under assumptions (1) - (4) , the game* Γ *has a noncooperative equilibrium* $(q, \lambda, (m_n)_n)$*, where* m_n *is a regular measure over the pointwise Borel* σ*-field on* S_n

The proof of this theorem can be found in A. Jofré - J. Escobar [13].

1.2.1 Example were no pure strategy Nash equilibrium exists

This example is also from the main paper from A. Jofré and J. Escobar [13]. Here we will see that the mixed strategy equilibrium solution is weaker than the standard pure strategy Nash equilibrium. In fact, a pure stratregy equilibrium is also a mixed strategy equilibrium (consult Fundenberg and Tirole [12] for additional discussion). We consider the mixed strategy equilibrium as a solution to the game among producers because very often the pure strategy equilibrium fails to exist.

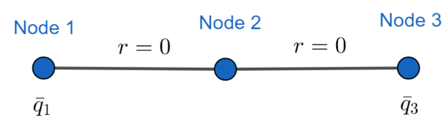


Figure 1.1: Three-node model

To see this, consider the following simplified case where we have a three-node model ilustrated by figure 1.1. Demand $\mathrm{d} \in \mathbb{R}$ is located at node 2 and distributed according to F. Node 1 (resp. node 3) has a generator with production capacity $\bar{q}_1$ (resp. $\bar{q}_3$). There are no transmisson constraints, and transmission losses are 0. For simplicity, consider the symmetric case in which $\bar{q}_1 = \bar{q}_3 = \bar{q}$. We assume that it is always feasible to satisfy the demand : $2\bar{q} > \mathrm{d}$ for all $\mathrm{d} \in D$. Additionally, we assume that the probabilities that only one generator is dispatched and that both generators are dispatches are stricly positive:

$$P[\mathrm{d} < \bar{q}] \in (0, 1)$$

Generator n's payoff function is $u_n(p_n, q_n) = p_n q_n$ (that is,costs are 0). The auction design is such that each firm n is allowed to bid a single price $p_n \in [0, p^*]$, where p^* is a price cap, which represents its marginal cost function. Equivalently, each firm may be seen as bidding a linear cost function. More formally, the set of bids can be written as:

$$S_n = \{c_n : \mathbb{R} \to \mathbb{R}_+ : c_n \text{ is linear, } c_n(0) = 0, \text{ and } c_n'(0) \in [0, p^*]\}$$

There is no strategic difference between bidding prices representing marginal costs and bidding costs. So, we identify each $c_n \in S_n$ with its derivative $c_n' = p_n$.

Given bids $p = (p_1, p_3)$ and demand d , the dispatch program can be written as:

$$\min\{p_1 q_1 + p_3 q_3 : q_1 + q_3 \geq \mathrm{d}, q_\mathrm{i} \in [0, \bar{q}]\}$$

The solution set to this program, $Q(p, \mathrm{d})$, needs not to be a singleton. Indeed, if $p_1 = p_3$, the dispatch problem has a continuoum of solutions. The shadow price of electricity is nodeindependent (this is so because in this simple model there are no transmission constraints), and given by:

$$\Lambda_{\mathrm{i}}(p_{\mathrm{i}}, p_j, \mathrm{d}) = \begin{cases} \min\{p_1, p_3\} & \text{if } \mathrm{d} < \bar{q} \\ [\min\{p_1, p_3\}, \max\{p_1, p_3\}] & \text{if } \mathrm{d} = \bar{q} \\ \max\{p_1, p_3\} & \text{if } \mathrm{d} > \bar{q} \end{cases}$$

Consider selections $q(p, \mathrm{d}) \in Q(p, \mathrm{d})$ and $\lambda(p, \mathrm{d}) \in \Lambda(p, \mathrm{d})$. We will show that the induced game among generators cannot have a pure strategy equilibrium. If not, there is a pure strategy equilibrium $p_1, p_3 \in [0, p^*]$. Suposse first that $p_1 < p_3$. Then, firm i could increase its payoff by slightly increasing its bid. So, $p_1 = p_3$. If $p_1 = p_3 = 0$, then either firm could increase its payoff by bidding the price cap p^*. So $p_1 = p_3 > 0$. Then, no matter what selection $q(p, \mathrm{d})$ is, there is one firm, say n, whose expected payoff is, at most,

$$p_n \mathbb{E}[\mathrm{d}]/2$$

But firm n, by bidding $p_n - \varepsilon$ (with $\varepsilon > 0$ small), could get a payoff:

$$p_m \bar{q}(1 - F(\bar{q})) + (p_n - \varepsilon) \int_0^{\bar{q}} \psi \mathrm{d}F(\psi)$$

Where we considered the normalized case (so that $0 \leq \bar{q} \leq 1$) Note that:

$$2\bar{q} > sup\{\mathrm{d} : \mathrm{d} \in D\} > \frac{\int_{\bar{q}}^1 \psi \mathrm{d}F(\psi)}{1 - F(\bar{q})} - \frac{\int_0^{\bar{q}} \psi \mathrm{d}F(\psi)}{1 - F(\bar{q})}$$

Where the inequalities follows from the feasiblility constraint, i.e, $2\bar{q} > 1$.

Therefore,

$$\bar{q}(1 - F(\bar{q})) + \frac{1}{2} \int_0^{\bar{q}} \psi \mathrm{d}F(\psi) - \frac{1}{2} \int_{\bar{q}}^1 \psi \mathrm{d}F(\psi) > 0$$

which implies that:

$$\bar{q}(1 - F(\bar{q})) + \int_0^{\bar{q}} \psi \mathrm{d}F(\psi) > \frac{1}{2} \int_0^1 \psi \mathrm{d}F(\psi)$$

Therefore, for $\varepsilon > 0$ small enough

$$p_n \mathbb{E}[\mathrm{d}]/2 < p_m \bar{q}(1 - F(\bar{q})) + (p_n - \varepsilon) \int_0^{\bar{q}} \psi \mathrm{d}F(\psi)$$

i.e, bidding $p_n - \varepsilon$ is stricly more profitable than bidding p_n for firm n. It follows that no pure strategy equilibrium can exist.

1.3 Bilevel Optimization

Here we follow Dempe's Book [11]. Where he summarizes the state of the art in bilevel problems.

Bilevel optimization problems are hierarchical optimization problems of two or more players. Consider $f, g_{\mathrm{i}} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, i $= 1, ..., p$ and $Y \subseteq \mathbb{R}^n$ is a nonempty closed set, then consider the parametric optimization problem:

$$\min\{f(x,y) : g(x,y) \leq 0, y \in Y\} \tag{1.6}$$

This is the lower problem, also called follower's problem. There can be more than one decision maker/follower, for example in the case were we search for a Nash equilibrium. Let

$$\phi(x) := \min_y\{f(x,y) : g(x,y) \leq 0, y \in Y\} \tag{1.7}$$

Denote the optimal value function of problem (1.6) and

$$\Psi(x) := \{y \in Y : g(x,y) \leq 0, f(x,y) \leq \phi(x)\} \tag{1.8}$$

the solution set mapping of problem (1.6).

Lets denote $gph(\Psi) := \{(x,y) : y \in \Psi(x)\}$ to the graph of the solution set mapping Ψ, the following bilevel optimization problem

$$\min_x\{F(x,y) : G(x) \leq 0, (x,y) \in gph(\Psi), x \in X\} \tag{1.9}$$

Can be formulated with $X \subseteq \mathbb{R}^m$, $F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $G_j : \mathbb{R}^m \to \mathbb{R}$, $j = 1, ..., q$. Sometimes, this is called the upper level optimization problem or leader's problem.

Problem (1.6) , (1.9) was first formulated in an economic context by V. Stackelber [15].

Bilevel optimization problems are nonconvex and nondifferentiable optimization problems. Also even the linear-linear bilevel optimization problem is NP-Hard as we can see in [7]

Notice that problem (1.9) is not well-defined in case of multiple lower level optimal solutions. The usual approach to deal with this is using the *Optimistic or pessimistic formulation.*

Definition 1.3 *The leader may assume that the follower can be motivated to select a best optimal solution in $\Psi(x)$ with respect to the leader's objective function. This is the so-called optimistic or weak formulation of the bilevel optimization problem, adopted in most of papers:*

$$\min\{\phi_0(x) : G(x) \leq 0, x \in X\} \tag{1.10}$$

Where

$$\phi_0(x) = \min_y \{F(x,y) : y \in \Psi(x)\} \tag{1.11}$$

This problem is almost equivalent to

$$\min_{x,y} \{F(x,y) : G(x) \leq 0, x \in X, (x,y) \in gph(\Psi)\} \tag{1.12}$$

As it can be seen in [1]. If the upper level objective function is of a special type the optimistic bilevel optimization problem can be interpreted as an inverse optimization problem. [2],[16] and [28].

If this is not possible or even not allowed, the leader is forced to bound the damage resulting from an unwelcome selection of the follower resulting in the pessimistic or strong formulation of the bilevel optimization problem.

Definition 1.4 *The Pessimistic or strong formulation of the bilevel optimization problem is:*

$$\min\{\phi_p(x) : G(x) \leq 0, x \in X\} \tag{1.13}$$

Where

$$\phi_p(x) = \max_y \{F(x,y) : y \in \Psi(x)\} \tag{1.14}$$

To investigate properties, for the formulation of optimality conditions and solutions algorithms, the bilevel optimization problem can be transformed into a single level problem. For this, different approaches are possible and are the ones that were used:

Use of the Karush-Kuhn-Tucker conditions of the lower level problem. If the functionts $y \to f(x,y)$, $y \to g_i(x,y)$, i $= 1,...,p$ are differentiable and a regularity condition is satisfied for the lower level problem for all $(x,y) \in gph(\Psi)$, problem (1.12) can be replaced by:

$$\begin{cases} \min_{x,y,z}\{F(x,y) : G(x) \leq 0\}, \quad x \in X \\ \nabla_y\{f(x,y) + u^T g(x,y)\} = 0 \\ \qquad u \geq 0 \\ \qquad g(x,y) \leq 0 \\ \qquad u^T g(x,y) = 0 \end{cases} \tag{1.15}$$

It is shown in ([19]) that this approach is only possible if the lower level problem is a convex one. Problem (1.15) is a so-called mathematical program with equilibrium (or complementarity) constraints (MPEC).

There is also another techniques like the *Use of necessary optimality conditions without Lagrange multipliers* or *Use of the optimal value function.* In the last case problem (1.12) can be equivalently replaced by:

$$\min\{F(x,y) : G(x) \leq 0, x \in X, g(x,y) \leq 0, f(x,y) \leq \phi(x)\} \tag{1.16}$$

This transformation has first been used in [21],[20]. Problem (1.16) is a nonsmooth optimization problem since the optimal value function $\phi(x)$ is, even under restrictive assumptions, in general not differentiable. Moreover, the nonsmooth Mangasarian- Fromovitz constraint qualification is violated at every feasible point [22], [27].

We will be mostly focus on the mathematical program with equilibrium constraints, since it's better for numerical purposes and having the Lagrange multipliers comes in handy with the shadow prices.

1.4 Algorithms for Nash Equilibrium

In this section we'll see algorithms used to solve the Nash Equilibrium problem.

1.4.1 Lemke Howson Algorithm

The Lemke-Howson algorithm originally appeared in the paper [18] of Lemke and Howson, in 1964 and is able to deliver a Nash equilibria within all possible equilibria in a 2 player game. This algorithm resembles the simplex algorithm (from linear programming). One similarity is that both methods can take an exponential number of iterations (see Savani and von Stengel 2004 [24]). Other techniques to solve linear programs are known that run in polynomial time (e.g, the ellipsoid and interior point methods) but no such technique is known for finding Nash equilibria.

Lets follow [23] in order to introduce the basic definitions, propeties and results.

Consider a two person bimatrix game where the payoff matrices are $A_{m\times n}$ and $B_{m\times n}$. A pair of strategies (x, y) is a Nash equilibrium for game (A, B) if and only if:

$$\forall 1 \leq \mathrm{i} \leq m, x_\mathrm{i} > 0 \implies (Ay)_\mathrm{i} = \max_k (Ay)_k$$

$$\forall m+1 \leq j \leq m+n, y_j > 0 \implies (x^\top B)_j = \max_k (x^\top B)_k$$

Let $M = \{1, 2, \ldots, m\}$ and $N = \{m+1, m+2, \ldots, m+n\}$. Define the support of x by $S(x) = \{\mathrm{i} : x_\mathrm{i} > 0\}$ and the support of y analogously.

Definition 1.5 *A Bimatrix game (A, B) is non-degenerate if and only if for every strategy x of the row player $|S(x)|$ is at least the number of pure best responses to x and for every strategy y of the column player, $|S(y)|$ is bigger than or equal to the number of pure best responses to y.*

It can be assumed that the game is non-degenerate since we can slightly perturb the payoff matrices to make the game that way.

Proposition 1.6 *If (x, y) is a Nash equilibrium of a non-degenerate bimatrix game, then $|S(x)| = |S(y)|$*

Let B_j denote the column of B corresponding to action j and let A^i denote the row of A corresponding to action i. We define the following polytopes:

$$P_1 = \{x \in \mathbb{R}^M : (\forall \mathrm{i} \in M : x_\mathrm{i} \geq 0) \land (\forall j \in N : x^t B_j \leq 1)\}$$

$$P_2 = \{y \in \mathbb{R}^N : (\forall \mathrm{i} \in M : A^\mathrm{i} y \leq 1) \land (\forall j \in N : y_j \geq 0)\}$$

For a nonzero nonnegative x, we can normalize it to a stochastic vector `norml(x)` as follows:

$$norml(x) := \frac{x}{(\sum_{\text{i}} x_{\text{i}})}$$

The inequalities that define P_1 have the following meaning:

- If $x \in P_1$ meets $x_{\text{i}} \geq 0$ with equality then i is not in the support of x.
- If $x \in P_1$ meets $x^t B_j \leq 1$ with equality then j is a best response to `norml(x)`.

Let us say that $x \in P_1$ has label k, where $k \in MN = \{1, ..., m+n\}$, if either $k \in M$ and $x_k = 0$, or $k \in N$ and $x^t B_k = 1$. Similarly $y \in P_2$ has label k if either $k \in N$ and $y_k = 0$, or $k \in M$ and $A_k y = 1$. As a consequence of the Support Characterization, we have the following.

Theorem 1.7 *Suppose that $x \in P1$ and $y \in P2$, and neither x nor y is the all-zero vector. Then x and y together have all labels from 1 to k if and only if (nrml(x), nrml(y)) is a Nash equilibrium. All Nash equilibria arise in this way.*

A $\text{d} - \text{d}i\textit{mensional}$ polytope is simple if every vertex meets exactly d of the defining inequalities with equality.

Assumption : The polytopes P_1 and P_2 are simple.

As a consequence of the previous theorem we have the following:

Theorem 1.8 *A non-degenerate bimatrix game has an odd number of Nash equilibria*

Proposition 1.9 *A 2-player finite strategic game is nondegenerate if and only if, for any mixed strategy α of a player, the number of pure best responses by their opponent does not exceed $|supp(\alpha)|$.*

Therefore we have the following algorithm for a nondegenerate 2 player game (A, B) which returns a Nash equilibrium:

1. Start at $(0, 0)$.
2. Choose a label to drop.
3. Remove this label from the corresponding vertex by traversing an edge of the corresponding polytope to another vertex.
4. The new vertex will now have a duplicate label in the other polytope. Remove this label from the vertex of the other polytope and traverse an edge of that polytope to another vertex.
5. Repeat step 4 until the pair of vertices is fully labelled.

1.4.2 Example

Conider the matching pennies game. It is played between two players, Even and Odd. Each player has a penny and must secretly turn the penny to heads or tails. The players then reveal their choices simultaneously. If the ppenies match (both heads or both tails), then Even keeps both pennies, so wins one from Odd (+1 for Evenn -1 for odd). If the pennies do not match (one heads and one tails) Odd keeps both pennies, so receives one from Even (-1 for Even, +1 for Odd).

The payoff matrices are:

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$$

First we add 2 to all utilities:

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

We have the following inequalities:

$$\begin{aligned} -x_1 &\leq 0 \\ -x_2 &\leq 0 \\ x_1 + 3x_2 &\leq 1 \\ 3x_1 + x_2 &\leq 1 \end{aligned}$$

The intersection of the two non trivial constraints is at the point:

$$1/3 - x_1/3 = 1 - 3x_1$$

The vertices are:

$$V = \{(0,0), (1/3,0), (1/4,1/4), (0,1/3)\}$$

These vertices are no longer probability vectors. Recall the four inequalities of this polytope:

1. $x_1 \geq 0$: if this inequality is "binding" (ie $x_1 = 0$) that implies that the row player does not play that strategy.
2. $x_2 \geq 0$: if this inequality is "binding" (ie $x_2 = 0$) that implies that the row player does not play that strategy.

3. $x_1 + 3x_2 1$: if this inequality is binding then that implies that the utility to the column player for that particular column is as big as it can be.
4. $3x_1 + x_2 1$: if this inequality is binding then that implies that the utility to the column player for that particular column is as big as it can be.

Lets label our vertices:

- $(0, 0)$ has labels $\{0, 1\}$.
- $(1/3, 0)$ has labels $\{1, 3\}$.
- $(1/4, 1/4)$ has labels $\{2, 3\}$.
- $(0, 1/3)$ has labels $\{0, 2\}$

Similarly the vertices and labels for P_2 are:

- $(0, 0)$ has labels $\{2, 3\}$.
- $(1/3, 0)$ has labels $\{0, 3\}$.
- $(1/4, 1/4)$ has labels $\{0, 1\}$.
- $(0, 1/3)$ has labels $\{1, 2\}$

Let us apply the algorithm:

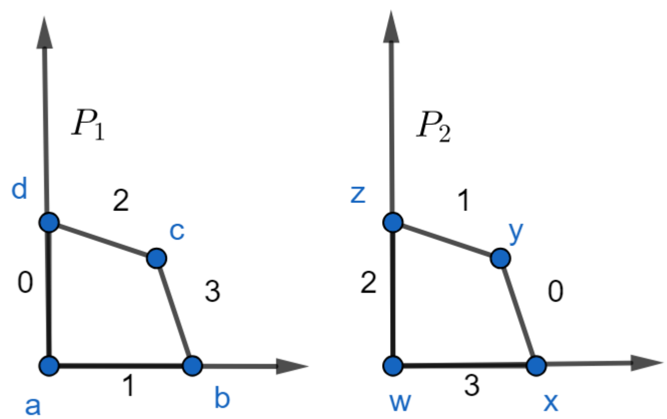


Figure 1.2: Lemke Howson

- (a, w) have labels: $\{0, 1\}, \{2, 3\}$. Drop 0 (arbitrary decision) in P_1.
- (b, w) have labels: $\{1, 3\}, \{2, 3\}$. In P_2 drop 3.
- (b, z) have labels: $\{1, 3\}, \{1, 2\}$. In P_1 drop 1.
- (c, z) have labels: $\{2, 3\}, \{1, 2\}$. In P_2 drop 2.
- (c, y) have labels: $\{2, 3\}, \{0, 1\}$. Fully labeled vertex pair.

Now we normalize these vertices and return the strategy pair:

$$((1/2, 1/2), (1/2, 1/2))$$

1.4.3 Vertex Enumeration

For a nondegenerate 2 player game (A, B) the following algorithm returns all nash equilibria:

1. For all pairs of vertices of the best response polytopes, check if the vertices have full labels.
2. Return the normalised probabilities.

For the previous example, the only par of vertices that is fully labeled is:

$$((1/4, 1/4), (1/4, 1/4))$$

which, when normalised correspond to:

$$((1/2, 1/2), (1/2, 1/2))$$

Vertex enumeration is important since most of the recent methods that improve the basic Lemke Howson method are based on making an smarter enumeration, see for instance [6].

1.4.4 Tableau

To apply the tableau method to find a Nash equilibria using the Lemke-Howson algorithm, we use the following four steps:

1. Proprocessing (elimination of stricly dominated strategies)
2. Initialization of tableaux.
3. Repeated pivoting.
4. Recover Nash equilibrium from final tableaux.

Let r_{i} be the slack in the constraint $A^{\text{i}}y \leq 1$ and let s_j be the slack in the constraint $x^t B_j \leq 1$. We obtain the system:

$$Ay + r = 1, B^t x + s = 1, \text{ and } x, y, r, s \text{ are nonnegative.}$$

In the initial tableaux, the basis is $\{r_{\text{i}} : \text{i} \in M\} \cup \{s_j : j \in N\}$

p_1 / p_2	3	4
1	3,1	1,3
2	1,3	3,1

The initial tableaux is:

$$r_1 = 1 - 3y_3 - y_4$$
$$r_2 = 1 - y_3 - 3y_4$$

$$s_3 = 1 - x_1 - 3x_2$$
$$s_4 = 1 - 3x_1 - x_2$$

We need to arbitrarily choose some x or y variable to bring in to the basis, corresponding to the arbitrary choise k_0 of label that we remove. Let's bring x_1 in. By considering the min-ratio rule s_4 must leave the basis.

$$x_1 = 1/3(1 - s_4 - x_2)$$
$$s_3 = 1/3(2 - 8x_2 + s_4)$$

Now let's bring y_4 in (because s_4 was out). So r_2 is out

$$y_4 = 1/3(1 - y_3 - r_2)$$
$$r_1 = 1/3(2 - 8y_3 + r_2)$$

x_2 in and s_3 out:

$$x_2 = 3/8(2/3 + 1/3s_4 - s_3)$$
$$x_1 = 1/3(1 - 3/8(2/3 + 1/3s_4 - s_3))$$

y_3 in and r_1 out

$$y_3 = 3/8(2/3 + 1/3r_2 - r_1)$$
$$y_4 = 1/3(1 - 3/8(2/3 + 1/3r_2 - r_1))$$

Since x_1 was the initial variable to enter the basis, and r_1 just left complementarity conditions are now satisfied. (More generally, if x_i was the initial variable to enter, we stop when x_i or its complement leaves). In a tableau, we obtain values for the basic variables by setting the non-basic variables to zero. Hence the variables' values are:

$$r = (0, 0), s = (0, 0), x = (1/4, 1/4), y = (1/4, 1/4)$$

Therefore the Nash equilibrium we just found is:

$$(norml(x), norml(y)) = ((1/2, 1/2), (1/2, 1/2))$$

In practice the 3 implementations are used, however, the best versions that have come out recently come from vertex enumeration.

1.4.5 N-Players Algorithm

Here we use an algorithm from from B.Chatterjee [10]. To explained it will be needed to introduce the game formulation and notation as in the paper. The method is capable of giving one sample Nash equilibrium out of probably many present in a given game.

`GAME FORMULATION AND NOTATION.`

A finite n-person non cooperative game in normal or strategic form is represented by a tuple

$$\Gamma = (N, \{S^{\mathrm{i}}\}_{\mathrm{i}\in N}, \{u^{\mathrm{i}}\}_{\mathrm{i}\in N})$$

Where N is a finite set of players, S^{i} is space of pure strategies of player i and u^{i} is the payoff function of player i.

