UNIVERSIDADE FEDERAL DO AMAZONAS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA INSTITUCIONAL DE INICIAÇÃO CIENTÍFICA

Jogos em Grafos de Ações e Equilíbrios de Nash

Bolsista: Daniel Prado Mendes de Mello - CNPq

MANAUS
2014

UNIVERSIDADE FEDERAL DO AMAZONAS

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

DEPARTAMENTO DE APOIO À PESQUISA

PROGRAMA INSTITUCIONAL DE INICIAÇÃO CIENTÍFICA

RELATÓRIO FINAL
Jogos em Grafos de Ações e Equilíbrios de Nash

Bolsista: Daniel Prado Mendes de Mello - CNPq

Orientador: Prof° Dr. Mário Salvatierra Júnior

MANAUS
2014

Resumo

Este trabalho trata de uma importante área da matemática: A Teoria dos Jogos, cuja a ênfase é a análise de como modelar fenômenos que podem ser observados quando dois ou mais agentes de decisão interagem entre si buscando sempre otimizar suas decisões. Nos concentraremos na elaboração de algoritmos e implementações computacionais para a solução de problemas envolvendo a teoria dos jogos, tendo como foco o equilíbrio de Nash em jogos de soma zero. Os tópicos de teoria dos jogos serão abordados apenas de forma enunciativa, não nos interessando demonstrações de teoremas.

Palavras-chave: Teoria dos Jogos. Equilíbrio de Nash. Decisão.

Abstract

This work addresses an important area of mathematics: The Game Theory, whose focus is the analysis of how to model phenomena that can be observed when two or more decision agents interacting always seeking to optimize their decisions. We will focus on the development of algorithms and implementations computational for solving problems involving game theory, focusing on Nash equilibria in zero sum games. The topics of game theory will be covered only from an expository way, no matter the proof of theorems.

Keywords: Game Theory. Nash equilibria. Decision.

Sumário

Introdução **6**

0.1 Objetivo geral 7

0.2 Objetivo específico 7

1 Fundamentos Teóricos **8**

1.1 Definição de jogo 8

1.2 Soluções de um jogo 10

2 Resultados Auxiliares **12**

2.1 Jogos de soma constante com dois jogadores em estratégias puras 12

2.2 Equilíbrio de Nash em estratégias puras 13

3 Resultado Principal **14**

3.1 Estratégias mistas 14

3.2 Soluções em estratégias mistas 15

3.3 Jogos de soma constante com dois jogadores em estratégias mistas 15

3.4 Equilíbrio de Nash em estratégias mistas 17

3.5 Teorema minimax de Von Neumann 18

3.6 Resoluções em Matlab 23

4 Encontrando Equilíbrios De Nash Com Um Modelo De Otimização **25**

4.1 Modelo De Otimização 25

4.2 Algoritmo para calcular w(p) 26

4.3 Trabalhos Futuros 32

Conclusão **33**

Cronograma **34**

Bibliografia **35**

Introdução

A teoria dos jogos não indica como jogar um jogo, mas nos mostra as consequências quando adotamos certas estratégias no jogo. Há momentos em nossas vidas que precisamos tomar certas decisões e, com certeza, desejamos que essa seja a melhor escolha.

Porém, às vezes, nossas ações depende das ações de outros. E essa interdependência entre tomadores de decisão é a essência de um jogo.

Contudo, a estratégia de tentar encontrar uma solução para esses problemas nem semple é tão simples como imaginamos e, isso, leva a estudar métodos e teorias criadas por pessquisadores.

A motivação para este trabalho consiste em estudar os jogos que envolvem apenas dois jogadores e adquirir conhecimento e habilidade para identificar situações do cotidiano que possam ser modeladas usando a teoria estudada.

0.1 Objetivo geral

Este trabalho tem por objetivo compreender a parte teórica da teoria dos jogos. Com a compreensão da teoria deveremos adquirir conhecimento e habilidade para tentar, em trabalhos futuros, utilizá-los em problemas de tomadas de decisões.

0.2 Objetivo específico

Utilizar a parte teórica da teoria dos jogos para solucionar problemas referentes a tomadas de decisões . Entender e demonstrar teoremas fundamentais da Teoria dos Jogos.

Capítulo 1

Fundamentos Teóricos

Neste capítulo serão apresentadas algumas definições importantes deste trabalho e que foram estudadas para melhor entendimento do resultado principal deste relatório.

1.1 Definição de jogo

Um jogo é um cenário no qual são definidos jogadores e as decisões que estes podem tomar. O objetivo é estudar as relações de vantagens e desvantagens decorridas da decisão de um jogador em função das decisões dos demais jogadores. Uma vez observado o ponto de vista de cada jogador, é interessante traçar um equilíbrio de escolhas para o qual o jogo tende. O termo utilizado para a ação de cada jogador é estratégia. As vantagens e desvantagens de cada estratégia são representadas por um número real, o Pay Off. Defini-se uma função utilidade para cada estratégia como uma função que retornará o valor do Pay Off. A definição matemática construída com base na teoria dos conjuntos é a seguinte:

$$G = \{g_1, g_2, ..., g_n\}.$$

Cada jogador $g_i \in G$ possui um conjunto finito

$$S_i = \{s_{i1}, s_{i2}, ..., s_{im_i}\}$$

de opções, denominadas estratégias puras do jogador g_i $(m \geq 2)$. Um vetor

$$\mathbf{s} = (s_{1j_1}, s_{2j_2}, ..., s_{nj_n}),$$

onde s_{ij_i} é uma estratégia pura para o jogador $g_i \in G$, é denominado um perfil de estratégia pura. O conjunto de todos os perfis de estratégia pura formam, portanto, o produto cartesiano

$$S = \prod_{i=1}^{n} S_i = S_1 \times S_2 \times \cdots \times S_n$$

denominado espaço de estratégia pura do jogo. Para