A mixed strategy of player i is interpreted as a probability distribution over the space S^{i} and the space of all mixed strategies of player i is denoted by $\sum^{\mathrm{i}} = \{\sigma^{\mathrm{i}} \in \mathbb{R}^{m^{\mathrm{i}}+} : \sum_{j=1}^{m^{\mathrm{i}}} \sigma_j^{\mathrm{i}} = 1\}$ where m^{i} is the number of pure strategies player i has. For $\sigma^{\mathrm{i}} \in \sum^{\mathrm{i}}$, the probability assigned to pure strategy s_j^{i} is σ_j^{i}. The strategy space of the game is $\sum = \prod_{\mathrm{i}\in N} \sum^{\mathrm{i}}$.

If a mixed strategy combination σ is played then the probability that the pure strategies combination $s = (s_{j^1}^{\mathrm{i}}, s_{j^2}^2, ..., s_{j^n}^n)$ occurs is given by $\sigma(s) = \Pi_{\mathrm{i}\in N}\sigma_{j^{\mathrm{i}}}^{\mathrm{i}}$. In such a situation the payoff assigned to player i is given by $u^{\mathrm{i}}(\sigma) = \sum_{s\in S} \sigma(s)u^{\mathrm{i}}(s)$, where $u^{\mathrm{i}}(s)$ is the payoff to player i at the pure strategies combination s.

If $\sigma^{-\mathrm{i}}$ denotes the mixed strategy vector formed by all players except player i, then we can replace the mixed strategies combination σ by $(\sigma^{-\mathrm{i}}, \sigma^{\mathrm{i}})$.

Definition 1.10 *A Mixed strategy profile σ^* is called a Nash equilibrium of the game Γ if:*

$$u^{\mathrm{i}}(\sigma^*) \geq u^{\mathrm{i}}(\sigma^{*-\mathrm{i}}, \sigma^{\mathrm{i}}), \forall \mathrm{i} \in N, \forall \sigma^{\mathrm{i}} \in \sum^{\mathrm{i}}$$

This means that each player i could not obtain a better payoff than the one he obtains at Nash equilibrium, by changing only his own mixed strategy, i.e, leaving all other strategies unchanged.

The idea of the paper is to minimize the gap between the optimal payoff and the payoff obtained by a possible mixed strategy combination.

EQUIVALENT OPTIMIZATION FORMULATION

If β^{i} is the optimal payoff of player i then the optimization problem of player i, $\mathrm{i} \in N$ is:

$$(P^{\mathrm{i}}) \begin{cases} min & \beta^{\mathrm{i}} - u^{\mathrm{i}}(\sigma) \\ s.t & u^{\mathrm{i}}(\sigma^{-\mathrm{i}}, s_j^{\mathrm{i}}) - \beta^{\mathrm{i}} \leq 0 \quad \forall j = 1, ..., m^{\mathrm{i}} \\ & \sum_{j=1}^{m^{\mathrm{i}}} \sigma_j^{\mathrm{i}} = 1 \\ & \sigma_j^{\mathrm{i}} \geq 0 \quad \forall j = 1, ..., m^{\mathrm{i}} \end{cases}$$

where $(\sigma^{-\mathrm{i}}, s_j^{\mathrm{i}})$ denotes the mixed strategies combination in which player i plays with his j^{th} pure strategy, that is, a mixed strategy in which the j^{th} pure strategy of the i^{th} player is assigned the probability 1.

On the paper the following Lemma and Theorem are proved:

Lemma 1.11 *A necessary and sufficient condition for σ to be a Nash equilibrium of the game Γ is:*

$$\beta^{\mathrm{i}} - u^{\mathrm{i}}(\sigma) \quad \forall \mathrm{i} \in N \tag{1.17}$$

$$u^{\mathrm{i}}(\sigma^{-\mathrm{i}}, s_j^{\mathrm{i}}) - \beta^{\mathrm{i}} \leq 0 \quad \forall j = 1, ..., m^{\mathrm{i}}, \forall \mathrm{i} \in N \tag{1.18}$$

$$\sum_{j=1}^{m^{\mathrm{i}}} \sigma_j^{\mathrm{i}} = 1, \forall \mathrm{i} \in N \tag{1.19}$$

$$\sigma_j^{\mathrm{i}} \geq 0 \quad \forall j = 1, ..., m^{\mathrm{i}}, \forall \mathrm{i} \in N \tag{1.20}$$

Therefore it can be seen that if such σ exists then it is an optimal solution of nonlinear programming problems :

$$(P^{\mathrm{i}}) = \begin{cases} \min & \beta^{\mathrm{i}} - u^{\mathrm{i}}(\sigma) \\ s.t & u^{\mathrm{i}}(\sigma^{-\mathrm{i}}, s_j^{\mathrm{i}}) - \beta^{\mathrm{i}} \leq 0 \quad \forall j = 1, ..., m^{\mathrm{i}}, \forall \mathrm{i} \in N \\ & \sum_{j=1}^{m^{\mathrm{i}}} \sigma_j^{\mathrm{i}} = 1, \forall \mathrm{i} \in N \\ & \sigma_j^{\mathrm{i}} \geq 0 \quad \forall j = 1, ..., m^{\mathrm{i}}, \forall \mathrm{i} \in N \end{cases}$$

For $\mathrm{i} \in N$, with global optimal value equals to 0. Then the theorem show us how to find the Nash equilibrium strategy as the optimal solution of a single optimization problem.

Theorem 1.12 *A necessary and sufficient condition for σ^* to be a nash equilibrium of game Γ is that it is an optimal solution of the following minimization problem:*

$$(P) = \begin{cases} min & \sum_{\mathrm{i} \in N} \beta^{\mathrm{i}} - u^{\mathrm{i}}(\sigma) \\ s.t & u^{\mathrm{i}}(\sigma^{-\mathrm{i}}, s_j^{\mathrm{i}}) - \beta^{\mathrm{i}} \leq 0 \quad \forall j = 1, ..., m^{\mathrm{i}}, \forall \mathrm{i} \in N \\ & \sum_{j=1}^{m^{\mathrm{i}}} \sigma_j^{\mathrm{i}} = 1, \forall \mathrm{i} \in N \\ & \sigma_j^{\mathrm{i}} \geq 0 \quad \forall j = 1, ..., m^{\mathrm{i}}, \forall \mathrm{i} \in N \end{cases}$$

So the problem of computing a Nash equilibrium of the game Γ, reduces to solve the optimization problem (P) with optimal value zero.

Let $m = \sum_{i \in N} m^i$, we need to rank the possible strategies combinations, thefore a vector x of length $m + n$ is created as follows. Arranging the strategies of players 1 to n in order, we have a total of m strategies and we take x_i's in order as: $x_1 = \sigma_1^1, x_2 = \sigma_2^1, ..., x_{m^i} = \sigma_{m^i}^1, ..., x_m = \sigma_{m^m}^n$, where the subscripts in σ denote the strategies and superscripts stand for the players. Then take $x_{m+i} = \beta^i, i = 1, 2, ..., n$. Performing this transformation of variables in (P), the optimization problem gets converted to the following form:

$$(P^{new}) = \begin{cases} min & f(x) \\ s.t & g(x) \le 0 \\ & h(x) = 0 \\ & x_i \ge 0 \quad \forall i = 1, ...m \\ & x_i \text{ are unrestricted } \forall i = m+1, ..., m+n \end{cases}$$

Where:

$$f(x) = \sum_{i \in N} \beta^i - u^i(\sigma)$$

$$g(x) = u^i(\sigma^{-i}, s_j^i) - \beta^i \le 0 \quad \forall j = 1, ..., m^i, \forall i \in N$$

$$h(x) = \sum_{j=1}^{m^i} \sigma_j^i - 1, \forall i \in N$$

To get a solution of this nonlinear minimization problem with nonlinear constraints they use the sequential quadratic programming based quasi Newton method. The steps for the algorithms are the following:

1. Represent the game in normal form.
2. Rank the possible pure strategies combinations as desired.
3. Take varibles x_1 to x_{m+n} and form the optimization model (P^{new}) .
4. Solve the problem (P^{new}) using SQP based quasi Newton method.

By applying this formulation, we get the following bilevel problem to find the optimal strategies:

For fixed demand d:

$$(P_d) = \begin{cases} \min & \sum_{k \in G} \delta^k - u^k(\sigma) = \sum_{k \in G} (\delta^k - \sum_{s \in S} \sigma(s) \lambda q_s^k) \\ s.t & u^k(\sigma^{-k}, s_j^k) - \delta^k \le 0 \quad \forall j = 1, ..., m^k, \forall k \in G \\ & \sum_{j=1}^{m^k} \sigma_j^k = 1, \forall k \in G \\ & \sigma_j^k \ge 0 \quad \forall j = 1, ..., m^k, \forall k \in G \\ & (\lambda, q_s) \in ISO(s, d) \end{cases}$$

Where $ISO(s, \mathrm{d})$ is the dispatch program solved by the ISO when the demand value is d and the strategy profile s is played.

$$ISO(s,\mathrm{d}) = \begin{cases} min & \sum_{n\in G} c_s^n(q_s^n) \\ s.t & \sum_{\mathrm{e}\in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq q_s^n + \sum_{\mathrm{e}\in K_n} f_\mathrm{e} sgn(\mathrm{e},n), \quad n \in G \\ & \sum_{\mathrm{e}\in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq \sum_{\mathrm{e}\in K_n} f_\mathrm{e} sgn(\mathrm{e},n), \quad n \notin G \\ & q_s^n \in [0, \bar{q}^n] \\ & f \in F \end{cases}$$

Here δ^k is the optimal payoff of player k, (σ^{-k}, s_j^k) denotes the mixed strategies combination in which player k plays with his j^{th} pure strategy, that is, a mixed strategy in which the j^{th} pure strategy of the k^{th} player is assigned the probability 1. And $u^k(\sigma) = \sum_{s\in S} \sigma(s) u^k(s)$, where $u^k(s)$ is the payoff to player k at the pure strategies combination s. That is $u^k(s) = \lambda q_k$ with λ the Lagrange multiplier associated to the nodal inequalities (1.1) and (1.2).

Remarks:

- Given a pure strategy combination $s \in S = \Pi_{k\in G} S^k$, we write $s = (s_{j_1}^1, s_{j_2}^2, ..., s_{j_{|G|}}^{|G|})$ meaning that player k plays his j_k pure strategy $s_{j_k}^k$.
- Given a pure strategy combination $s \in S$, we use the notation $c_s^k(q_s^k)$ to represent the cost function for player k when playing the strategy combination $s \in S$. Here q_s^k is the quantity player k dispatch when the strategy combination s is played. Therefore q_s is a vector who has the quantities given by the dispatch program when the strategy combination $s \in S$ is played.
- In reality q_s^k also depends on the fixed demand value d, but it is not written explicitly so the notation is not to overload and because it is understood.
- Since λ is the shadow price associated to (1.1) and (1.2) then it depends on the demand d and on the profile strategy $s \in S$,i.e, $\lambda = \lambda(s, \mathrm{d})$.
- If $s = (s_{j^1}^1, s_{j^2}^2, ..., s_{j^{|G|}}^{|G|})$, then $\sigma(s) = \prod_{k\in G} \sigma_{j^k}^k$
- If we want a parametrization of the lower level problem in terms of $\sigma(s)$ instead of s, we need to define for $s \in S$, $q_{\sigma(s)}^k := q_s^k 1_{\{\sigma_s^k > 0\}} + 2\bar{q} 1_{\{\sigma_s^k = 0\}}$ so it's just q_s^k when the probability $\sigma_s^k > 0$ and $2\bar{q}$ in other case, the idea is that the system is infeasible in the later case.

But the general model uses a probablity distribution P of d, and the expected payoff is optimized.

$$(P) = \begin{cases} \min \quad \displaystyle\sum_{k \in G} \left(\delta^k - \sum_{s \in S} \int \sigma(s) \lambda(s, \mathrm{d}) q_s^k \mathrm{d}P(\mathrm{d}) \right) \\ s.t \quad u^k(\sigma^{-k}, s_j^k) - \delta^k \leq 0 \quad \forall j = 1, ..., m^k, \forall k \in G \\ \sum_{j=1}^{m^k} \sigma_j^k = 1, \forall k \in G \\ \sigma_j^k \geq 0 \quad \forall j = 1, ..., m^k, \forall k \in G \\ (\lambda, q_s) \in ISO(s, \mathrm{d}) \end{cases}$$

As we will see later, the problems with fixed d are important for numerical purpuses specially when we want to approximate an expectation.

1.5 Piecewise linear strategies

It is studied now the case when the bids are piecewise linear, in particular, when they have 2 pieces. Generators choose two slopes α and β with $\alpha < \beta$ which define the cost function.

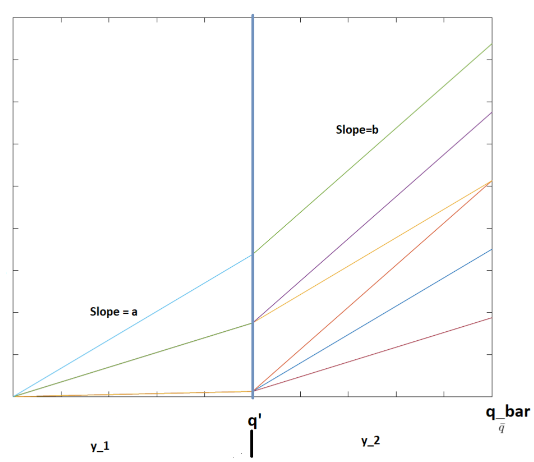
So if generator $n \in G$ chooses the slopes α and β then:

$$c_n(q_n) = \begin{cases} \alpha q_n & \text{if } 0 \leq q \leq q' \\ (q - q')\beta + \alpha q' & \text{if } q' < q \leq \bar{q} \end{cases}$$

1.5.1 Modeling Piecewise linear functions

In this subsection we see how to model a 2 pieces continous linear function in order to obtain a mixed integer programming problem.

$$c_n(q_n) = \begin{cases} \alpha q_n & \text{if } 0 \leq q \leq q' \\ (q - q')\beta + \alpha q' & \text{if } q' < q \leq \bar{q} \end{cases}$$



Just to simplify notation we get rid of the n and define two binary variables y_1 and y_2 which are going to tell us if we are in the inverval $[0, q']$ or $[q', \bar{q}]$.

- If $y_1 = 1$ then we are in $[0, q']$, and every point in that interval can be written as a convex combination of 0 and q', lets say $x \in [0, q']$ then $x = 0 \times x_1 + q'x_2 = q'x_2$ with $x_1 + x_2 = 1$.
- If $y_2 = 1$ then $x \in [q', \bar{q}] \implies x = q'x_2 + \bar{q}x_3$

Now we need to add the restrictions $y_1 + y_2 = 1$, $x_1 \leq y_1$,$x_3 \leq y_2$. which means that if we take a point $x \in [0, \bar{q}]$,$x \neq q'$ it can only be in $[0, q']$ or in $[q', \bar{q}]$, and if its the first case then $x_3 = 0$ and in the second case $x_1 = 0$. Also , since we want to write every point in $[0, \bar{q}]$ as a convex combination of 0, q' and $\bar{q}$ we can add the restriction $x_1 + x_2 + x_3 = 1$, and thanks to the other restrictions we don't need to write $x_1+x_2=1$ and $x_2+x_3=1$ separately, since x_1 and x_3 can't be greater than 0 simultaneously (SOS2). Now the function in terms of x_1, x_2 and x_3 is just $x_1c(0)+x_2c(q')+x_3c(\bar{q}) = \alpha q'x_2 + [(\bar{q}-q')\beta + \alpha q']x_3$.

Its important to notice that the SOS2 condition has to be included, otherwise we can have $x_1 = x_3 = 0.5$ then $x = \bar{q}/2$ and $c(\bar{q}/2) \neq [(\bar{q}-q')\beta + \alpha q']/2$, unless $\bar{q}/2 \leq q'$ and $[(\alpha = \beta)$ or $\bar{q} = q']$ or $\bar{q}/2 > q'$ and $[q' = 0$ or $\alpha = \beta]$. Since in our model $\alpha < \beta$, $\bar{q} > q'$ and $q' \neq 0$, none of those conditions can happen.

Then we can write c_n in terms of x^n as:

$$c_n(x^n) = \begin{cases} \alpha q' x_2^n + [(\bar{q}-q')\beta + \alpha q']x_3^n \\ s.t \quad x_1^n + x_2^n + x_3^n = 1 \\ x_1^n \leq y_1^n \\ x_3^n \leq y_2^n \\ y_1^n + y_2^n = 1 \\ y_{\mathrm{i}}^n \in \{0,1\} & \text{for i} = 1,2 \\ x_{\mathrm{i}}^n \geq 0 & \text{for i} = 1,2,3 \\ x^n = q'x_2^n + \bar{q}x_3^n \end{cases}$$

1.5.2 ISO solution for 2 pieces linear bid

Lets define the following variables and notation:

- $|G| = n$.
- For $k = 1, \ldots, n$ we'll use $\alpha^{n+k} := \beta^k$.
- We define the quantities vector $q = 0 \in \mathbb{R}^n$. So at the start everyone is dispatching 0.
- For $\mathrm{i} = 1, \ldots 2n$, we define $X^{\mathrm{i}} := \left\{ k \in \{1, \ldots, 2n\} \setminus \bigcup_{j=1}^{\mathrm{i}-1} X^j : \alpha^k = \min_{j \in \{1,\ldots,2n\} \setminus \bigcup_{j=1}^{\mathrm{i}-1} X^j} \alpha^j \right\}$
  So in the case when we have repeated strategies, some of the last X^{i} will be empty.
- $\tau = \min\{\mathrm{i} \in \{1, \ldots, 2n\} : X^{\mathrm{i}} = \emptyset\}$ This index represent worst case scenario. When we finish in the iteration τ.
- For $\mathrm{i} = 1, \ldots \tau$ we define $A^{\mathrm{i}} = \{j \in X^{\mathrm{i}} : j \leq n\}$ and $B^{\mathrm{i}} = \{j \in X^{\mathrm{i}} : j > n\}$, so $X^{\mathrm{i}} = A^{\mathrm{i}} \cup B^{\mathrm{i}}$.
- We also define $A_f^{\mathrm{i}} = \{j \in \{1, \ldots, n\} : q^j = q'\}$ and $B_f^{\mathrm{i}} = \{j \in \{n+1, \ldots, 2n\} : q^{j-n} = \bar{q}\}$
- For each $\mathrm{i} = 1, \ldots, \tau$:
  if $j \in A^{\mathrm{i}}$

1. if $B^{\mathrm{i}} = \emptyset$ and $\mathrm{d}^{\mathrm{i}} \neq 0$ then:

$$q^j = \min\left(\frac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|}, q'\right)$$

2. if $B^{\mathrm{i}} \neq \emptyset$ and $\mathrm{d}^{\mathrm{i}} \neq 0$ then:

$$\max\left(sign(\bar{q} - 2q')\left[\min\left(\frac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|}, q'\right), -\min\left(\frac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|}, \frac{\mathrm{d}^{\mathrm{i}} - (\bar{q} + q')|B^{\mathrm{i}}|}{|A^{\mathrm{i}}|}, q'\right)\right]\right)$$

3. Otherwise $q^j = 0$

if $j \in B^{\mathrm{i}}$

1. if $A^{\mathrm{i}} = \emptyset$ and $\mathrm{d}^{\mathrm{i}} \neq 0$ then:

$$q^{j-n} = \min\left(\frac{\mathrm{d}^{\mathrm{i}} + |B^{\mathrm{i}}|q'}{|X^{\mathrm{i}}|}, \bar{q}\right)$$

2. if $A^{\mathrm{i}} \neq \emptyset$ and $\mathrm{d}^{\mathrm{i}} \neq 0$ then:

$$\max\left(sign(\bar{q} - 2q')\left[\min\left(\frac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|} + q', \frac{\mathrm{d}^{\mathrm{i}} + (|B^{\mathrm{i}}| - |A^{\mathrm{i}}|)q'}{|B^{\mathrm{i}}|1_{\mathrm{d}^{\mathrm{i}} > q'|X^{\mathrm{i}}|}}, \bar{q}\right), -\min\left(\frac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|} + q', \bar{q}\right)\right]\right)$$

3. Otherwise $q^j = 0$

Where d^{i} is the residual demand defined by $\mathrm{d}^1 = \mathrm{d}$ and for $\mathrm{i} \geq 2$

$$\mathrm{d}^{\mathrm{i}} = \mathrm{d}^{\mathrm{i}-1} - \sum_{j \in A^{\mathrm{i}}} q^j - \sum_{j \in B^{\mathrm{i}}} q^{j-n}$$

- The last iteration is $\mathrm{i}^* = \min\{\mathrm{i} \in \{1, \ldots, 2n\} : \mathrm{d}^{\mathrm{i}} = 0\} - 1$
- The shadow price is $\lambda = \alpha^{\mathrm{i}^*}$, except, when d is writen as $jq' + k\bar{q}$ with $j, k = 1, \ldots, n$. In which case, $\lambda \in [\alpha^{\mathrm{i}^*}, \alpha^{\mathrm{i}^{*+1}}]$. When this happends we will consider $\lambda = \alpha^{\mathrm{i}^*}$ as the shasdow price, since de ISO wants to minimize the overall cost.

Remark: This can be seen in the case we have 2 firms with $\alpha_1 < \alpha_2$ and $\mathrm{d} < 2q'$, we can see the problems as one where firms bid linear functions.

$$ISO(a, \mathrm{d}) = \begin{cases} \min\limits_{q} & \alpha_1 q_1 + \alpha_2 q_2 \\ s.t & q_1 + q_2 \geq \mathrm{d} \\ & q_{\mathrm{i}} \leq \bar{q} \qquad \forall \mathrm{i} \in G = \{1, 2\} \\ & q_{\mathrm{i}} \geq 0 \qquad \forall \mathrm{i} \in G = \{1, 2\} \end{cases}$$

Then from the KKT conditions we get:

$$\lambda(\alpha_1, \alpha_2, \mathrm{d}) = \begin{cases} \min\{\alpha_1, \alpha_2\} & \text{if } \mathrm{d} < q' \\ [\alpha_1, \alpha_2] & \text{if } \mathrm{d} = q' \\ \max\{\alpha_1, \alpha_2\} & \text{if } 2q' > \mathrm{d} > q \end{cases}$$

So the shadow price is not unique when we fill the transmission line of all generators that use the same strategy β or when the demand is just the breaking point of all the generators that use the strategy α multiplied by the number of generators who use that strategy.

Then the solution to the dispatch program is the vector q at iteration $\mathrm{i}^* \leq \tau - 1$

It is important to notice that q is just the vector that assigns as much as possible to generators with low bids until the demand is satisfied.

Remarks:

- Notice that in the general case, we have $\dfrac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|} \leq q'$ and $\dfrac{\mathrm{d}^{\mathrm{i}}}{|X^{\mathrm{i}}|} + q' \leq \bar{q}$. So the condition is:

$$q'\left(|X^{\mathrm{i}}| + |A_f^{\mathrm{i}}|\right) + \bar{q}|B_f^{\mathrm{i}}| \leq \bar{q}\left(|X^{\mathrm{i}}| + |B_f^{\mathrm{i}}|\right) + q'\left(|A_f^{\mathrm{i}}| - |X^{\mathrm{i}}|\right)$$

  Which is equivalent to

$$2|X^{\mathrm{i}}|q' \leq |X^{\mathrm{i}}|\bar{q} \iff 2q' \leq \bar{q}$$

  So the inequality $2q' \leq \bar{q}$ is key.
- To compute the residual demand at iteration i in the general case, we use the fact that if we are computing iteration i then all generators in iteration $\mathrm{i}-1$ have quantities $\bar{q}, q'$ or 0 otherwise the demand is smaller than $q'|A_f^{\mathrm{i}}| + \bar{q}|B_f^{\mathrm{i}}|$ and we would have finished at iteration $\mathrm{i}^* \leq \mathrm{i} - 1$.

Example:

Suppose we have 3 players and $\alpha_1 < \beta_1 < \alpha_2 < \alpha_3 = \beta_2 < \beta_3$.

- Step 1: Define $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4 = \beta_1, \alpha_5 = \beta_2, \alpha_6 = \beta_3)$, $\mathrm{d}^1 = \mathrm{d}$ and $q = 0 \in \mathbb{R}^3$
- Iteration 1: $X^1 = \{1\}$, $A^1 = \{1\}$ y $B^1 = \emptyset$, so $j \in A \implies j = 1$ then $q_1 = \min(\mathrm{d}, q')$ and $\mathrm{d}^2 = \mathrm{d} - q_1$. if $\mathrm{d} \leq q'$ then $\mathrm{d}^2 = 0$ and we finish, otherwise $\mathrm{d}^2 = \mathrm{d} - q' > 0$.
- Iteration 2: $X^2 = \{4\}$, $A^2 = \emptyset$,$B^2 = \{4\}$. Then $q_{4-3} = q_1 = min(\mathrm{d} - q' + q', \bar{q})$ and $\mathrm{d}^3 = \mathrm{d}^2 - q_1 + q' = \mathrm{d} - q_1$. If $q' < \mathrm{d} \leq \bar{q}$ then $q_1 = \mathrm{d}$ and $\mathrm{d}^3 = 0$ so we finish, otherwise $\mathrm{d} > \bar{q}$, $q_1 = \bar{q}$ and $\mathrm{d}^3 = \mathrm{d} - \bar{q} > 0$.
- Iteration 3: $X^3 = \{2\}$, $A^3 = \{2\}$,$B^3 = \emptyset$. Then $q_2 = \min(\mathrm{d} - \bar{q}, q'\}$. If $\mathrm{d} \leq q' + \bar{q}$ then $q_2 = \mathrm{d} - \bar{q}$ and $\mathrm{d}^4 = \mathrm{d}^3 - q_2 = \mathrm{d} - \bar{q} - (\mathrm{d} - \bar{q}) = 0$ and we finish, otherwise $\mathrm{d} > q' + \bar{q}$, $q_2 = q'$ and $\mathrm{d}^4 = \mathrm{d} - \bar{q} - q'$
- **Iteration 4:** This is the hardest iteration, since $X^4 = \{3, 5\}$, $A^4 = \{3\}$,$B^4 = \{5\}$. We also have $\mathrm{d} > q' + \bar{q}$ otherwise we would have finished already and $\mathrm{d}^4 = \mathrm{d} - \bar{q} - q'$, so we want to give each player an extra quantity of $\mathrm{d}^4/2$ if possible, and that's the problem, since in the general case $2q' \neq \bar{q}$.

  Notice the following:

1. $\frac{d^4}{2} = \frac{d - \bar{q} - q'}{2} \leq q' \iff d \leq 3q' + \bar{q}$ and in this case $q_3 = \min(d^4/2, q') = d^4/2$ then $q_2 = \min(d^4/2 + q', \bar{q}) = d^4/2 + q'$, since $\frac{d - \bar{q} - q'}{2} + q' \leq \bar{q} \iff d \leq 3\bar{q} - q'$ and $3q' + \bar{q} \leq 3\bar{q} - q' \iff 2q' \leq \bar{q}$
2. If $q_3 = \min(d^4/2, q') = q'$ then $q_2 = \min(d^4 - q' + q', \bar{q})$. Notice that $d^4 = d - \bar{q} - q' \leq \bar{q} \iff d \leq 2\bar{q} + q'$ and $2\bar{q} + q' \leq 3\bar{q} - q' \iff 2q' \leq \bar{q}$.

- Suppose $2q' \leq \bar{q}$:
  * Case $q' + \bar{q} \leq d \leq 3q' + \bar{q}$: Then $q_2 = \frac{d - \bar{q} - q'}{2} + q'$ and $q_3 = \frac{d - \bar{q} - q'}{2}$
  * Case $3q' + \bar{q} < d \leq 2\bar{q} + q'$ then $q_3 = q'$ and $q_2 = d - \bar{q} - q'$.
  * Case $2\bar{q} + q' < d$ then $q_2 = \bar{q}$, $q_3 = q'$ and $d^5 = d - 2\bar{q} - q' > 0$
- Suppose $2q' > \bar{q}$:
  * $q' + \bar{q} \leq d \leq 3\bar{q} - q'$: Then $q_3 = \frac{d - \bar{q} - q'}{2}$ and $q_2 = q' + \frac{d - \bar{q} - q'}{2}$
  * $3\bar{q} - q' < d < q' + 2\bar{q}$: Then $q_3 = d - 2\bar{q}$ y $q_2 = \bar{q}$
  * $q' + 2\bar{q} \leq d$: Then $q_3 = q'$, $q_2 = \bar{q}$ and $d^5 = d - 2\bar{q} - q' > 0$

- Iteration 5: Now $X^5 = \{6\}$, $A^5 = \emptyset$ and $B^5 = \{6\}$. So $q_3 = \min(d - 2\bar{q}, \bar{q}) = d - 2\bar{q}$ (Since $d < 3\bar{q}$), and $d^6 = 0$
- Notice that $X^6 = \emptyset$,$\tau = 6$ and $i^* = 5$.

Theorem 1.13 *If we define q as before then q at iteration i^* is the optimal solution to the ISO problem.*

PROOF. We'll first proof that at optimality, $\sum_{j \in G} q_j = d$. Indeed, suppose $q^* \in \mathbb{R}^n$ is the optimum and $\sum_{j \in G} q_j^* > d$, define $\varepsilon = \sum_{j \in G} q_j^* - d$, $P = \{j \in G : q_j^* > 0\}$. Considerer $k \in P$, $\varepsilon < \min(q_k^*, \sum_{j \in G} q_j^* - d)$ and

$$\tilde{q}_k = \begin{cases} q_k^* - \varepsilon & \text{if } j = k \\ q_j^* & \text{if } j \neq k \end{cases}$$

Then we can see the following

- $\tilde{q}_j \in [0, \bar{q}]\ \forall j \in G$, therefore $\tilde{q} \neq q^*$ is feasible
- Every function $c_j(q_j)$ is stricly increasing in q_j. Therefore

$$\sum_{j \in G} c_j(\tilde{q}_j) = \sum_{j \in P} c_j(\tilde{q}_j) < \sum_{j \in P} c_j(q_j^*) = \sum_{j \in G} c_j(q_j^*)$$

Which contradicts the fact that q^* is the optimal solution.

So the optimum q^* is such that:

$$\sum_{j \in G} c_j(q_j^*) \leq \sum_{j \in G} c(q_j) \quad \forall q \in \mathbb{R}^n \text{ feasible} \tag{1.21}$$

$$\sum_{j \in G} q_j^* = \mathrm{d} \tag{1.22}$$

$$q_j^* \in [0, \bar{q}], \quad \forall j \in G \tag{1.23}$$

$$q^* \text{ Satisfy the equipartition property} \tag{1.24}$$

Lets call $q \in \mathbb{R}^n$ the vector of quantities given by our construction. This is the vector that assigns as much as possible to generators with low bids until the demand is satisfied (the idea is that if the quantities are assign that way, generators with the lowest bids will have values q' or $\bar{q}$ and the others will have to share the residual demand). We want to proof that $q = q^*$

Without loss of generality take $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n$ and suppose by contradiction that $q \neq q^*$. Then $\exists k_1, k_2$ such that $q_{k_1}^* < q_{k_1}$ and $q_{k_2}^* > q_{k_2}$. Indeed from $\sum_{j \in G} q_j^* = \sum_{j \in G} q_j = \mathrm{d}$ we can't have an strict inequality without having the other one too and since $q \neq q^*$ we can't have only equalities.

We can conclude the following inequalities: $q_{k_1} > 0$, $q_{k_1}^* < \bar{q}$, $q_{k_2} < \bar{q}$ and $q_{k_2}^* > 0$.

- If $q_{k_1}^* \in [q', \bar{q})$: we have two cases:
  1. if there exits $\tilde{k_1} \in G$ such that $q_{\tilde{k}_1}^* > q_{k_1}^*$ and $\beta_{\tilde{k}_1} \geq \beta_{k_1}$ then we can considerer $\tilde{q}_j = q_j^*$ if $j \neq k_1, \tilde{k}_1$. $\tilde{q}_{k_1} = q_{k_1}^* + \varepsilon$ and $\tilde{q}_{\tilde{k}_1} = q_{\tilde{k}_1}^* - \varepsilon$, then is easy to see that:

$$\begin{aligned}
\sum_{j \in G} (c(q_j^*) - c(\tilde{q}_j)) =& (q_{k_1}^* - q')\beta_{k_1} + \alpha_{k_1} q' + (q_{\tilde{k}_1}^* - q')\beta_{\tilde{k}_1} + \alpha_{\tilde{k}_1} q' \\
& - \left[(q_{k_1}^* + \varepsilon - q')\beta_{k_1} + \alpha_{k_1} q' + (q_{\tilde{k}_1}^* - \varepsilon - q')\beta_{\tilde{k}_1} + \alpha_{\tilde{k}_1} q' \right] \\
& = \varepsilon(\beta_{\tilde{k}_1} - \beta_{k_1}) > 0
\end{aligned}$$

     Where the strict inequality comes from the equity property (if $\beta_{\tilde{k}_1} = \beta_{k_1}$ then $q_{\tilde{k}_1}^* = q_{k_1}^*$). $\rightarrow\leftarrow$.
  2. $\forall j \in G : q_j^* \leq q_{k_1}^*$ or $\beta_j < \beta_{k_1}$
     - Let's take $j = k_2$,if $q_{k_2}^* \leq q_{k_1}^*$ is true then $q_{k_2} < q_{k_2}^* \leq q_{k_1}^* < q_{k_1}$. We have 2 cases, if $q_{k_2} \geq q'$ then $\beta_{k_1} < \beta_{k_2}$. Therefore $q_{k_2}^* < q_{k_1}^*$, but since $q_{k_1}^* < \bar{q}$ we can find $\varepsilon > 0$ such that $\tilde{q}$ as before is better than the optimum. Now if $q_{k_2} < q'$ then $\alpha_{k_1} < \beta_{k_1} < \alpha_{k_2}$ (because $q_{k_1} > q'$), then again we can find $\varepsilon > 0$ such that $\tilde{q}$ is better than the optimum.$\rightarrow\leftarrow$. Therefore $\beta_{k_2} < \beta_{k_1}$ must be true, since $q_{k_1} > q_{k_1}^* \geq q'$ we should have $q_{k_2} = \bar{q}$ before having $q_{k_1} > q'$, but since $\bar{q} \geq q_{k_2}^* > q_{k_2}$ we have a contradiction.
- If $q_{k_1}^* \in (0, q')$
  - Case $\alpha_{k_1} \geq \alpha_{k_2}$.

1. if $q_{k_2} < q'$ then from the initial inequalities $q^*_{k_2} > q_{k_2} \geq q_{k_1} > q^*_{k_1}$ and the residual demand is 0 after all generators with slope equal to α_{k_2} are dispatched. So if $\alpha_{k_1} \neq \alpha_{k_2}$ $q_{k_1} = 0 > q^*_{k_1} > 0 \rightarrow\leftarrow$. Then $\alpha_{k_1} = \alpha_{k_2} < \beta_{k_2}$ and $q_{k_1} = q_{k_2}$ now if $q^*_{k_2} \leq q'$ then $q^*_{k_2} = q^*_{k_1}$ but from the inequality $q^*_{k_2} > q^*_{k_1}$, therefore $q^*_{k_2} > q'$ but since $q' > q_{k_2} = q_{k_1} > q^*_{k_1}$ we can take $\varepsilon > 0$ such that $\tilde{q}$ as before is better than the optimum.
2. if $q_{k_2} = q'$ then $\beta_{k_2} \geq \alpha_{k_1}$ otherwise $q_{k_2} > q'$ or $q_{k_1} = 0$, also if $\beta_{k_2} > \alpha_{k_1}$ then $\tilde{q}$ is better than the optimum, so we just need to see the case $\beta_{k_2} = \alpha_{k_1}$ but since $q_{k_2} = q'$ this means that the demand was satisfied before using the slopes β_{k_2} and α_{k_1} therefore $q_{k_1} = 0$ which is a contradiction.
3. if $q_{k_2} > q'$ we have 3 cases, if $\beta_{k_2} < \alpha_{k_1}$ then since $q_{k_2} < \bar{q}$ the residual demand is finished when q_{k_2} uses the slope β_{k_2}, therefore $q_{k_1} = 0$ which is a contradiction. If $\beta_{k_2} > \alpha_{k_1}$ then $q_{k_1} = q'$ so $q^*_{k_1} < q_{k_1} = q'$ then exists $\varepsilon > 0$ such that $\tilde{q}$ is better than the optimum. Finally in the case $\beta_{k_2} = \alpha_{k_1}$, since $q_{k_2} < \bar{q}$ the residual demand is fully dispatched in this iteration, if $q_{k_2} - q' = q_{k_1}$ then q^* doesn't satisfy the equipartition property, therefore we need to see the following cases: if $\bar{q} > 2q'$ then $q_{k_1} = q'$ and from the inequality $q^*_{k_1} < q'$ so $\tilde{q}$ is better than the optimum. The last case is $2q' > \bar{q}$ but this can't happend since it would mean that $q_{k_2} = \bar{q} < q^*_{k_2} \leq \bar{q}$.

– Case $\alpha_{k_1} < \alpha_{k_2}$

1. Since $0 < q^*_{k_1} < q'$ we should have $q^*_{k_2} = 0$ otherwise there exists $\varepsilon > 0$ such that $\tilde{q}$ is better than the optimum, but from the initial inequality this would mean $q_{k_2} < 0$ which is a contradiction.

Therefore $q = q^*$. □

It's not hard to see that the same remains true if the maximum capacity depends on the generator that we are considering,i.e, we have $\bar{q}_n$ for $n \in G$ instead of $\bar{q}$ $\forall n \in G$. The same idea can be extended to the case where we have a piecewise linear function with more than 2 pieces. The proof becomes more cumbersome because it increases the amount of combinations to verify. Also in both cases, we are not be able to writte the q^{i} values explicitly in each iteration i, as we can in the simplified case.

1.6 Procedure

Here we discuss the general procedure to compute the expected payoffs at equilibrium. We compute the expected payoff by taking the average value of de payoffs for each demand d in the discretization D which converges to the real expectation by the *law of large numbers*.

The following pseudocode gives the idea of the algorithms that are programmed for the different cases in type of strategies or number of players

Result: Expected Payoffs at Nash Equilibrium in mixed strategies
initialization;
Input: Number of players $|G|$ and Maximum capacity value $\bar{q}_k$ for each player $k \in G$;
Step 1: Discretize the demand distribution D;
Step 2: Define the strategies vectors S_k of each player $k \in G$;
for $\mathrm{d} \in D$ **do**
  for *each strategy combination* **do**
    1. Assign the optimal amounts q_k to each player $k \in G$ and calculate the shadow price λ using the ISO solution if we have it or by computing it using an algorithm ;
    2. Compute each player payoff $U_k^{\mathrm{d}}(s)$ for that strategy combination s
  end
end
for *each player* $k \in G$ **do**
  1. Compute the Expected payoff matrix:

$$\bar{U}_k = \frac{1}{|D|} \sum_{\mathrm{d} \in D} U_k^{\mathrm{d}}$$

end
Step 3: Compute Nash equilibrium σ^*;
for *each player* $k \in G$ **do**
  1. Compute the expected payoff $E_k = \sum_{s \in S} \sigma^*(s) \bar{U}_k(s)$
end
Step 5: Return Expected Payoffs at equilibrium in mixed strategies;

Algorithm 1: Expected Payoffs Algorithm

There is basically two important steps, first compute the expected payoff matrix and second compute the Nash equilibrium given those matrices.

1.7 Numerical Results For Piecewise Linear bids

1.7.1 2 pieces Linear function

We consider 2 parameters, $\alpha, \beta \in [0, 2]$ such that $\alpha < \beta$ this defines a piecewise linear function $c(q) = \alpha q$ if $q \leq q'$ and $c(q) = \beta(x - q') + \alpha q'$ otherwise.

We have 2 versions for this function, the first one using the dispatchProg function from section 1 and the second one using the algorithm previously described

N is how many points are we going to use for the discretization and to have more intuition about the results obtained, we consider the case symmetric.

We use the following slopes: The coordinate i from α is $\alpha_{\mathrm{i}} = 0.05 + \frac{2 - 0.05}{N}\mathrm{i}$, we also use this discretization for β. So we created a matrix of all feasible strategies x, in the first column it has the α coeficient and in the second column β coeficient such that $\alpha < \beta$. Since is the symmetric case, player every other player has the same strategy space.

Now that we have the payoff matrices, we compute the nash equilibriums using Lemke-Howson algorithm for the 2 players case and when we have 3 or more players, the code is analogous, except that we use Bapti [10] algorithm to find the Nash equilibrium.

1.7.2 Changing discretization length

Discretization length	Player 1's Expected Payoff	Player 2's Expected Payoff
6	0.1733	0.1733
10	0.1458	0.1458
14	0.1366	0.1366
18	0.1414	0.1414

Table 1.1: Expected Payoffs for different discretizations length

The difference between the expected payoffs for $N = 10$ and 14 with respect to $N = 18$ is $\approx 3\%$.

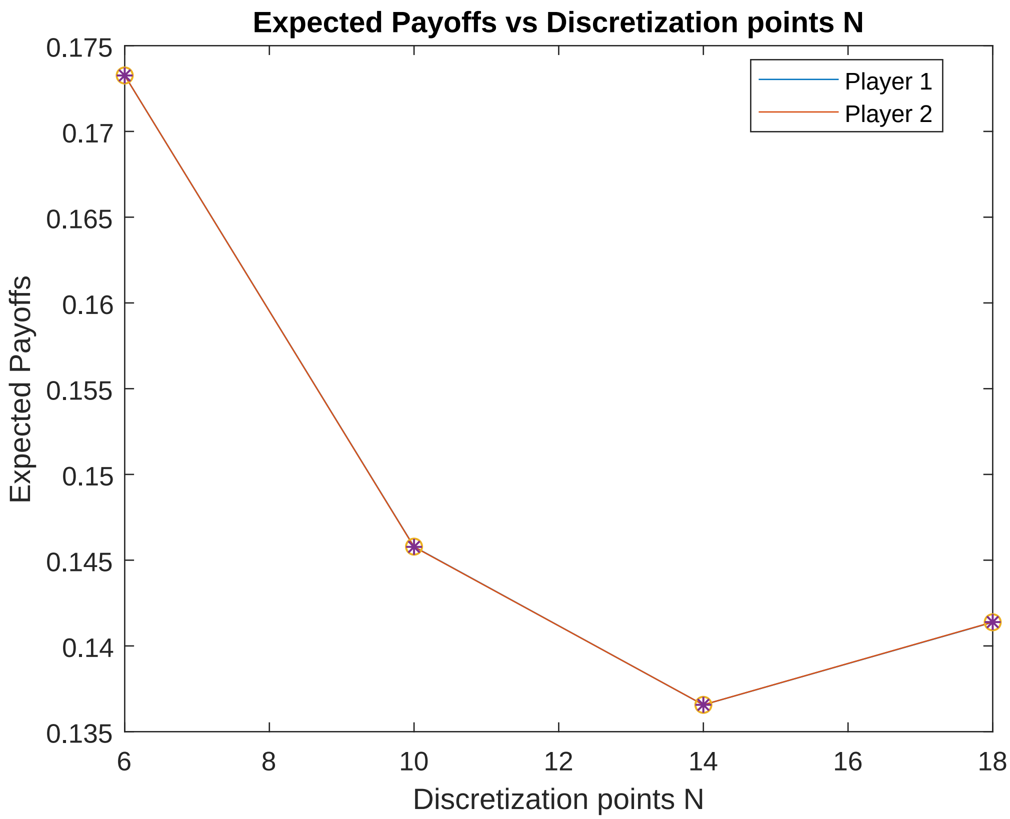


Figure 1.3: Expected payoffs vs N

1.7.3 Changing The PriceCap

Price-Cap value	Player 1's Expected Payoff	Player 2's Expected Payoff
1	0.1120	0.1110
1.1	0.1174	0.1193
1.2	0.1278	0.1278
1.3	0.1331	0.1236
1.4	0.1303	0.1275
1.5	0.1354	0.1349
1.6	0.1384	0.1376
1.7	0.1394	0.1394
1.8	0.1458	0.1458
1.9	0.1458	0.1458
2	0.1458	0.1458
3	0.1458	0.1458

Table 1.2: Expected Payoffs vs PriceCap

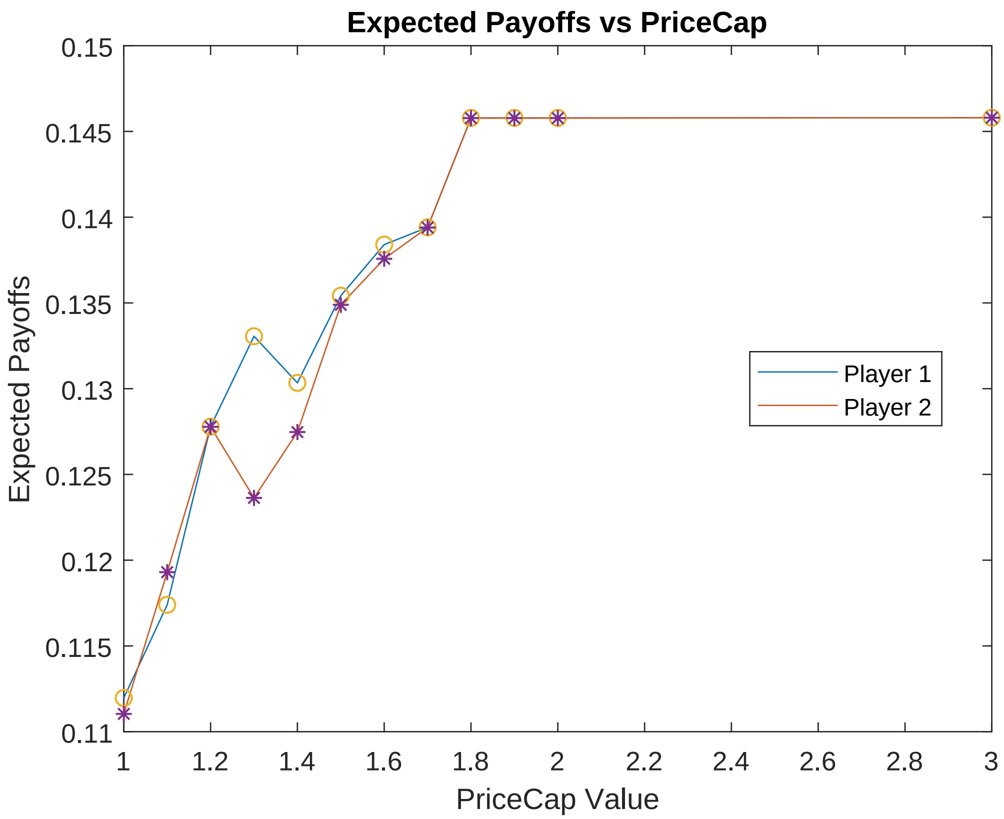


Figure 1.4: Expected Payoffs vs PriceCap

Since the maximum price given by λ is 2, having a PriceCap higher than that doesn't change the expected payoffs.

1.7.4 Changing $\bar{q}$

We fixed d = 0.5 and choose values of $\bar{q}$ such that d $< 2\bar{q}$.

$\bar{q}$ value	Player 1's Expected Payoff	Player 2's Expected Payoff
0.3	0.1974	0.1974
0.35	0.1159	0.1159
0.4	0.0890	0.0890
0.45	0.0635	0.0635
0.5	0.0631	0.0634
1	0.0634	0.0634

Table 1.3: Expected Payoffs vs $\bar{q}$

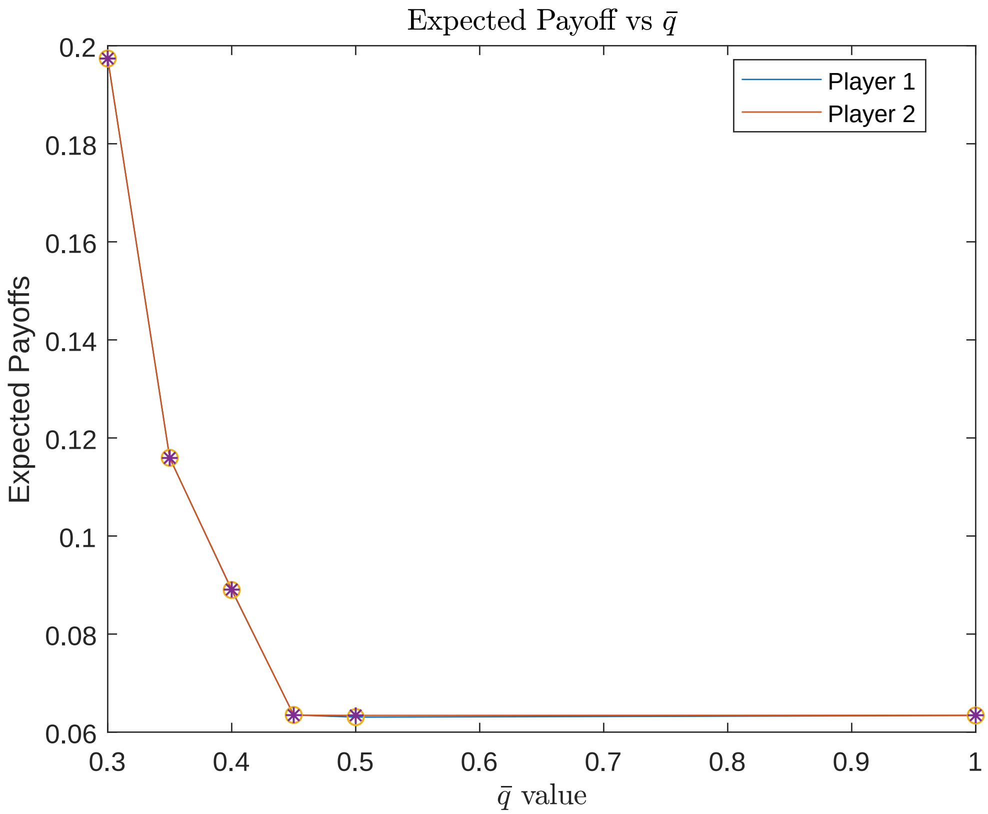


Figure 1.5: Expected payoffs for different values of $\bar{q}$

This makes sense given that decreasing the value of $\bar{q}$ makes the competition stronger and for $\bar{q} >$ d. i.e, $\bar{q} > 0.5$ nothing chances since both generators can dispatch the entire demand for every value $\bar{q} >$ d.

1.7.5 Sensitivity Analisys

Let's fix d $= 0.7$, $\bar{q} = 0.5$ and $q' = 0.25$. We'll first make a perturbation $\pm\varepsilon$ to the capacity $\bar{q}$.

The following table correspond to a perturbation in the capacity $\bar{q}$ of $+\varepsilon = 0.05$ which is 0.1% of the capacity value.

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.5022	0.4804	4.3409	0.868261	1.099544
7	0.4171	0.4157	0.3357	1.368339	1.837789
8	0.3793	0.3661	3.4801	3.464493	3.482522
9	0.3966	0.3358	15.3303	6.735987	6.545159
10	0.3925	0.3676	6.3439	12.727476	12.418359
11	0.4019	0.3564	11.3212	21.866019	22.039640
12	0.3988	0.3512	11.9358	38.157141	37.754282

Table 1.4: Perturbation $+\varepsilon = 0.05$ corresponding to 10% of $\bar{q}$

As we can see, we get payoffs from 0.3% to 15% lower just by increasing 10% the capacity, which means more competition between firms. While computation time is pretty much the same in all cases.

Now if we decrease $\bar{q}$ in $\varepsilon = 0.05$, i.e, in 10%

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.5022	0.5062	0.7902	0.868261	0.865619
7	0.4171	0.5010	16.7465	1.368339	1.848609
8	0.3793	0.4871	18.5765	3.464493	3.460056
9	0.3966	0.4874	18.6239	6.735987	6.543319
10	0.3925	0.4878	19.5365	12.727476	12.610643
11	0.4019	0.4896	17.9133	21.866019	21.837785
12	0.3988	0.4948	19.4087	38.157141	37.797952

Table 1.5: Perturbation $-\varepsilon = 0.05$ corresponding to 10% of $\bar{q}$

Payoffs gets between 0.8% to 20% higher since in this case we have less competition between firms.

If we make smaller perturbations 1% we still get big differences between payoffs. As we can see from the following 2 tables:

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.5022	0.5104	1.6008	0.868261	1.0134
7	0.4171	0.4159	0.2907	1.368339	1.1154
8	0.3793	0.3773	0.5149	3.464493	3.1372
9	0.3966	0.3977	0.2792	6.735987	6.3802
10	0.3925	0.3878	1.1970	12.727476	12.3696
11	0.4019	0.3912	2.6677	21.866019	22.0075
12	0.3988	0.3901	2.1847	38.157141	38.2518

Table 1.6: Perturbation $+\varepsilon = 0.005$ corresponding to 1% of $\bar{q}$

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.5022	0.4424	11.9097	0.868261	1.0686
7	0.4171	0.4615	9.6230	1.368339	1.4922
8	0.3793	0.4125	8.0413	3.464493	3.1396
9	0.3966	0.4216	5.9315	6.735987	6.2966
10	0.3925	0.4124	4.8258	12.727476	12.4007
11	0.4019	0.4106	2.1211	21.866019	22.0905
12	0.3988	0.4138	3.6275	38.157141	38.3744

Table 1.7: Perturbation $-\varepsilon = 0.005$ corresponding to 1% of $\bar{q}$

Now with even smaller perturbations 0.1% we get the following results:

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.5022	0.5025	0.0537	0.868261	0.9095
7	0.4171	0.4169	0.0458	1.368339	1.5043
8	0.3793	0.3808	0.3865	3.464493	3.1561
9	0.3966	0.3995	0.7262	6.735987	6.3002
10	0.3925	0.3918	0.1774	12.727476	12.4707
11	0.4019	0.4008	0.2651	21.866019	22.1682
12	0.3988	0.3976	0.2998	38.157141	38.5413

Table 1.8: Perturbation $\varepsilon = 0.005$ corresponding to 0.1% of $\bar{q}$

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.5022	0.4429	11.8060	0.868261	1.0604
7	0.4171	0.4613	9.5820	1.368339	1.4799
8	0.3793	0.4007	5.3448	3.464493	3.1710
9	0.3966	0.4179	5.1080	6.735987	6.3378
10	0.3925	0.4064	3.4147	12.727476	12.4600
11	0.4019	0.4036	0.4189	21.866019	22.3011
12	0.3988	0.4079	2.2242	38.157141	38.6884

Table 1.9: Perturbation $-\varepsilon = 0.005$ corresponding to 0.1% of $\bar{q}$

We can see that even for small values of ε making a small perturbation of -0.1% can make a significant difference between payoffs for a small value in the discretization length. While for perturbations of $+1\%$ or $+0.1\%$ the differences are small in all cases.

If we use a demand vector d = [0.05 0.1 0.15 0.2 0.25 0.75 0.8 0.85 0.9 0.95] instead of a fixed demand, we get the following:

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3837	0.0954	0.9048	0.9204
7	0.3808	0.3803	0.1305	1.3507	1.3769
8	0.3771	0.3767	0.1156	3.3409	3.3596
9	0.3762	0.3757	0.1310	6.7234	6.6437
10	0.3763	0.3759	0.1127	12.9409	12.9748
11	0.3773	0.3769	0.1076	23.0330	22.9176
12	0.3659	0.3654	0.1232	39.8119	39.8590

Table 1.10: Perturbation $\varepsilon = 0.005$ corresponding to 0.1% of $\bar{q}$

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3938	2.4516	0.9048	0.8018
7	0.3808	0.3904	2.4681	1.3507	1.6434
8	0.3771	0.3882	2.8607	3.3409	3.4097
9	0.3762	0.3816	1.4128	6.7234	6.6732
10	0.3763	0.3884	3.1261	12.9409	12.9237
11	0.3773	0.3811	0.9845	23.0330	22.9281
12	0.3659	0.3790	3.4574	39.8119	39.6501

Table 1.11: Perturbation $-\varepsilon = 0.005$ corresponding to 0.1% of $\bar{q}$

Notice that computation time is just a little bit higher, since we need to compute the best ISO solution for all the demands, but thanks to the fast algorithm given in the previous sections, this is done in less than 1 second.

The case when we increase the capacity value in ε we get a difference in payoffs similar to the percentaje given by that increase. While in the case when we lower the capacity value, we get difference in payoffs from 1% to 3.5% which is in any case better than for a fixed demand value.

The same happens when we make a perturbation $\pm\varepsilon$ of 1% of the capacity value $\bar{q}$

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3808	0.8630	0.9048	0.8360
7	0.3808	0.3754	1.4287	1.3507	1.6527
8	0.3771	0.3725	1.2218	3.3409	3.3677
9	0.3762	0.3717	1.1948	6.7234	6.6583
10	0.3763	0.3719	1.1718	12.9409	13.0408
11	0.3773	0.3730	1.1390	23.0330	23.1689
12	0.3659	0.3608	1.3849	39.8119	39.6372

Table 1.12: Perturbation $+\varepsilon = 0.05$ corresponding to 1% of $\bar{q}$

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3982	3.5400	0.9048	0.7607
7	0.3808	0.3955	3.7085	1.3507	1.6209
8	0.3771	0.3926	3.9559	3.3409	3.3477
9	0.3762	0.3866	2.7022	6.7234	6.6580
10	0.3763	0.3928	4.1954	12.9409	13.1059
11	0.3773	0.3857	2.1706	23.0330	23.1303
12	0.3659	0.3847	4.8748	39.8119	39.9530

Table 1.13: Perturbation $-\varepsilon = 0.05$ corresponding to 1% of $\bar{q}$

Having a demand vector, makes the problem less sensitive to perturbations on the capacity values.

Now we'll make perturbations to the bids.

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3852	0.2943	0.9065	0.8532
7	0.3808	0.3817	0.2366	1.3508	1.6187
8	0.3771	0.3783	0.3077	3.3534	3.3458
9	0.3762	0.3771	0.2405	6.7116	6.8102
10	0.3763	0.3772	0.2322	12.9257	13.0110
11	0.3773	0.3782	0.2364	22.8843	22.9885
12	0.3659	0.3667	0.2316	39.5118	39.7667

Table 1.14: Perturbation of $+\varepsilon = 0.005$ to all the slopes

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3829	0.3841	0.9065	1.1405
7	0.3808	0.3799	0.3808	1.3508	1.6306
8	0.3771	0.3760	0.3771	3.3534	3.3692
9	0.3762	0.3753	0.3762	6.7116	6.7686
10	0.3763	0.3754	0.3763	12.9257	12.9667
11	0.3773	0.3764	0.3773	22.8843	23.1508
12	0.3659	0.3651	0.3659	39.5118	39.4117

Table 1.15: Perturbation of $-\varepsilon = 0.005$ to all the slopes

We can see that the problem is not that sensitive to small perturbations.

Let's see what happens with a higher perturbation:

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3953	2.8296	0.9065	0.8382
7	0.3808	0.3900	2.3481	1.3508	1.6199
8	0.3771	0.3867	2.4815	3.3534	3.3925
9	0.3762	0.3856	2.4368	6.7116	6.6566
10	0.3763	0.3856	2.4234	12.9257	12.9103
11	0.3773	0.3867	2.4220	22.8843	22.8978
12	0.3659	0.3757	2.5979	39.5118	39.4666

Table 1.16: Perturbation of $+\varepsilon = 0.05$ to all the slopes

N	Payoff without perturbation	Payoff with perturbation	Difference %	Computation Time without perturbation [s]	Computation Time [s]
6	0.3841	0.3762	2.0654	0.9065	1.1364
7	0.3808	0.3712	2.5101	1.3508	1.3578
8	0.3771	0.3643	3.3826	3.3534	3.3866
9	0.3762	0.3665	2.5795	6.7116	6.6601
10	0.3763	0.3671	2.4500	12.9257	13.1507
11	0.3773	0.3680	2.4742	22.8843	22.7894
12	0.3659	0.3610	1.3472	39.5118	39.8509

Table 1.17: Perturbation of $-\varepsilon = 0.05$ to all the slopes

The difference is still small, a perturbation 10 times higher than before gives a difference in % smaller than 10 times the previous difference of 0.3%.

Adding more players

We considerer a discretization of length $N = 7$.

Number of players	**Expected Payoffs**
2	0.1733
3	0.0657
4	0.0173
5	0.0103

Table 1.18: Expected Payoffs vs Number of players in a symmetrical equilibrium

In all cases the expected payments are the same for all players It should be noted that payments decline rapidly. Notice that since the function changes depending on $\frac{\bar{q}}{2} = 0.25$ and d $< 1 = 2\bar{q}$ we have for $N \geq 4$ that all players use only the first slope α and for $N \leq 3$ only for d ≥ 0.75 we have residual demand available, so the competition becomes stronger.

It should be noted that the biggest problem in increasing the number of players is that the matrices increase greatly in size so that they can not be solved for fine discretizations due to RAM memory problems.

1.8 Quadratic strategies

Quadratic strategies has been studied in papers such as [4] and [5]. Where they characterize the set of Nash equilibria depending on different values of the demand and the best response of a producer that is the optimal bid(s) maximizing his profit. However, there are some fundamental differences between their model and the one considered here, in theirs de demand is fixed (not stochastic) and known by the players and ISO and the price is the bid, not the shadow price. These differences allow them to make a characterization of the set of best responses, while in our model you cannot follow the same steps to generalize it.

Assume that the generators chooses two parameter parameters α_n and β_n which define a quadratic function function $c_n(q_n) = \alpha_n q_n + \beta_n q_n^2$. The parameters (α, β) are considered in $[0,1]^2$. Then for fixed d the dispatch problem is:

$$ISO(c, \mathrm{d}) = \begin{cases} \min & \sum_{n\in G} \alpha_n q_n + \beta_n q_n^2 \\ s.t & \sum_{\mathrm{e}\in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq q_n + \sum_{\mathrm{e}\in K_n} f_\mathrm{e} sgn(\mathrm{e}, n), \quad n \in G \\ & \sum_{\mathrm{e}\in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq \sum_{\mathrm{e}\in K_n} f_\mathrm{e} sgn(\mathrm{e}, n), \quad n \notin G \\ & q_n \in [0, \bar{q}_n] \\ & f \in F \end{cases}$$

We will first consider the simplified case where there are not resistance losses and there are only 2 players. Then the dispatch problem is as follows:

$$min\{\alpha_1 q_1 + \beta_1 q_1^2 + \alpha_3 q_3 + \beta_3 q_3^2 : q_1 + q_3 \geq \mathrm{d}, q_\mathrm{i} \in [0, \bar{q}]\}$$

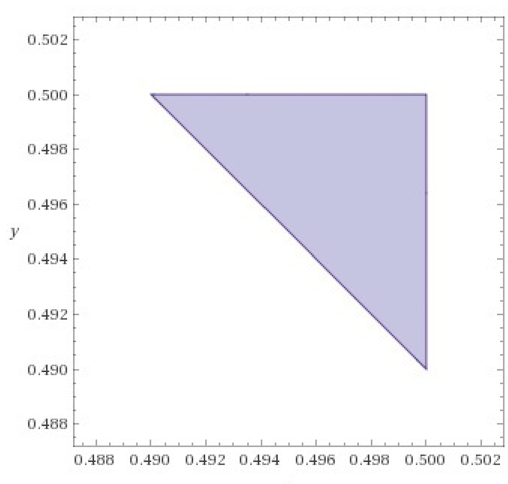


Figure 1.6: Feasible set for $\mathrm{d} = 0.99$ and $\bar{q} = 0.5$, The x coordinate represents q_1 and the y coordinate q_3

Note first that $f(q_1, q_3) = c_1(q_1) + c_3(q_3) = \alpha_1 q_1 + \beta_1 q_1^2 + \alpha_3 q_3 + \beta_3 q_3^2$ is continuous and

Define the following functions:

$$g_1(q_1, q_3) = \mathrm{d} - q_1 - q_3$$

$$g_2(q_1, q_3) = -q_1$$

$$g_3(q_1, q_3) = -q_3$$

$$g_4(q_1, q_3) = q_1 - \bar{q}$$

$$g_5(q_1, q_3) = q_3 - \bar{q}$$

It is clear that every $g_{\mathrm{i}}(q_1, q_3)$ $\mathrm{i} \in \{1, ..., 5\}$ is continuous and the constrains can be written as $g_{\mathrm{i}}^{-1}((-\infty, 0])$ therefore $F = \bigcap_{\mathrm{i} \in \{1,...,5\}} g_{\mathrm{i}}^{-1}((-\infty, 0])$ is a closed set in $\mathbb{R}^n$. On the other hand is clear that F is bounded, therefore F is compact in $\mathbb{R}^n$.

Notice that we have a convex problem and $\left(\frac{\bar{q}}{2}, \frac{\bar{q}}{2}\right)$ is in the feasible set and $g_{\mathrm{i}}\left(\frac{\bar{q}}{2}, \frac{\bar{q}}{2}\right) < 0$ for all $\mathrm{i} \in \{1, ..., 5\}$. Then the Slater condition is fulfilled.

Therefore there exists a minimum in the feasible set. An analytical solution can be found using KKT.[9]

$$min\{\alpha_1 q_1 + \beta_1 q_1^2 + \alpha_3 q_3 + \beta_3 q_3^2, q_1 + q_3 \geq \mathrm{d}, q_{\mathrm{i}} \in [0, \bar{q}]\} \tag{1.25}$$

$$L(q, \mu) = \alpha_1 q_1 + \beta_1 q_1^2 + \alpha_3 q_3 + \beta_3 q_3^2 + \mu_1(\mathrm{d} - q_1 - q_3) - \mu_2 q_1 - \mu_3 q_3 - \mu_4(\bar{q} - q_1) - \mu_5(\bar{q} - q_3) \tag{1.26}$$

$$\frac{\partial L}{\partial q_1} = \alpha_1 + 2q_1\beta_1 - \mu_1 - \mu_2 + \mu_4 = 0 \tag{1.27}$$

$$\frac{\partial L}{\partial q_3} = \alpha_3 + 2q_3\beta_3 - \mu_1 - \mu_3 + \mu_5 = 0 \tag{1.28}$$

$$\mu_1(\mathrm{d} - q_1 - q_3) = 0 \tag{1.29}$$

$$\mu_2 q_1 = 0 \tag{1.30}$$

$$\mu_3 q_3 = 0 \tag{1.31}$$

$$\mu_4(q_1 - \bar{q}) = 0 \tag{1.32}$$

$$\mu_5(q_3 - \bar{q}) = 0 \tag{1.33}$$

$$q_1 + q_3 \geq \mathrm{d} \tag{1.34}$$

$$\mu_{\mathrm{i}}, q_j, \bar{q} \geq 0, \quad \mathrm{i} \in \{1, ..., 5\}, j \in \{1, 2\} \tag{1.35}$$

1. If $q_1 + q_3 \neq \mathrm{d}$ then $\mu_1 = 0$ and there is 2 cases:
   - 1.1) If $q_1 = 0$ then $q_3 > 0$, so equation (1.31) implies that $\mu_3 = 0$ then equation (1.28) implies $\mu_5 = -\alpha_3 - 2q_3\beta_3 < 0$ which can't happend. The case $q_3 = 0$ is analogous.

1.2) If $q_1 \neq 0$ then $\mu_2 = 0$. Replacing this value in the equation (1.27) leads to $\mu_4 = -(\alpha_1 + 2q_1\beta_1) < 0$ which can't happend, and the same goes to the case when 3

2. If $q_1 + q_3 = \text{d}$.

   2.1) If $q_1 < \bar{q} \implies \mu_4 = 0$. We have 2 cases:

      2.1.1) $q_3 < \bar{q}$, then $\mu_5 = 0$. Since $q_1 + q_3 = \text{d}$ necessarily one is strictly greater than 0, suppose $q_1 > 0$ then $\mu_2 = 0$. Now if $q_3 = 0$ we have directly that $q_1 = \text{d}$, if not $q_3 > 0$ and $\mu_3 = 0$ so we get the system of equations: $\alpha_1 + 2q_1\beta_1 = \alpha_3 + 2q_3\beta_3$ and $q_1 + q_3 = \text{d}$. Whose solution is: $(q_1, q_3) = (\frac{\alpha_3 - \alpha_1 + 2\text{d}\beta_3}{2(\beta_3 + \beta_1)}, \frac{\alpha_1 - \alpha_3 + 2\text{d}\beta_1}{2(\beta_3 + \beta_1)})$ and whose multiplier value is $\mu_1 = \alpha_1 + 2_1\beta_1$ .Now if $q_1 = 0$ then $q_3 = \text{d}$.

      2.1.2) If $q_3 = \bar{q}$, then $q_1 = \text{d} - \bar{q}$.

   2.2) If $q_1 = \bar{q}$, then $\mu_2 = 0$, $q_3 = \text{d} - \bar{q}$ y $\mu_1 = \alpha_1 + 2_1\beta_1$

In summary:

Generator 1's amount	**Generator 2's amount**	**Multiplier**
$\bar{q}$	$\text{d} - \bar{q}$	$\alpha_3 + 2(\text{d} - \bar{q})\beta_3$
$\text{d} - \bar{q}$	$\bar{q}$	$\alpha_1 + 2(\text{d} - \bar{q})\beta_1$
0	d	$\alpha_3 + 2\text{d}\beta_3$
d	0	$\alpha_1 + 2\text{d}\beta_1$
$\frac{\alpha_3 - \alpha_1 + 2\text{d}\beta_3}{2(\beta_3 + \beta_1)}$	$\frac{\alpha_1 - \alpha_3 + 2\text{d}\beta_1}{2(\beta_3 + \beta_1)}$	$\frac{\alpha_3\beta_1 + \alpha_1\beta_3 + 2\text{d}\beta_1\beta_3}{(\beta_1 + \beta_3)}$

Table 1.19: Local Minimum Points and Multiplier Value μ_1

Where its calculation indicates for what cases it works, for example q_1 can not be d if d is greater than $\bar{q}$.

Remarks:

- When $\alpha_1 = \alpha_3$ and $\beta_1 = \beta_3$ then $\frac{\alpha_3 - \alpha_1 + 2\text{d}\beta_3}{2(\beta_3 + \beta_1)} = \frac{\alpha_1 - \alpha_3 + 2\text{d}\beta_1}{2(\beta_3 + \beta_1)} = \frac{\text{d}}{2}$ which is one of the intuitive solutions to the problem.
- For the uniqueness of multiplier μ_1 in the case in which any of the amounts is $\bar{q}$ it is required that $2\bar{q} > \text{d}$, condition that was already part of the model. In the case that any of the amounts is 0 it is required that $\bar{q} \neq \text{d}$, this condition was not part of the model and in the simulations values of $\bar{q}$ and d are used so that this is fulfilled.

1.9 Numerical Results for Quadratic bids

Using a distribution of demands d = $[0.05, 0.1, 0.15, 0.2, 0.25, 0.75, 0.8, 0.85, 0.9, 0.95]$ and $\bar{q} = 0.5$ and. We obtain the following:

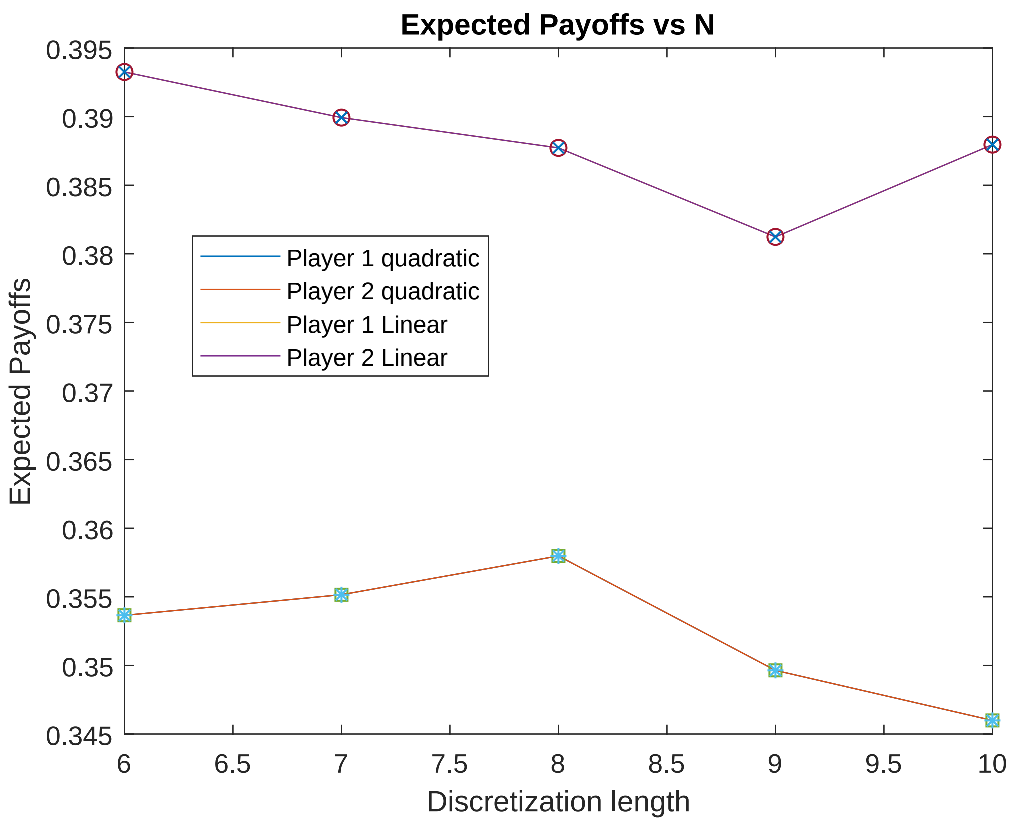


Figure 1.7: Quadratic and 2 pieces Linear comparison

When comparing both models with 2 players we noticed that the results are similar. The difference is ≈ 0.03 which corresponds to an 8%

Let's see what is happening for $N = 6$. In this case, for the linear problem in 2 pieces we have the following:

α	β	**weight**
0.0500	0.9500	0.1982
0.0500	1.1000	0.1609
0.0500	1.2500	0.2840
0.0500	1.4000	0.3566

Table 1.20: Strategies with their respective probabilities in Equilibrium

Recall that the vector of demands that we are using is

$$\mathrm{d} = [0.05, 0.1, 0.15, 0.2, 0.25, 0.75, 0.8, 0.85, 0.9, 0.95]$$

And $q' = 0.25$, then, when $\mathrm{d} \leq 0.25 = q'$ the firm with the lowest bid will be the dispatched first, with what it makes sense that for α you choose the lowest value you can take. Now for $q' + \bar{q} < \mathrm{d}$, both firms will dispatch using the second piece of their function regardless of strategy, for $q' < \mathrm{d} \leq 2q'$ both firms dispatch using the first piece and for $\bar{q} = 2q' < \mathrm{d} \leq q' + \bar{q}$ a firm dispatches using the second piece, while the other uses the first, basically comparing the values $f_1(\mathrm{d} - q') + f_2(q'), f_1(q') + f_2(\mathrm{d} - q'), f_1(\bar{q}) + f_2(\mathrm{d} - \bar{q})$ y $f_1(\mathrm{d} - \bar{q}) + f_2(\bar{q})$. So it makes sense that for β you do not choose the lowest value you can take, because for the first cases of demand, does not influence the β and for those who follow, $\mathrm{d} \geq 0.75$ always both firms are dispatched using the second slope.

For each $\mathrm{d} \in D$ we have an amount Q_{d}, so we define the average amount as $\bar{Q} = \sum_{\mathrm{d} \in D} Q_{\mathrm{d}}$. Below we show the average quantities obtained for the i, j coordinates of strategies with positive probability in the mixed equilibrium.

$Q_1 =$

0.2500	0.2875	0.2875	0.2875
0.2125	0.2500	0.2875	0.2875
0.2125	0.2125	0.2500	0.2875
0.2125	0.2125	0.2125	0.2500

$Q_2 =$

0.2500	0.2125	0.2125	0.2125
0.2875	0.2500	0.2125	0.2125
0.2875	0.2875	0.2500	0.2125
0.2875	0.2875	0.2875	0.2500

Table 1.21: Average amounts for non-zero weight strategies

$U_1 =$

0.1929	0.3215	0.4190	0.5165
0.2300	0.2757	0.4190	0.5165
0.2983	0.2983	0.3586	0.5165
0.3665	0.3665	0.3665	0.4415

$U_2 =$

0.1929	0.2300	0.2983	0.3665
0.3215	0.2757	0.2983	0.3665
0.4190	0.4190	0.3586	0.3665
0.5165	0.5165	0.5165	0.4415

Table 1.22: Expected Payoffs

On the diagonal when both have the same strategy the amounts are $\frac{\bar{\mathrm{d}}}{2}$ what makes sense and when $j < \mathrm{i}$ the average amount is the residual, because the other firm has a lower price.

Now when we use the uniform demand that was used in the previous simulations:

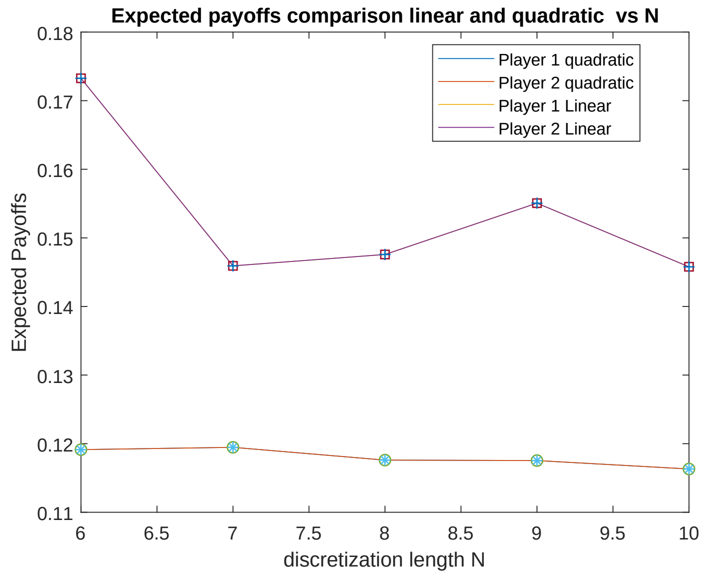


Figure 1.8: Quadratic and 2 parts Linear comparison with uniform demand

The difference is $\approx 15\%$.

We can also consider α, β and $\gamma \in [0, 2]$ and that the function changes the slope in the points $\frac{\bar{q}}{3}$ and $\frac{2\bar{q}}{3}$. So We can compare with a 3 pieces linear function.

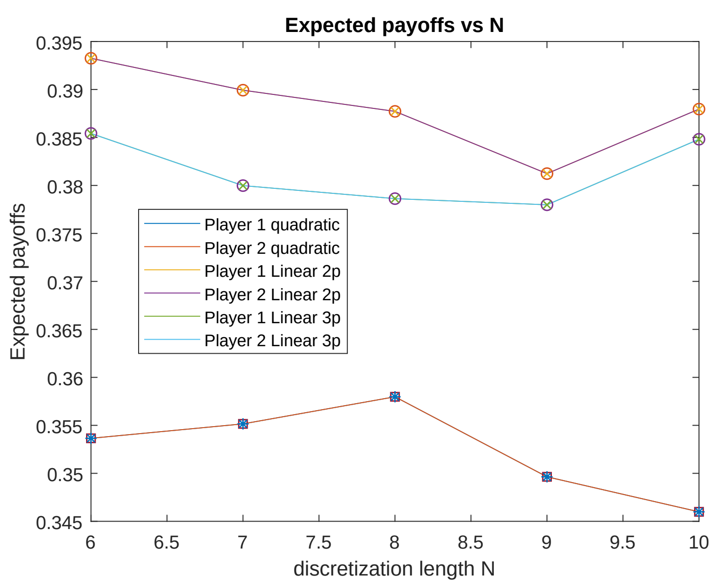


Figure 1.9: Expected Payoffs vs Discretization length N for different strategies

We noticed that the approximation using the 3 pieces linear function is better.

1.9.1 Sensitivity Analysis

We will like to know what happends when the bids or the capacity vary in a quantity ε small. We will consider the simplier case when there is only 2 firms and no resistance losses.

Suppose first that every bid changes in ε

Expected Value					
N	$\varepsilon = +0.001$	$\varepsilon = -0.001$	$\varepsilon = 0.01$	$\varepsilon = -0.01$	**No variation**
6	0.3543	0.3530	0.3571	0.3472	0.3537
7	0.3559	0.3543	0.3464	0.3546	0.3552
8	0.3584	0.3573	0.3627	0.3513	0.3580
9	0.3465	0.3490	0.3459	0.3432	0.3496
10	0.3468	0.3451	0.3414	0.3429	0.3460

Table 1.23: Expected payoffs by player 1 when varying the bids

We can see that when $\varepsilon = 0.001$ the changes are around 0.2%, while for a change ten times larger, i.e, $\varepsilon = 0.01$ the changes are between 0.2% and 2%.

We can also note that there is a tendency to have a higher expected gain when all the bids are larger, however, it is not always so. In the case in which the amount ε was subtracted the expected values decreased in all cases.

Suposse the capacity $\bar{q}$ changes in ε:

Expected Value					
N	$\varepsilon = +0.001$	$\varepsilon = -0.001$	$\varepsilon = 0.01$	$\varepsilon = -0.01$	**No variation**
6	0.3495	0.3523	0.3462	0.3682	0.3537
7	0.3556	0.3532	0.3338	0.3668	0.3552
8	0.3578	0.3567	0.3358	0.3586	0.3580
9	0.3578	0.3485	0.3368	0.3680	0.3496
10	0.3361	0.3444	0.3287	0.3662	0.3460

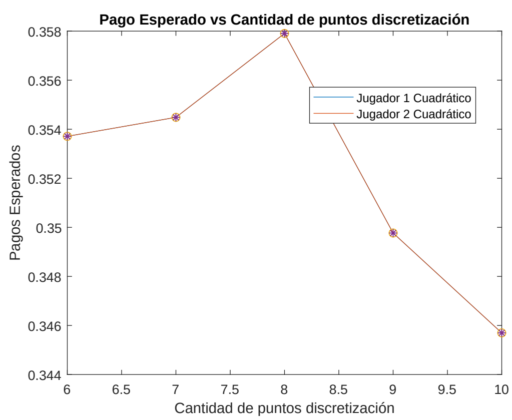
Table 1.24: Expected payoffs by player 1 when varying the capacity $\bar{q}$

1.10 Simulations with small resistances

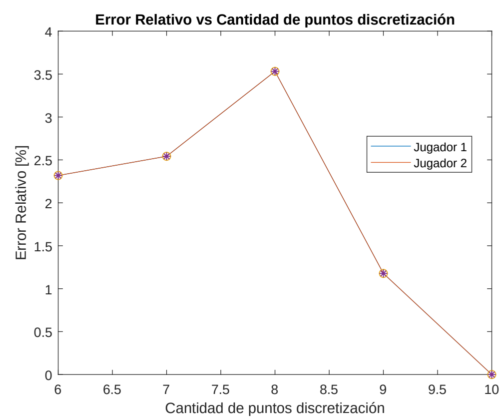
We introduce a resistance $r = 10^{-3}$ to the system so that the ISO problem is the general one:

$$
ISO(c, \mathrm{d}) = \begin{cases} \min & \displaystyle\sum_{n \in G} \alpha_n q_n + \beta_n q_n^2 \\ s.t & \displaystyle\sum_{\mathrm{e} \in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq q_n + \sum_{\mathrm{e} \in K_n} f_\mathrm{e} sgn(\mathrm{e}, n), \quad n \in G \\ & \displaystyle\sum_{\mathrm{e} \in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq \sum_{\mathrm{e} \in K_n} f_\mathrm{e} sgn(\mathrm{e}, n), \quad n \notin G \\ & q_n \in [0, \bar{q}_n] \\ & f \in F \end{cases}
$$

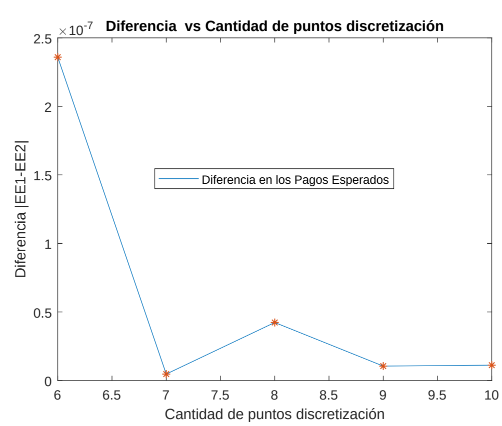
We obtain the following results:



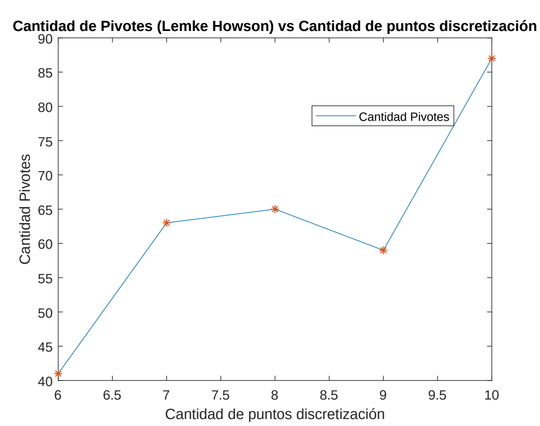
(a) Expected Payoffs



(b) Relative error %

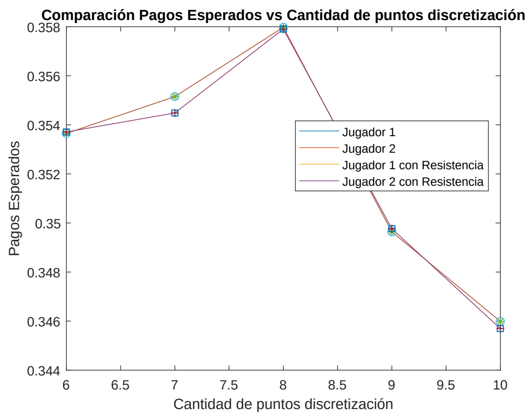


(c) Difference between expected payoffs for players 1 and 2

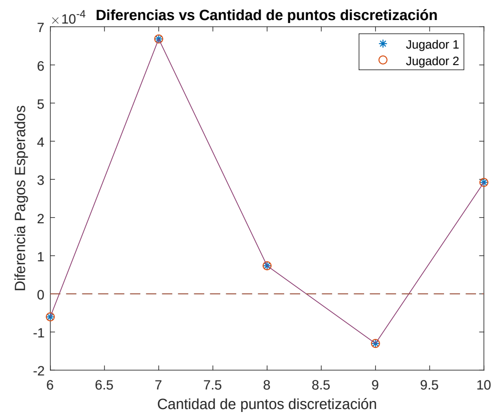


(d) Number of pivots used by Lemke-Howson

We see that the results are similar which makes sense since the resistance is small. Below are 2 graphs, the first one has the expected values (profit) of both players in the case without resistance and in the case with small resistance, the second has the differences of these profit for each player in both cases.



(e) Expected Payoffs both cases



(f) Difference between expected payoffs

It should be noted that by introducing the same resistance to both players we continue to have symmetry which is reflected in the results.

In general, the number of pivots used by the Lemke Howson algorithm decreases when considering the case with resistance. The following table shows the values.

Discretization Length	6	7	8	9	10
	Number of Pivots				
Without resistance	45	65	65	59	87
With resistance	41	64	65	60	86

Therefore we can compute the Nash equilibria a little bit faster, at the cost of solving the ISO problem slower. Since we can solve the ISO problem really fast when there is no resistance and the results are similar when the resistance is small, we focus on developing in the first instance, algorithms and routines for the case in which the resistance is zero.

Chapter 2

Scenarios Approach

The complexity of the optimal bidding problem is greatly compounded by the fact that the calculation of the shadow price and the dispatched quantities, depends on the knowledge of price vectors for all generators, as well as their generation capacity. However, this information is not available to any single company at the time of its bid. Therefore, the bidding strategy has to take intro account the uncertainty around these values.

One approach to model this simultaneous competition process is through a Nash equilibrium which is what was done in chapter 1. Another approach, which is done in this chapter, is to define a set of scenarios for the remaining generators and maximize the expected profit over all scenarios. This approach was presented by Baíllo et al. [8]. The idea is to assume that, after the clearing of each market mechanism, information about the submitted aggregate offer and demand curves is made publicly available and agents can then build scenarios for its rivals bids.

Let the bids from generators $j \in G \setminus \{\mathrm{i}\}$ be represented by a set of scenarios indexed by s, which occur with exogenous probabilities $(p_s)_{s \in S}$. Then the problem that generator i solves is:

$$B^{\mathrm{i}}(\mathrm{d}, p) = \begin{cases} \max & \sum_{s \in S} p_s \lambda_s(\mathrm{d}) q_s^{\mathrm{i}}(\mathrm{d}) \\ s.t & (q_s^{\mathrm{i}}, \lambda_s) \in ISO(s, \mathrm{d}) \quad \forall s \in S \end{cases}$$

Where $ISO(s, \mathrm{d})$ is the dispatch problem solved by the ISO when the demand value is d an the scenario $s \in S$ is played.

$$ISO(s, \mathrm{d}) = \begin{cases} min & \sum_{n \in G} c_s^n(q_s^n) \\ s.t & \sum_{\mathrm{e} \in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq q_s^n + \sum_{\mathrm{e} \in K_n} f_\mathrm{e} sgn(\mathrm{e}, n), \quad n \in G \\ & \sum_{\mathrm{e} \in K_n} \frac{r_\mathrm{e}}{2} f_\mathrm{e}^2 + \mathrm{d}_n \leq \sum_{\mathrm{e} \in K_n} f_\mathrm{e} sgn(\mathrm{e}, n), \quad n \notin G \\ & q_s^n \in [0, \bar{q}^n] \\ & f \in F \end{cases}$$

Notation	
G	Set of Generators
d	Total demand
S	Set of possible scenarios
p_s	Probability of scenario s happening
p_S	Vector of probabilities $(p_s)_{s \in S}$
$\lambda_s(\mathrm{d})$	Shadow price associated to scenario s and demand d
$\lambda_S(\mathrm{d})$	Vector of shadow prices $\lambda_s(\mathrm{d})$ $s \in S$
$q_s^\mathrm{i}(\mathrm{d})$	Quantity that the ISO assigns to generator i in scenario s
$q_S^\mathrm{i}(\mathrm{d})$	Vector of quantities q_s^i for $s \in S$
q_s	Vector of quantities q^n for $n \in G$ in scenario s

We also considered $\lambda_s \perp \left(\sum_{n \in G} q_s^n \geq \mathrm{d} \right), \quad \forall s \in S$ which means that is the shadow price associated to the nodal inequality as in chapter 1, and since the ISO problem does not necessarily have a unique solution we added the equipartition property. This can be thought of as if two or more generators have the same bids, they dispatch the same quantity. This option is chosen instead of using the optimistic or pessimistic formulation of the problem since its fair for every player.

In our simplified case, all generators have the same maximum capacity of production, the demand is fixed and there are no energy losses. Therefore the simplified ISO problem is:

$$ISO(s, \mathrm{d}) = \begin{cases} min & \sum_{n \in G} c_s^n(q_s^n) \\ s.t & \sum_{n \in G} q_s^n \geq \mathrm{d} \\ & q_s^n \in [0, \bar{q}] \end{cases}$$

2.1 Linear Bids

Consider the case when we have linear bids, then the problem can be written as:

$$B^{\mathrm{i}}(\alpha, \beta, \mathrm{d}, p)_{lin} = \begin{cases} \max\limits_{\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, q_S^{\mathrm{i}}, \lambda_S} & \sum\limits_{s \in S} p_s \lambda_s(\alpha^{\mathrm{i}}, \alpha^{-\mathrm{i}}, \beta^{\mathrm{i}}, \beta^{-\mathrm{i}}, \mathrm{d}) q_s^{\mathrm{i}}(\alpha^{\mathrm{i}}, \alpha^{-\mathrm{i}}, \beta^{\mathrm{i}}, \beta^{-\mathrm{i}}, \mathrm{d}) = \sum\limits_{s \in S} p_s \lambda_s q_s^{\mathrm{i}} \\ s.t & (q_S^{\mathrm{i}}, \lambda_S) \in ISO(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \mathrm{d}, p) \end{cases}$$

In order to gain some insight, suppose that $\alpha_2 \leq \ldots \leq \alpha_{|G|}$. We are going to solve the problem for generator 1 for this fixed scenario.

At the optimal solution of the lower level problem, for a given price bid α_1, generators are loaded by increasing price until demand is met.

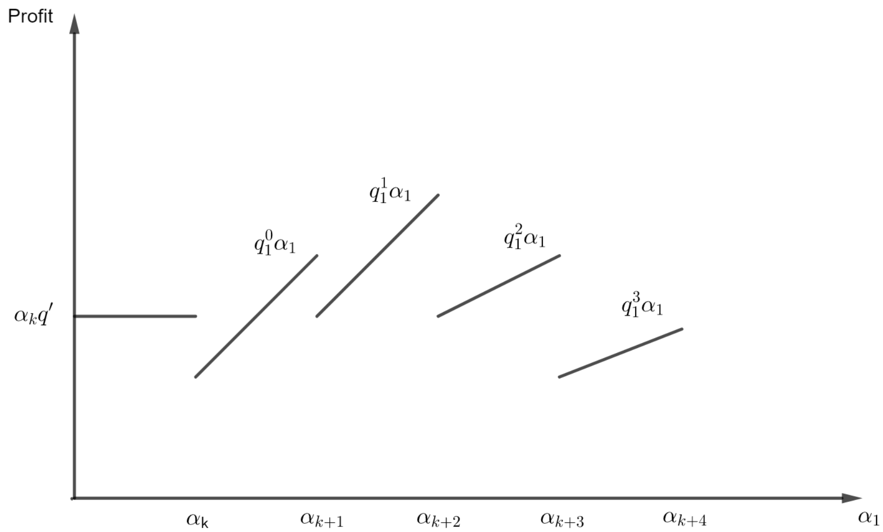


Figure 2.1: Leader's profit

Let k be the index that indicated the maximum number of generators that can be dispatched. Therefore k is the the minimum index such that $q' + \sum_{j=2}^{k} q' > \mathrm{d}$.

If $\alpha_1 \leq \alpha_k$, generator 1 produces its maximum capacity at the optimal solution of the follower problem and the leader's objective is constant and equal to $\alpha_k q'$.

For $\alpha_l < \alpha_1 \leq \alpha_{l+1}$, where $l \in [k, |G| - 1]$, the production of company 1 at the optimal solution of the follower problem is given by $q_1^{l-k} = \max\{0, \mathrm{d} - \sum_{j=2}^{l} q'\}$ and the leader's objective is a linear function of α_1 with slope q_1^{l-k}. Clearly $q_1^{l-k} > q_1^{l+1-k}$ for every l. When α_1 exceeds a sufficiently large value, $q_1 = 0$ at the optimal solution of the follower problem and the leader's profit is zero.

Therefore the leader's objective function is neither continuous nor concave. it is a piecewise linear function that presents local maxima at points where α_1 assumes the value of the price bid of another generator.

It can be seen then that even in one of the simplest cases, the problem is not trivial at all since algorithms can be stuck in stacionary points.

Moreover different formulations give very different results,$\alpha_1 = \alpha_k$ means $\alpha_1 = \alpha_2 = \ldots = \alpha_k$ since we assume that $\alpha_2 \leq \ldots \leq \alpha_{|G|}$. In the optimistic case for the leader the profit will be $\alpha_1 q'$. In the pessimistic case the profit will be $\max\{0, \mathrm{d} - \sum_{j=2}^{k} q'\}\alpha_1 < q'\alpha_1$ and the fair case, this is the one with the equipartition property will give a profit of $\frac{\mathrm{d}}{k}\alpha_1 < q\alpha_1$ where the inequality follows from $q' + \sum_{j=2}^{k} q' = kq' > \mathrm{d} \implies q' > \frac{\mathrm{d}}{k}$. This is why the profit function comes from below the point at the break points.

2.1.1 Convergence result for Linear Bids

Here we will follow Fampa's Paper [14] together with the techniques shown in chapter 1. In order to show the convergence of heuristics.

Since we consider $\mathrm{d} < N\bar{q}$. The previous problem is equivalent to his (MPEC). Then we have:

$$(MPEC)^{\mathrm{i}}(\alpha^{-\mathrm{i}}, \mathrm{d}, p) = \begin{cases} \max\limits_{\alpha^{\mathrm{i}}, q_S^{\mathrm{i}}, \lambda_S} & \sum\limits_{s\in S} p_s \lambda_s q_s^{\mathrm{i}} \\ s.t & \sum\limits_{n\in G} q_s^n = \mathrm{d}, \quad s \in S \\ & 0 \leq q_s^n \leq \bar{q}, \quad n \in G, s \in S \\ & \lambda_s + \pi_s^{q^n} - a_s^n \leq 0, \quad n \in G, s \in S \\ & \pi_s^{q^n} \leq 0, \quad n \in G, s \in S \\ & \sum\limits_{s\in S}\left(\sum\limits_{n\in G} a_s^n q_s^n - \mathrm{d}\lambda_s - \sum\limits_{n\in G} \bar{q}\pi_s^{q^n}\right) = 0 \end{cases}$$

Consider the following problem obtained when we penalize the non-linear complementarity constraint:

$$(MPEC)^{\mathrm{i}}_{pen}(\alpha^{-\mathrm{i}}, \mathrm{d}, p) = \begin{cases} \max\limits_{\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, q_S^{\mathrm{i}}, \lambda_S} & \sum\limits_{s\in S} p_s \lambda_s q_s^{\mathrm{i}} - \mu(a^{\mathrm{i}} q_s^{\mathrm{i}} - \bar{q}\pi_s^{q^{\mathrm{i}}} + \mathrm{d}\lambda_s) \\ s.t & \sum\limits_{n\in G} q_s^n = \mathrm{d}, \quad s \in S \\ & 0 \leq q_s^n \leq \bar{q}, \quad n \in G, s \in S \\ & \lambda_s + \pi_s^{q^n} - a_s^n \leq 0, \quad n \in G, s \in S \\ & \pi_s^{q^n} \leq 0, \quad n \in G, s \in S \end{cases}$$

Where $\mu > 0$ is the penalty parameter.

We will verify that the penalty scheme considered is an exact penalty scheme, i.e, when the penalty parameter is large enough the complementary constraints will be satisfied. This result was proven by Anandalingam and White [3] for linear bilevel programs and by Labbé,Marcotte and Savard [17] for the model of taxation which is a bilinear-bilinear bilevel problem. Here the bidding problem is an extension of the taxation problem.

Theorem 2.1 *There is a penalty parameter $\bar{\mu} > 0$ such that problems $B^{\mathrm{i}}(\alpha^{-\mathrm{i}}, \mathrm{d}, p)$ and $(MPEC)^{\mathrm{i}}_{pen}$ are equivalent for every $\mu > \bar{\mu}$*

PROOF. For simplicity we are going to use $\mathrm{i} = 1$, i.e, solve for the first player.

Following the notation used in Labbé's paper [17], we can write problem $(MPEC)^{\mathrm{i}}(\alpha^{-\mathrm{i}}, \mathrm{d}, p)$ as:

$$\begin{cases} \max\limits_{T,x,y\mu} & Tx \\ s.a & Ax \geq b \\ & \mu A = c + y \\ & \mu \geq 0 \\ & \mu(Ax - b) = 0 \end{cases}$$

Where:

1. T is a row vector with values $p_s\lambda_s$, $s \in S$ on the first $|S| := m$ coordinates and 0 in the following $(|G| - 1)m$ coordinates.
2. x is a column vector with values $(q^1_{s_1}, q^1_{s_2}, \ldots, q^1_{s_m}, q^2_{s_1}, \ldots, q^{|G|}_{s_m})$.
3. c is a row vector with values 0 on the first m coordinates and equal to the bids α^n_s , $n \in G \setminus \{1\}$. Notice that this bids are parameters of the problem.
4. y is a row vector with values α^1_s , $s \in S$ in the first m coordinates and 0 in the following $(|G| - 1)m$ coordinates.
5. Since x is the quantity vector q_S, the matrix A is simply the one that has the inequalities $q^n_s \in [0, \bar{q}]$ for $n \in G$,$s \in S$ plus the demand inequality $\sum_{n\in G} q^n_s \geq \mathrm{d}$ for $s \in S$.
6. We define the matrix P such that $\mu P = T$, this follows from the fact that the probabilities p_s ,$s \in S$ are parameters of the problem and the multipliers λ_s are part of the multipliers from μ. Therefore P is a matrix with values p_s,$s \in S$ and 0.

This yields to the bilinear program:

$$(MPEC)^{\mathrm{i}}_{pen}(\alpha^{-\mathrm{i}}, \mathrm{d}, p) = \begin{cases} \max\limits_{T,x,y,\mu} & Tx - K\mu(Ax - b) \\ s.a & Ax \geq b \\ & \mu A = c + y \\ & \mu \geq 0 \end{cases}$$

From here the proof is the almost the same as the paper [17], the main difference is that we have a more explicit relation between T and μ, also our follower problem has the from

$(c+y)x$ instead of $(c+T)x+\mathrm{d}y$, but as we'll see in the proof, the explicit relation between T and μ helps us with the proof.

Since strong duality holds for the lower level problem, a dual optimal solution of the lower level problem is achieved at an extreme point of the dual polyhedron $\Gamma = \{\mu : \mu A = c+y, \mu \geq 0\}$, otherwise the maximum will be 0 which is not optimal. Denote by $\{\mu^{\mathrm{i}}, \mathrm{i} \in I\}$ its extreme points and by $\{(x^j, y^j), j \in J\}$ the set of extreme points of the primal polyhedron Π. Since this polyhedron is bounded by the hypothesis of the network, we may also assume, without loss of generality, that $\Pi = conv_{j\in J}\{(x^j, y^j)\}$. The maximum of the disjoint bilinear program must be achieved at an extreme point $(x^j, y^j, \mu^{\mathrm{i}}) \in \Pi \times \Gamma$ (see [25]). Let:

$$K \geq K^* = \max_{\mathrm{i}\in I, j\in J} \left\{ \frac{\mu^{\mathrm{i}} P x^j}{\mu^{\mathrm{i}}(Ax^j - b)} : \mu^{\mathrm{i}}(Ax^j - b) > 0 \right\}$$

Thus a point $(x^j, y^j, \mu^{\mathrm{i}})$ qualifies for optimality if:

$$\mu^{\mathrm{i}}(Ax^j - b) = 0$$

Since from the choise of K, at any other extreme point, the leader's objective is negative, and therefore non optimal. It follows that the term $\mu(Ax-b)$ constitutes an exact penalty function for the mathematical program $(MPEC)^{\mathrm{i}}_{pen}(\alpha^{-\mathrm{i}}, \mathrm{d}, p)$, thus $B^{\mathrm{i}}(\alpha, \beta, \mathrm{d}, p)_{lin}$ and $(MPEC)^{\mathrm{i}}_{pen}(\alpha^{-\mathrm{i}}, \mathrm{d}, p)$ are equivalent, whenever K is larger than K^* □

2.2 Piecewise Linear case

As we saw in Chapter 1 the linear case is contained in the linear case by pieces. In this case the problem can be written as:

$$B^{\text{i}}(\alpha,\beta,\text{d},p)=\begin{cases}\max\limits_{\alpha^{\text{i}},\beta^{\text{i}},q_S^{\text{i}},\lambda_S} & \sum\limits_{s\in S}p_s\lambda_s(\alpha^{\text{i}},\alpha^{-\text{i}},\beta^{\text{i}},\beta^{-\text{i}},\text{d})q_s^{\text{i}}(\alpha^{\text{i}},\alpha^{-\text{i}},\beta^{\text{i}},\beta^{-\text{i}},\text{d})=\sum\limits_{s\in S}p_s\lambda_s(q'x_{2,s}^{\text{i}}+\bar{q}x_{3,s}^{\text{i}})\\ s.t & (q_S^{\text{i}},\lambda_S)\in ISO(\alpha^{\text{i}},\beta^{\text{i}},\text{d},p)\end{cases}$$

$$ISO(\alpha^{\text{i}},\beta^{\text{i}},\text{d},p)=\begin{cases}\min\limits_{q_S} & \sum\limits_{s\in S}\sum\limits_{n\in G}\left(\alpha_s^n q' x_{2,s}^n+[(\bar{q}-q')\beta_s^n+\alpha_s^n q']x_{3,s}^n\right)\\ s.t & \sum\limits_{n\in G}\left(q'x_{2,s}^n+\bar{q}x_{3,s}^n\right)\geq \text{d},\quad \forall s\in S\\ & x_{1,s}^n+x_{2,s}^n+x_{3,s}^n=1,\quad \forall n\in G,\forall s\in S\\ & x_{1,s}^n\leq y_{1,s}^n,\quad \forall n\in G,\forall s\in S\\ & x_{3,s}^n\leq y_{2,s}^n,\quad \forall n\in G,\forall s\in S\\ & y_{1,s}^n+y_{2,s}^n=1,\quad \forall n\in G,\forall s\in S\\ & y_{j,s}^n\in\{0,1\}\quad \text{for } j=1,2.\quad \forall n\in G,\forall s\in S\\ & x_{j,s}^n\geq 0\quad \text{for } j=1,2,3.\quad \forall n\in G,\forall s\in S\end{cases}$$

Notation	
G	Set of Generators
$\alpha^{\text{i}},\beta^{\text{i}}$	Generator i slopes
α,β	Vectors with the slopes of all Generators
$\alpha^{-\text{i}},\beta^{-\text{i}}$	Vector with de slopes of all Generators but i
d	Total demand
S	Set of possible scenarios
p_s	Probability of scenario s happening
p_S	Vector of probabilities $(p_s)_{s\in S}$
$\lambda_s(\alpha,\beta,\text{d})$	Shadow price associated to scenario s
$\lambda_S(\alpha,\beta,\text{d})$	Vector of shadow prices $\lambda_s(\alpha,\beta,\text{d})$ $s\in S$
$q_s^{\text{i}}(\alpha,\beta,\text{d})$	Quantity that the ISO assigns to generator i in scenario s
$q_S^{\text{i}}(\alpha,\beta,\text{d})$	Vector of quantities q_s^{i} for $s\in S$
q_s	Vector of quantities q^n for $n\in G$ in scenario s
$x_{2,s}^n,x_{3,s}^n$	Continuous variables belonging to the interval [0,1] which are part of the decomposition of q_s^n
$y_{2,s}^n,y_{3,s}^n$	Binary variables associated to $x_{2,s}^n$, and $x_{3,s}^n$ resp.

Since the slopes $\alpha^{-\text{i}}$ and $\beta^{-\text{i}}$ are considered in the scenario s, we will write $\lambda_s(\alpha^{\text{i}},\beta^{\text{i}},\text{d})$ instead of $\lambda_s(\alpha^{\text{i}},\alpha^{-\text{i}},\beta^{\text{i}},\beta^{-\text{i}},\text{d})$ and the same with q_s^{i}.

Here we used the model from Chapter 1.

$$q_s^n=q'x_{2,s}^n+\bar{q}x_{3,s}^n\quad \forall n\in G,\forall s\in S$$

$$q_s \text{ Satisfies the equipartition property} \forall s \in S$$

$$\lambda_s \perp \left(\sum_{n \in G} q_s^n \geq \mathrm{d} \right), \quad \forall s \in S$$

2.2.1 Convergence result for Piecewise Linear Bids

The problem written in the previous way presents the complication that it is not straightforward how to write it linearly so we can use or extend in a simple way the result of the linear case. So that for this part, we propose to see the problem as follows (for simplicity we assume the case in which the function has 2 parts with break point q').

Each generator $n \in G$ is going to be considered as 2 generators one with bid α_n and capacity q' an other with capacity $\bar{q} - q'$ and bid β_n. As we saw in chapter 1, the dispatch problem solved by the ISO is dispatching energy from the generator with the lowest bid until demand is reached. So because of the structure of the problem, it is not necessary to incorporate more restrictions. Therefore the piecewise linear problem can be seen as a bilinear problem. So we can apply the previous penalty algorithm along with its convergence result.

$$B^{\mathrm{i}}(\alpha, \beta, \mathrm{d}, p) = \begin{cases} \max & \sum_{s \in S} p_s \lambda_s \tilde{q}^{\mathrm{i}} \\ s.t & (q_S^{\mathrm{i}}, \lambda_S) \in ISO(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \mathrm{d}, p) \end{cases}$$

Since we have a convex piecewise linear function, we can write it as a maximum of affine functions.

$$ISO(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \mathrm{d}, p) = \begin{cases} \min\limits_{q_S} & \sum_{s \in S} \sum_{n \in G} \max\{\alpha_s^n q_s^n, \beta_s^n q_s^n + (\alpha_s^n - \beta_s^n) q'\} \\ s.a & \sum_{n \in G} q_s^n \geq \mathrm{d}, \quad \forall n \in G, s \in S \\ & q_s^n \in [0, \bar{q}], \quad \forall n \in G, s \in S \end{cases}$$

Which is equivalent to the lineal program:

$$ISO(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \mathrm{d}, p) = \begin{cases} \min\limits_{q_S} & \sum_{s \in S} \sum_{n \in G} t_s^n \\ s.a & \alpha_s^n q_s^n \leq t_s^n, \quad \forall n \in G, s \in S \\ & \beta_s^n q_s^n + (\alpha_s^n - \beta_s^n) q' \leq t_s^n, \quad \forall n \in G, s \in S \\ & \sum_{n \in G} q_s^n \geq \mathrm{d}, \quad \forall n \in G, s \in S \\ & q_s^n \in [0, \bar{q}], \quad \forall n \in G, s \in S \end{cases}$$

Then the convergence results follows from Theorem 2.1.

2.3 Numerical results

2.3.1 Procedure

Here we discuss the general procedures to compute generator's i expected payoff and best strategy. For simplicity we'll considerer i = 1.

The following pseudocodes gives the idea of the algorithms that are programmed for the different cases in type of strategies or number of players

Result: Expected Payoff and best strategy for generator 1
initialization;
Input: Number of players $|G|$, Maximum capacity value $\bar{q}_k$ for each player $k \in G$ and probability vector p_S of each scenario;
Step 1: For $k \in \{2, \dots, |G|\}$. Define the sets $I_k := \{j \in \{1, \dots, |S_k|\} : p_k(j) > 0\}$;
Step 2: Define a scenario as
$s \in S = \{(t_{j_2}, \dots, t_{j_{|G|}} \in S_2 \times \dots \times S_{|G|} : j_2 \in I_2, \dots, j_{|G|} \in I_{|G|}\}$;
for $\mathrm{i} \in |S_1|$ **do**
1. Solve the ISO's problem using our algorithm from chapter 1 ;
2. Compute the value $p_s \lambda(t_\mathrm{i}, s) q^\mathrm{i}_{t_\mathrm{i},s}$, where $s \in S$
3. Save the value $\sum_{s \in S} p_s \lambda(t_\mathrm{i}, s) q^\mathrm{i}_{t_\mathrm{i},s}$ as the new maximum if its greater than the previous maximum

end
Step 3: Return Expected Payoff of generator 1 and best strategy.
Algorithm 2: Scenarios Approach Algorithm 1

And the penalization heuristic

Result: Expected Payoff and best strategy for generator 1
initialization;
Input: Number of players $|G|$, Maximum capacity value $\bar{q}_k$ for each player $k \in G$, the probability vector p_S of each scenario and some initial point $\tilde{q}^s_n, n \in G, s \in S$;
while *Complementary condition* $\neq 0$ **do**
Solve penalized problem with $q^\mathrm{i}_s = \tilde{q}^\mathrm{i}_s$ and obtain a solution $\tilde{\alpha}$,$\tilde{\lambda_s}$,$\pi^s_{q_n}$ $n \in G, s \in S$;
Solve the ISO problem for each scenario $s \in S$, considering $\alpha = \tilde{\alpha}$ and obtain a solution $\tilde{q}^s_n$,$n \in G$;
Increase μ
end
Return: Expected Payoff of generator 1 and best strategy.
Algorithm 3: Scenarios Approach Algorithm 2

As we saw, we can solve for piecewise linear functions using the linear case and the ISO problem can be solved using our algorithm form chapter 1. The convergence of the sequence produced by this procedure to the feasible set of the problem is guaranteed by theorem 1. The idea of the heuristic is to start with the best solution for the leader problem and move from this solution to the feasible set of the problem, where the complementary conditions of the follower problem are satisfied.

The heuristics consider the solution of each nonlinear penalized problem iteratively and approximately, through the solution of linear programs, so the non-convex optimization problem is replaced by a sequence of linear programs, which tend to be easier and where the primal variables q_s^j are separated from the dual variables and from the bids.

The sequence may not converge to an optimal solution of the bilevel program. Since bilevel problems are non-convex, the heuristics may converge to a local optimal solution, as illustrated by Figure 2.1. We notice that since the leader's objective function is a discontinuous piecewise linear function of the bids, all stationary points are either locally minimal or locally maximal. Because of the first step on the while, the solution obtained by the heuristic is always a local maximum. In order to avoid local optimum, we can do clasic techniques like in [14] that is making a diversification of the initial solution followed with a local search.

2.3.2 Numerical results for 2 Players

First we set the values $\bar{q}$, q' , d and N (the discretization length of the interval $[0,\bar{\lambda}]$). Then we need to define the space of strategies, since every player choose two slopes with $\alpha < \beta$, the total number of strategies for each player is $\frac{N(N-1)}{2} = m$ and we list these strategies in the following way:

$$t_1 = [\alpha_1, \beta_2], t_2 = [\alpha_1, \beta_3], \dots t_{N-1} = [\alpha_1, \beta_N], t_N = [\alpha_2, \beta_3], \dots, t_m = [\alpha_{N-1}, \beta_N]$$

Since we are going to solve for generator 1. We have to assign or give as input the probability that the other generator choose a certain strategy. Therefore, if $p_k(\mathrm{i})$ is the probability that generator k chooses the i strategy, the probability of generator 2 choosing strategy i is $p_2(\mathrm{i})$

Lets define the following set:

$$J := \{j \in \{1, \dots, m\} : p_2(j) > 0\}$$

Then we can define a scenario as $s \in S = \{t_j \in S_2 : j \in J\}$

Therefore for each strategy $\mathrm{i} = 1, \dots, m$ for player 1, we solve the ISO's problem using our solution, and while we are solving it, we compute the value $p_s \lambda(t_\mathrm{i}, s) q^\mathrm{i}_{t_\mathrm{i},s}$, where $s \in S$. So for fixed i we compute the value $\sum_{s \in S} p_s \lambda(t_\mathrm{i}, s) q^\mathrm{i}_{t_\mathrm{i},s}$ and save the strategy and value as the new maximum only if its greater or equal to the previus maximum (the first maximun is $\mathrm{i} = 1$ by default).

We had results about the Nash equilibrium for 2 players, in particular, we have the mixed nash equilibria probabilities for each player, so if we use that probabilities as the $p_k(j)$ we obtain the following results for $\bar{q} = 1$, $q' = 0.5$ and d = 1.6:

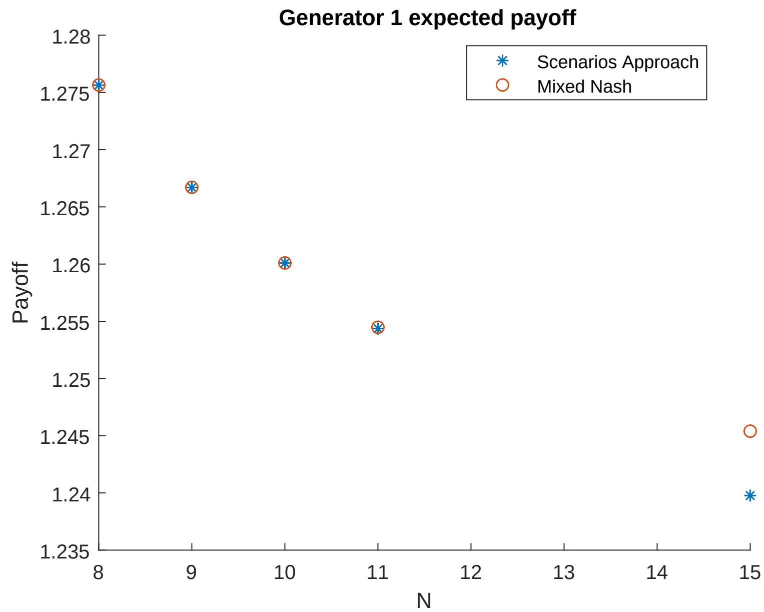


Figure 2.2: Player 1 Payoffs

N	Mixed Nash	Scenarios Approach
8	1.2756	1.2756
9	1.2667	1.2667
10	1.2601	1.2601
11	1.2545	1.2545
15	1.2454	1.2398

Table 2.1: Player 1 Payoffs

It can be seen that the payoffs are really similar. Naturally the payoffs under scenarios approach is less or equal than the payoffs or the Mixed Nash equilibria, since it can be seen as playing a pure strategy.

The advantage of this method is that we can solve the problem for large discretizations.

Let's suppose player 2 plays every strategy with probability $\frac{1}{N}$.Then we get the following payoffs for player 1:

N	Mixed Nash	Scenarios Approach Uniform Probability
8	1.2756	1.3000
9	1.2667	1.2889
10	1.2601	1.2800
11	1.2545	1.2727
15	1.2454	1.2533

Table 2.2: Player 1 Payoffs Uniform Probability for player 2

Here the scenarios approach payoff is greater than the mixed nash, because player 2 is playing using uniform probability $1/N$ instead of the mixed nash equilibria probability. Notice that for $N = 100$ we can't solve the Nash equilibria approach, but we can solve the Scenarios Approach and get a payoff of 1.2080 which is not that far of the previous results for smaller values of N. We'll see more about the different probabilities that can be used when we don't have mixed strategies in a section later on.

2.3.3 Sensitivity Analysis

First we will do a small perturbation ε on the capacity value $\bar{q}$.

We get the following results for $\varepsilon = 0.005$:

N	Payoff Scenarios P1 Without Perturbation	Payoff Scenarios P1 With Perturbation	Difference %
8	1.2756	1.2653	0.8075
9	1.2667	1.2563	0.8210
10	1.2601	1.2498	0.8174
11	1.2545	1.2440	0.8370
15	1.2398	1.2295	0.8308

Table 2.3: $\bar{q} + \varepsilon$ with $\varepsilon = 0.005$

N	Payoff Scenarios P1 Without Perturbation	Payoff Scenarios P1 With Perturbation	Difference %
8	1.2756	1.2859	0.8010
9	1.2667	1.2771	0.8143
10	1.2601	1.2704	0.8108
11	1.2545	1.2647	0.8065
15	1.2398	1.2500	0.8160

Table 2.4: $\bar{q} - \varepsilon$ with $\varepsilon = 0.005$

A small perturbation of 1% the capacity value produces a change the expected payoff of player 1 in 0.8%.

It is also interesting to see how much it changes with respect to the Nash equilibrium perturbated problem and also see the difference in payoff with respect to player 2.

N	Payoff Nash P1	Payoff Scenarios P1	Difference %	Payoff Nash P2	Payoff Scenarios P2	Difference %
8	1.2653	1.2653	0.0012	1.2653	1.9347	34.5976
9	1.2563	1.2563	0.0016	1.2563	1.9437	35.3645
10	1.2499	1.2498	0.0079	1.2499	1.9502	35.9113
11	1.2440	1.2440	0.0005	1.2440	1.9560	36.3997
15	1.2353	1.2295	0.4701	1.2353	1.9705	37.3086

Table 2.5: $\bar{q} + \varepsilon$ with $\varepsilon = 0.005$

N	Payoff Nash P1	Payoff Scenarios P1	Difference %	Payoff Nash P2	Payoff Scenarios P2	Difference %
8	1.2860	1.2859	0.0011	1.2860	1.9141	32.8150
9	1.2772	1.2771	0.0073	1.2772	1.9229	33.5830
10	1.2704	1.2704	0.0004	1.2704	1.9296	34.1622
11	1.2647	1.2647	0.0023	1.2647	1.9353	34.6511
15	1.2555	1.2500	0.4346	1.2555	1.9500	35.6162

Table 2.6: $\bar{q} - \varepsilon$ with $\varepsilon = 0.005$

We can see that even when the Scenarios Approach expected payoff for player 1 is really close to the one from the mixed Nash equilibria, the expected payoff for player 2 increases in $\approx 35\%$.

If we make a perturbation $\varepsilon = 0.005$ on the bids we get the following tables:

N	Payoff Scenarios P1 Without Perturbation	Payoff Scenarios P1 With Perturbation	Difference %	Nash Payoff P1 With Perturbation	Difference Nash and Scenarios %
8	1.2756	1.2787	0.2397	1.2787	0.0058
9	1.2667	1.2697	0.2357	1.2697	0.0015
10	1.2601	1.2631	0.2385	1.2631	0.0004
11	1.2545	1.2574	0.2270	1.2574	0.0005
15	1.2398	1.2428	0.2392	1.2484	0.4521

Table 2.7: bids $+\varepsilon = 0.005$

N	Payoff Scenarios P1 Without Perturbation	Payoff Scenarios P1 With Perturbation	Difference %	Nash Payoff P1 With Perturbation	Difference Nash and Scenari %
8	1.2756	1.2726	0.2349	1.2726	0.0012
9	1.2667	1.2637	0.2382	1.2637	0.0015
10	1.2601	1.2571	0.2401	1.2571	0.0004
11	1.2545	1.2514	0.2507	1.2515	0.0094
15	1.2398	1.2368	0.2449	1.2424	0.4521

Table 2.8: bids $-\varepsilon = 0.005$

Both approachs give us less expected payoff when we make a -0.005 perturbation to the bids than when we make a $+0.005$ perturbation. Also we can see that the problem is not that sensitive to bids changes.

2.3.4 Numerical results for 3 Players

Here we list the strategies the same way as for 2 players. The probability of generator 2 choosing strategy i and generator 3 choosing strategy j is $p_2(\mathrm{i})p_3(j) = p_{\mathrm{i},j}$

Lets define the following sets:

$$J := \{j \in \{1, \dots, m\} : p_2(j) > 0\}$$

$$K := \{k \in \{1, \dots, m\} : p_3(k) > 0\}$$

Then we can define a scenario as $s \in S = \{(t_j, t_k) \in S_2 \times S_3 : j \in J, k \in K\}$

Therefore for each strategy $\mathrm{i} = 1, \dots, m$ for player 1, we solve the ISO's problem using our solution, and while we are solving it, we compute the value $p_s \lambda(t_\mathrm{i}, s) q^\mathrm{i}_{t_\mathrm{i},s}$, where $s \in S$. So for fixed i we compute the value $\sum_{s \in S} p_s \lambda(t_\mathrm{i}, s) q^\mathrm{i}_{t_\mathrm{i},s}$ and save the strategy and value as the new maximum only if its greater or equal to the previus maximum (the first maximun is $\mathrm{i} = 1$ by default).

We had results about the Nash equilibrium for 3 players, in particular, we have the mixed nash equilibria probabilities for each player, so if we use that probabilities as the $p_k(j)$ we obtain the following results for $\bar{q} = 1$, $q' = 0.5$ and $\mathrm{d} = 2$:

N	Nash Equilibria Payoff	Scenarios Approach Payoff	Difference in [%]
11	0.4281	0.4272	0.2102 %
10	0.5095	0.5043	1.02 %
9	0.4557	0.4526	0.6803%
8	0.3737	0.3716	0.5619 %
7	0.3250	0.3247	0.0923 %
6	0.3188	0.3176	0.3774 %
5	0.4608	0.4608	0 %

For $N = 10$ we have 45 strategies for each player, so at most 91125 combinations. The total running time is $2.882s$ while the Nash equilibria approach takes around 1 hour. For $N = 7$ we have 21 strategies for each player, so at most 9261 combinations. The total running time is $0.449s$, while the Nash equilibria approach takes $3844.525s$, so is $\approx$ 7700 times faster. For $N = 5$ we have 10 strategies for each player, so at most 1000 combinations. The total running time is $0.140s$, while the Nash equilibria approach takes $159.786s$, so is $\approx$ 1141 times faster.

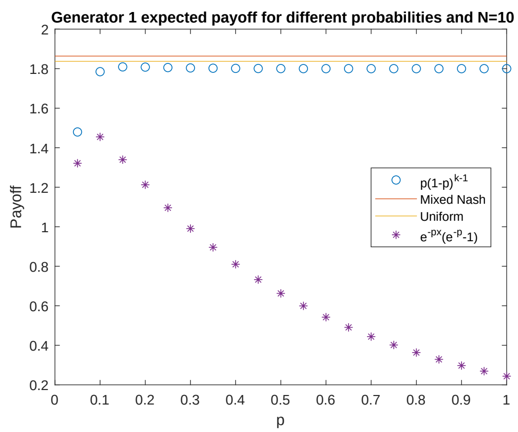
So even though we are finding a pure strategy for player 1, the Payoff is close to the mixed strategies one.

Using our ISO solution is at least 1000 times faster than using a generic algorithm to solve it. However, the computation time of a Nash equilibrium is only reduced by 5%. This is why we'll like to see how the payoff changes if one considers probabilities close or far from to those of the nash equilibrium in mixed strategies for the different scenarios

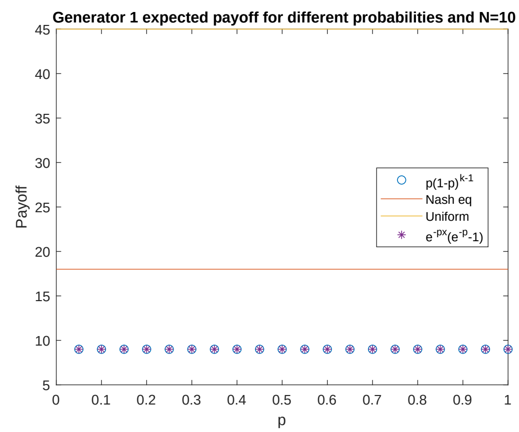
2.3.5 Experimenting with different probabilities

Let's see what happend when we use different probabilities for each scenario. We'll try with uniform , geometric , exponential and with the mixed Nash equilibrium probabilities.

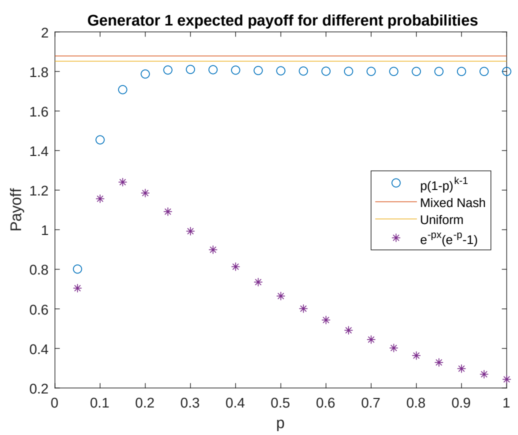
d = 2.9



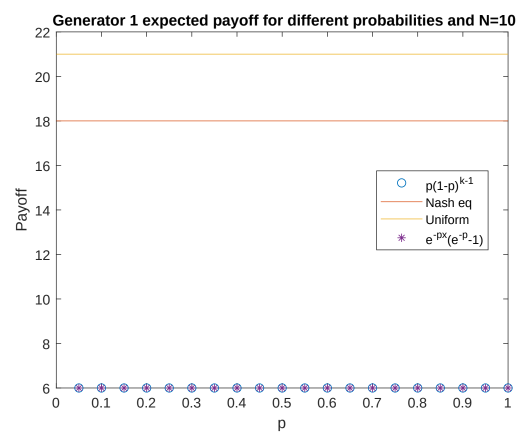
(a) Generator 1 Payoff, case $N = 10$ and d = 2.9



(b) Generator 1 Best strategy index

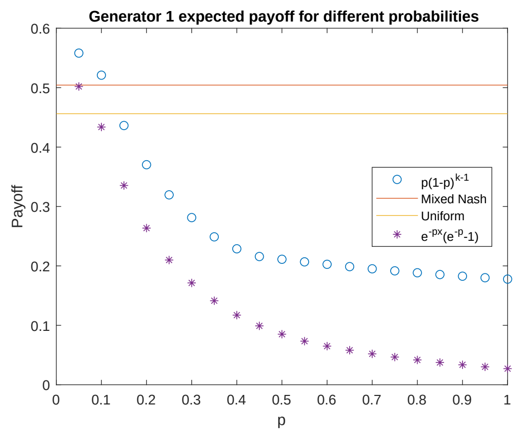


(c) Generator 1 Payoff, case $N = 7$ and d = 2.9

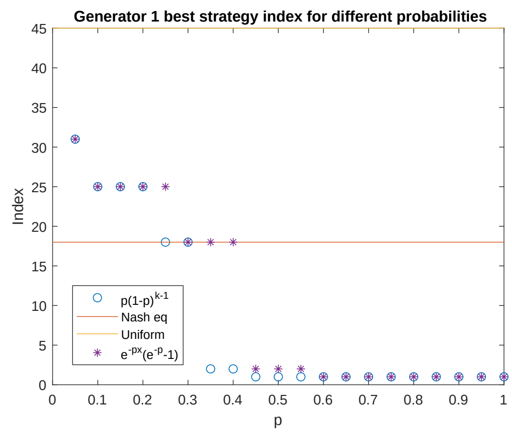


(d) Generator 1 Best strategy index

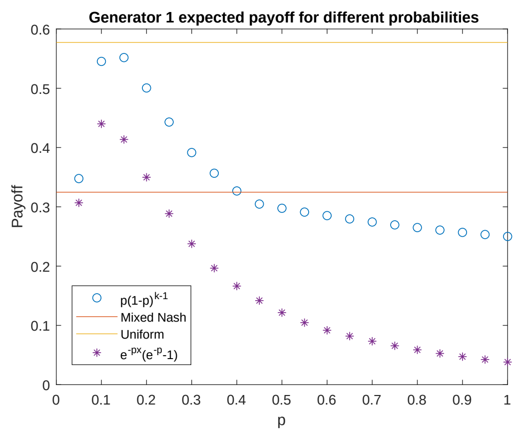
d = 2



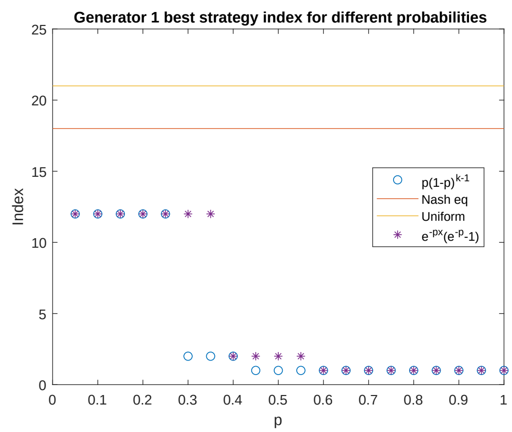
(e) Generator 1 Payoff, case $N = 10$ and d = 2



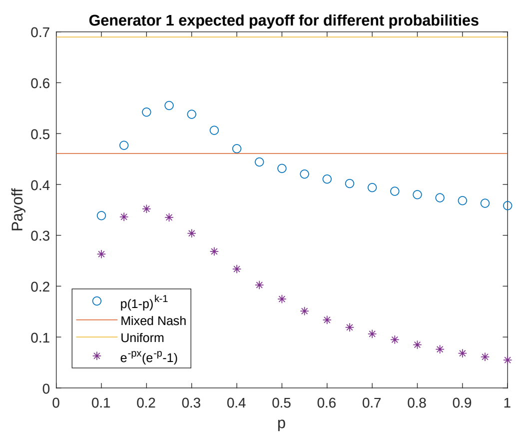
(f) Generator 1 Best strategy index



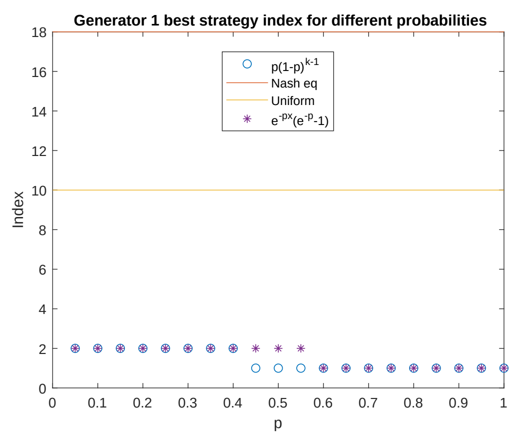
(g) Generator 1 Payoff, case $N = 7$ and d = 2



(h) Generator 1 Best strategy index

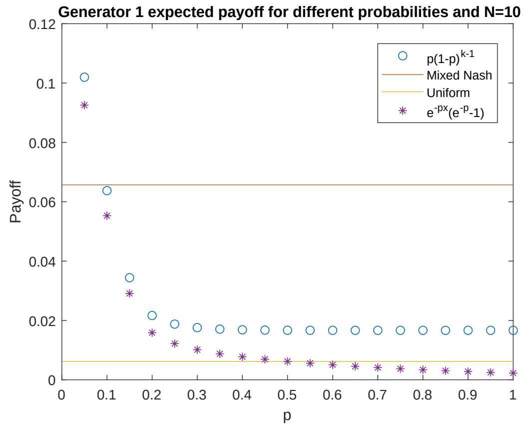


(i) Generator 1 Payoff, case $N = 5$ and d = 2

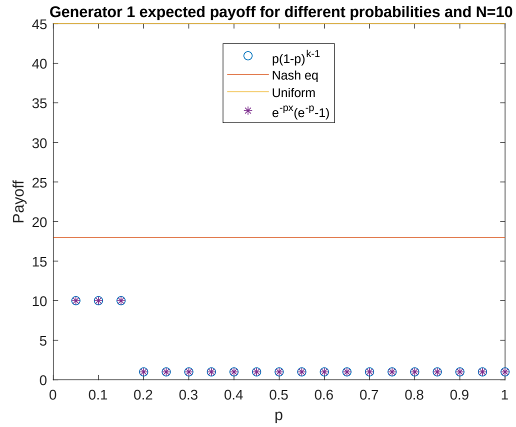


(j) Generator 1 Best strategy index

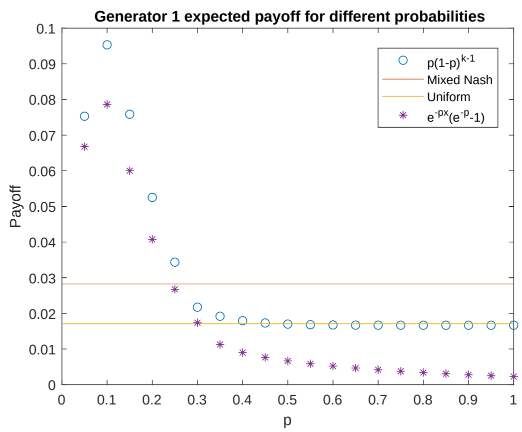
When d = 1



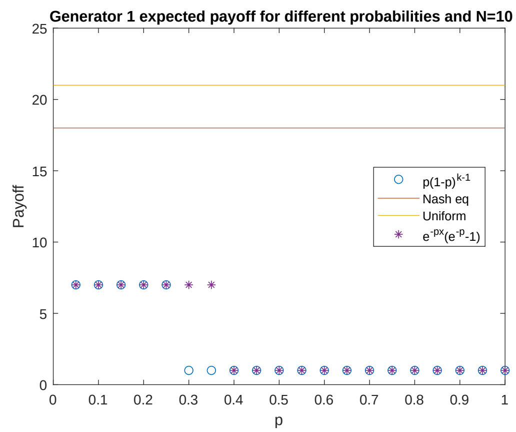
(k) Generator 1 Payoff, case $N = 10$ and d = 1



(l) Generator 1 Best strategy index

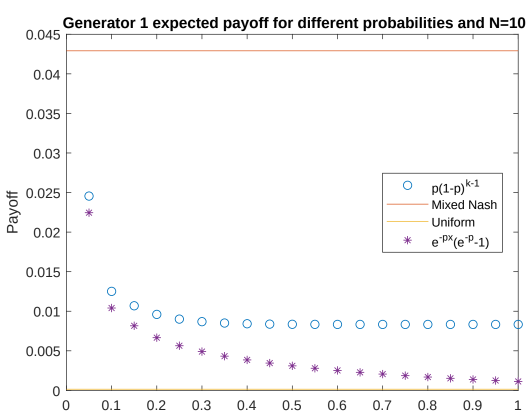


(m) Generator 1 Payoff, case $N = 7$ and d = 1

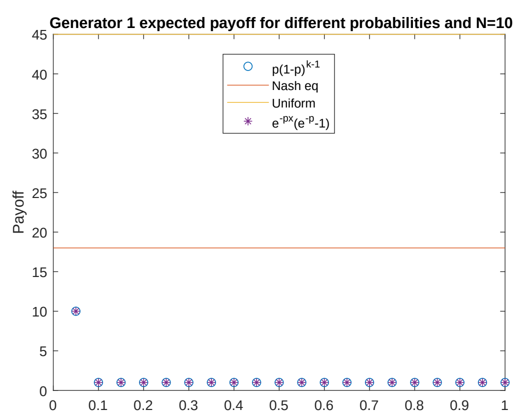


(n) Generator 1 Best strategy index

d = 0.5



(o) Generator 1 Payoff, case $N = 10$ and d = 0.5



(p) Generator 1 Best strategy index

We noticed that in general is better to play strategies with small α and β values. This makes sense since doing it ensures that the generator will be dispatched and since we are considering the shadow price, the payoff will be at least $\alpha q 1_{q \leq q'} + \beta q 1_{q > q'} \geq \alpha q$. Also in most cases there exists an interval of parameters p such that the Scenarios Approach expected payoff is close to the Nash equilibrium one, this plus the sensitivity analysis of the probabilities give us the idea of using this approach with real data in the future, since we can estimate the probabilities and have similar results to the Nash equilibrium one.

2.3.6 Using different slopes

Now every generator can choose slopes in $[0, 2]$, just as before, but they are not equispaced as before, in fact we will use random slopes uniformly distributed in $[0, 2]$ for every generator.

The first colum is player 1 payoff when we compute a Nash equilibrium in mixed strategies, the second colum is when we use the probabilities from the Nash equilibrium for players 2 and 3 as the scenarios probabilities, the third column is when we use a perturbation to those probabilities, in this case we use $\varepsilon = 10^{-4}$ and we add that quantity to every probability an then we normalize it, finally the last column is when we subtract $\varepsilon = 10^{-4}$ to every positive probability and then normalize it.

N	**Mixed Nash Equilibrium**	**Mixed Nash Scenarios**	**Mixed Nash Perturbation + Scenarios**	**Mixed Nash Perturbation - Scenarios**
8	0.5272	0.5266	0.5263	0.5263
9	0.3328	0.3263	0.3263	0.3260
10	0.2438	0.2234	0.2233	0.2236
11	0.5588	0.5590	0.5595	0.5585

We can see that there is not much difference when we use perturbations.

N	**Mixed Nash Scenarios**		**Mixed Nash Perturbation + Scenarios**		**Mixed Nash Perturbation - Scenarios**	
	α_1	β_1	α_1	β_1	α_1	β_1
8	0.2772	0.5022	0.2772	0.5022	0.2772	0.5022
9	0.3048	0.5197	0.3048	0.5197	0.3048	0.5197
10	0.3048	0.5197	0.3048	0.5197	0.3048	0.5197
11	0.5570	0.8435	0.5570	0.8435	0.2540	0.2838

In terms of the best strategy, we can see that it changes in only one case.

Sensitivity with respect to the probabilities used is one of the most important aspects since in practice they are acquired as a result of the clearing of each market mechanism, information about the submitted aggregate offer and demand curves is made publicly available and agents can then build scenarios for its rivals bids. In doing this generator i will not necessarily have the exact probabilities of each scenario. So having little sensitivity to change in these probabilities is a good thing.

2.3.7 Non linearities

Here we can see that even in the simplier case the problem we cannot eliminate nonlinearities without exponentially increasing the number of variables. We can write generator i problem as:

$$B^{\mathrm{i}}(\alpha,\beta,\mathrm{d},p)=\begin{cases}\max\limits_{\alpha^{\mathrm{i}},\beta^{\mathrm{i}},q^{\mathrm{i}}_S,\lambda_S} & \sum\limits_{s\in S}p_s\lambda_s(\alpha^{\mathrm{i}},\alpha^{-\mathrm{i}},\beta^{\mathrm{i}},\beta^{-\mathrm{i}},\mathrm{d})q^{\mathrm{i}}_s(\alpha^{\mathrm{i}},\alpha^{-\mathrm{i}},\beta^{\mathrm{i}},\beta^{-\mathrm{i}},\mathrm{d})=\sum\limits_{s\in S}p_s\lambda_s(q'x^{\mathrm{i}}_{2,s}+\bar{q}x^{\mathrm{i}}_{3,s})\\ s.t & (q^{\mathrm{i}}_S,\lambda_S)\in ISO(\alpha^{\mathrm{i}},\beta^{\mathrm{i}},\mathrm{d},p)\end{cases}$$

Where $ISO(\alpha^{\mathrm{i}},\beta^{\mathrm{i}},\mathrm{d},p)$ is the dispatch program solved by the ISO when

$$ISO(\alpha^{\mathrm{i}},\beta^{\mathrm{i}},\mathrm{d},p)=\begin{cases}\min\limits_{q_S} & \sum\limits_{s\in S}\sum\limits_{n\in G}\left(\alpha^n_s q' x^n_{2,s}+[(\bar{q}-q')\beta^n_s+\alpha^n_s q']x^n_{3,s}\right)\\ s.t & \sum\limits_{n\in G}\left(q'x^n_{2,s}+\bar{q}x^n_{3,s}\right)\geq \mathrm{d},\quad \forall s\in S\\ & x^n_{1,s}+x^n_{2,s}+x^n_{3,s}=1,\quad \forall n\in G,\forall s\in S\\ & x^n_{1,s}\leq y^n_{1,s},\quad \forall n\in G,\forall s\in S\\ & x^n_{3,s}\leq y^n_{2,s},\quad \forall n\in G,\forall s\in S\\ & y^n_{1,s}+y^n_{2,s}=1,\quad \forall n\in G,\forall s\in S\\ & y^n_{j,s}\in\{0,1\}\quad \text{for } j=1,2.\quad \forall n\in G,\forall s\in S\\ & x^n_{j,s}\geq 0\quad \text{for } j=1,2,3.\quad \forall n\in G,\forall s\in S\\ & q^n_s=q'x^n_{2,s}+\bar{q}x^n_{3,s}\quad \forall n\in G,\forall s\in S\\ & \mathrm{q}_s\ \text{ Satisfies the equipartition property } \forall s\in S\\ & \lambda_s\perp\left(\sum\limits_{n\in G}q^n_s\geq \mathrm{d}\right),\quad \forall s\in S\end{cases}$$

We can define the variables $w_s=\lambda_s x^{\mathrm{i}}_{2,s}$ and $z_s=\lambda_s x^{\mathrm{i}}_{3,s}$ and add the following restrictions $0\leq w_s\leq x^{\mathrm{i}}_{2,s}\bar{\lambda}$, $0\leq z_s\leq x^{\mathrm{i}}_{3,2}\bar{\lambda}$. Then the problem can be written as:

$$B^{\mathrm{i}}(\alpha,\beta,\mathrm{d},p)=\begin{cases}\max\limits_{\substack{\alpha^{\mathrm{i}},\beta^{\mathrm{i}},w_S,z_S,\\ x^{\mathrm{i}}_{1,S},x^{\mathrm{i}}_{2,S},x^{\mathrm{i}}_{3,S}}} & \sum\limits_{s\in S}p_s(q'w_s+\bar{q}z_s)\\ s.t & 0\leq w_s\leq x^{\mathrm{i}}_{2,s}\bar{\lambda},\quad \forall s\in S\\ & 0\leq z_s\leq x^{\mathrm{i}}_{3,s}\bar{\lambda},\quad \forall s\in S\\ & (q^{\mathrm{i}}_S,\lambda_S)\in ISO(\alpha^{\mathrm{i}},\beta^{\mathrm{i}},\mathrm{d},p)\end{cases}$$

Then we can recover λ_S by computing $\frac{w_s}{x^{\mathrm{i}}_{2,s}}$ or $\frac{z_s}{x^{\mathrm{i}}_{3,s}}$ for each $s\in S$. We have a problem since we have to ensure that in the optimum we can't have $x^{\mathrm{i}}_{2,s}=x^{\mathrm{i}}_{3,s}=0$, otherwise we can not determine the value of λ_s for that scenario.

The other way is to introduce the variables

$$z_{1,s}=\frac{1}{2}(\lambda_s+q'x^{\mathrm{i}}_{2,s}),\, z_{2,s}=\frac{1}{2}(\lambda_s-q'x^{\mathrm{i}}_{2,s})$$

$$z_{3,s} = \frac{1}{2}(\lambda_s + \bar{q}x^{\mathrm{i}}_{3,s}), z_{4,s} = \frac{1}{2}(\lambda_s - \bar{q}x^{\mathrm{i}}_{3,s})$$

with the restrictions:

$$0 \leq z_{1,s} \leq \frac{1}{2}(\bar{\lambda} + q'), -\frac{1}{2}q' \leq z_{2,s} \leq \frac{1}{2}\bar{\lambda}$$
$$0 \leq z_{3,s} \leq \frac{1}{2}(\bar{\lambda} + \bar{q}), -\frac{1}{2}\bar{q} \leq z_{4,s} \leq \frac{1}{2}\bar{\lambda}$$

Notice the following:

$$z^2_{1,s} - z^2_{2,s} = \lambda_s q' x_{2,s}$$
$$z_{1,s} - z_{2,s} = q' x_{2,s}$$
$$z^2_{3,s} - z^2_{4,s} = \lambda_s \bar{q} x_{3,s}$$
$$z_{3,s} - z_{4,s} = \bar{q} x_{3,s}$$

Then we can write the problem as:

$$B^{\mathrm{i}}(\alpha, \beta, \mathrm{d}, p) = \begin{cases} \max\limits_{\substack{\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, z_{1,S} \\ z_{2,S}, z_{3,S}, z_{4,S}}} & \sum\limits_{s \in S} p_s (z^2_{1,s} - z^2_{2,s} + z^2_{3,s} - z^2_{4,s}) \\ s.t & 0 \leq z_{1,s} \leq \frac{1}{2}(\bar{\lambda} + q'), \quad \forall s \in S \\ & -\frac{1}{2}q' \leq z_{2,s} \leq \frac{1}{2}\bar{\lambda}, \quad \forall s \in S \\ & 0 \leq z_{3,s} \leq \frac{1}{2}(\bar{\lambda} + \bar{q}), \quad \forall s \in S \\ & -\frac{1}{2}\bar{q} \leq z_{4,s} \leq \frac{1}{2}\bar{\lambda}, \quad \forall s \in S \\ & (q^{\mathrm{i}}_S, \lambda_S) \in ISO(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \mathrm{d}, p) \end{cases}$$

Which is quadratic and can be approximated by piecewise linear functions. Then after applying a binary descomposition scheme we get a MILP that can provide an optimal solutiuon to the strategic bidding problem, but it presents the drawback to deal with a large number of integer variables as the number of generators increase. This has motivated the development of alternatives solution approaches, such as the ones presented in this chapter, which can algo be used to generate bounds to be used in a branch-and-bound scheme.

As we can see from Fampa's paper [14] even in the linear bids case the MILP formulation can't be solved for 5 companies and 10 scenarios. Therefore it's not worth it trying in the piecewise linear case, since with our algorithms we can solve that problem and biggers ones in seconds.

2.4 Quadratic Bids

We considerer quadratic bids as in chapter 1. Therefore for 2 player we can use the analytic solution and we can see if the expected values are similar to those we got in chapter 1.

In fact, we get the following table for small values of N and using the same parameters than in chapter 1, i.e , $\bar{q} = 0.5$ and d $= [0.05; 0.1; 0.15; 0.2; 0.25; 0.75; 0.8; 0.85; 0.9; 0.95]$.

N	Mixed Nash	Scenarios Approach
6	0.3537	0.3537
7	0.3552	0.3552
8	0.3580	0.3580
9	0.3496	0.3496
10	0.3460	0.3460
11	0.3471	0.3471
12	0.3448	0.3443
13	0.3452	0.3452
14	0.3468	0.3468
15	0.3436	0.3436

Table 2.9: $\bar{q} = 0.5$

Here we solved the scenarios problem for each d and then took the average. We noticed that the results are equal at least until the fifth power for almost all N.

N	Mixed Nash	Scenarios Approach
6	0.3530	0.3530
7	0.3540	0.3540
8	0.3574	0.3574
9	0.3492	0.3492
10	0.3452	0.3452
11	0.3505	0.3505
12	0.3452	0.3448
13	0.3464	0.3464
14	0.3496	0.3496
15	0.3471	0.3471

Table 2.10: $\bar{q} = 0.5 - 0.001$

N	Mixed Nash	Scenarios Approach
6	0.3501	0.3501
7	0.3562	0.3562
8	0.3520	0.3520
9	0.3465	0.3465
10	0.3367	0.3367
11	0.3454	0.3454
12	0.3442	0.3436
13	0.3470	0.3470
14	0.3471	0.3471
15	0.3447	0.3447

Table 2.11: $\bar{q} = 0.5 + 0.001$

Tables (2.10) and (2.11) are when we do a small perturbation $\pm 0.2\%$ of de $\bar{q}$ value. We can see that the payoffs changes between 0.1% to 1%.

Now we use $\bar{q} = 0.5$ fixed, and do a small perturbation on the bids and probabilities.

N	Mixed Nash	Scenarios Approach
6	0.3530	0.3530
7	0.3543	0.3543
8	0.3573	0.3573
9	0.3490	0.3490
10	0.3451	0.3451
11	0.3485	0.3485
12	0.3443	0.3438
13	0.3448	0.3448
14	0.3460	0.3460
15	0.3462	0.3459

Table 2.12: $\bar{q} = 0.5$ bids $-\varepsilon = 0.001$

N	Mixed Nash	Scenarios Approach
6	0.3543	0.3543
7	0.3560	0.3560
8	0.3586	0.3586
9	0.3503	0.3503
10	0.3469	0.3469
11	0.3477	0.3477
12	0.3453	0.3447
13	0.3415	0.3415
14	0.3477	0.3477
15	0.3468	0.3468

Table 2.13: $\bar{q} = 0.5$ bids $+\varepsilon = 0.001$

N	Mixed Nash	Scenarios Approach
6	0.3537	0.3523
7	0.3552	0.3546
8	0.3580	0.3574
9	0.3496	0.3492
10	0.3460	0.3447
11	0.3471	0.3468
12	0.3448	0.3432
13	0.3452	0.3445
14	0.3468	0.3455
15	0.3436	0.3419

Table 2.14: $\bar{q} = 0.5$ and probability perturbation $\varepsilon = 0.01$

We can see from table (2.12) and (2.13) that the payoffs changes in no more than 0.5%, when we do small perturbation on the bids.

On the other hand, doing a perturbation of 0.01 to all non zero probabilities changes the payoffs in no more than 0.5%.

Thefore the problem is stable under small perturbations.

2.4.1 Simulations with small resistances

Now we considered the ISO problem with resistance.

Just like in Chapter 1, we considered a small resistance $r = 10^{-3}$ and we obtain the following results:

N	Mixed Nash	Scenarios Approach
6	0.3516	0.3516
7	0.3501	0.3501
8	0.3596	0.3596
9	0.3486	0.3486
10	0.3585	0.3580

Table 2.15: $r = 10^{-3}$

We can see that even with small resistance the payoffs given by the scenarios approach is really similar to the Nash equilibrium aproach, when the probabilities used for the other player are similar to their mixed Nash equilibrium ones.

Since in reality resistances are small we can make an approximation of that problem by solving the one with resistance equal to zero just like in chapter 1.

Conclusion

The study of energy markets is complex. The same problem can be seen from different points of view. One of the main ones is from the point of view of game theory, where the goal is to find Nash equilibria [13] between the firms. The problem with this approach is that although it allows us from the theoretical point of view, to proof the existence of such equilibria, and therefore, existence of the optima of the bilevel problem, in practice it is difficult and expensive to find them. These difficulties come mainly from the growth in the number of possible combinations of strategies by discretizing more finely or by adding more players which translates into solving the lower level problem for all new combinations and saving the payoff matrices (which for more than two players are tensors), so even leaving aside the computation time of all combinations, which can be very high, a large amount of RAM is required to solve medium-sized problems. We managed to find routines to solve the problem of the lower level quickly and efficiently for the quadratic and piecewise linear case, therefore the only limitation to solve for medium problems was the amount of ram memory required. Algorithms to solve nash equilibria take exponential time in the worst case. In practice, the Lemke - Howson algorithm and its variants find equilibrium in polynomial time so in general it was not a problem for a 2 player game. When adding more players, it was necessary to use another algorithm, in the literature there are not many algorithms that solve Nash equilibria for more than 2 players. The majority is based on formulating the problem as a fixed point. The most significant difference between these is that there are a couple of more recent algorithms that are written in order to facilitate parallelism and thus increase speed, maintaining the basis of solving the fixed point problem.

On chapter 2, we studied the problem from the Bilevel point of view. The main difference of this procedure with respect to the calculation of Nash equilibria given the payoff matrices, is that the problem is solved as a whole and not in two successive stages. This allows us to solve them faster and consume less memory. Classical techniques were used to transform the bilevel problem to a single level one in the case of piecewise linear and quadratic bids, which can be applied because the hypotheses of the model assured us the uniqueness of the solution of the problem of the lower level and equivalence between the bilevel problem and its single level formulation. For this formulation different possible procedures were presented and those considered the best were performed. For small problems, it could be solved efficiently by seeing all the combinations thanks to the algorithms and solutions implemented for the follower problem in the piecewise linear and quadratic case. For medium-sized problems, it was also possible to solve them using the previous way, however, a penalty method was presented to find the solution in the piecewise linear case, this allowed us to divide the problem into smaller problems, which, as stated above, are quick to solve, progress was made in proving the convergence of this method, extending the result of the linear case presented in [14]. For the quadratic case the analytical solution of the problem with out resistance is used and the problem is solved as a single level. In all cases, the implemented methods deliver solutions in less than a minute, while using general standard methods, they even took hours in delivering the same solution.

It was seen that the first approach is better from the theoretical point of view, but from the numerical point of view, it is much more expensive, so that if you can obtain reliable estimates of the strategies of the other players, using an approach such as the one from Chapter 2, allows to solve the problem in instances that we could not under the first approach and obtain an expected payoff very close to the optimal problem using mixed strategies

It was seen that doing a perturbation on the capacities by 0.2% caused changes in the expected payments between 0.1% and 1%. Perturbing the bids by 0.1% changed the expected payoffs between 0.1% and 0.5% as well as when the probability for each scenario was perturbed. Thefore the problem is quite stable under perturbations in both piecewise linear and quadratic case.

In the next work, we want to apply these ideas to a more general version of the problem, where there are renewable energies, this means that a risk term must be added and that the capacities associated with such generators are not fixed. One way to do it is by considering as a scenario not only the bid but the capacity. We would also like to apply the scenario approach using real data.

Bibliography

[1] *Foundations of Bilevel Programming.* Kluwer Academic Publishers, 2002.

[2] Ravindra K. Ahuja and James B. Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001.

[3] G. Anandalingam and D. J. White. A solution method for the linear static stackelberg problem using penalty functions. *IEEE Transactions on Automatic Control*, 35(10):1170–1173, Oct 1990.

[4] D. Aussel, P. Bendotti, and M. Pištěk. Nash equilibrium in a pay-as-bid electricity market: Part 1 – existence and characterization. *Optimization*, 66(6):1013–1025, 2017.

[5] D. Aussel, P. Bendotti, and M. Pištěk. Nash equilibrium in a pay-as-bid electricity market part 2 - best response of a producer. *Optimization*, 66(6):1027–1053, 2017.

[6] David Avis, Gabriel D. Rosenberg, Rahul Savani, and Bernhard von Stengel. Enumeration of nash equilibria for two-player games. *Economic Theory*, 42(1):9–37, Jan 2010.

[7] J. F. Bard. Some properties of the bilevel programming problem. *Journal of Optimization Theory and Applications*, 68(2):371–378, Feb 1991.

[8] Ventosa M. Rivier M.-Ramos A. Baíllo, A. Optimal offering strategies for generation companies operating in electricity spot markets. 2004.

[9] Stephen Boyd. *Convex Optimization.* Cambridge University Press New York, NY, USA, 2004.

[10] B. Chatterjee. An optimization formulation to compute nash equilibrium in finite games. 2009.

[11] S. Dempe. *Bilevel optimization: theory, algorithms and applications.* (2018).

[12] D.Fudenberg and J.Tirole. *Game Theory.* MIT Press, 1991.

[13] Juan Escobar and Alejandro Jofré. Equilibrium analysis of electricity auctions. 2008.

[14] Barroso L. A. Candal-D. Simonetti L. Fampa, M. Bilevel optimization applied to strategic pricing in competitive electricity markets. 2007.

[15] Springer-Verlag H.v. Stackelberg, Marktform und Gleichgewicht. The theory of the market economy. 1952.

[16] Garud Iyengar and Wanmo Kang. Inverse conic programming with applications. *Operations Research Letters*, 33(3):319 – 330, 2005.

[17] Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12-part-1):1608–1622, 1998.

[18] C. Lemke and J. Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423, 1964.

[19] J. A. Mirrlees. The theory of moral hazard and unobservable behaviour: Part i. *The Review of Economic Studies*, 66(1):3–21, 1999.

[20] J. V. Outrata. On the numerical solution of a class of stackelberg problems. *ZOR Zeitschrift Operations Research Methods and Models of Operations Research*, 34(4):255–277, July 1990.

[21] Jiří V. Outrata. A note on the usage of nondifferentiable exact penalties in some special optimization problems. *Kybernetika*, 24(4):251–258, 1988.

[22] M. Pilecka. Combined reformulation of bilevel programming problems. 2011.

[23] Amin Saberi and Yu Wu. *Lemke-Howson Algorithm, Lecture 4*. MSE 334 Computation of Equilibria, Stanford, 2009.

[24] Bernhard Von Stengel. Chapter 45 computing equilibria for two-person games. volume 3 of *Handbook of Game Theory with Economic Applications*, pages 1723 – 1759. Elsevier, 2002.

[25] Harish Vaish and C. M. Shetty. The bilinear programming problem. *Naval Research Logistics Quarterly*, 23(2):303–309, 1976.

[26] Robert Wilson. Architecture of power markets. *Econometrica*, 70(4):1299–1340, 2002.

[27] J. J. Ye and D. L. Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, January 1995.

[28] Jianzhong Zhang and Chengxian Xu. Inverse optimization for linearly constrained convex separable programming problems. *European Journal of Operational Research*, 200(3):671 – 679, 2010.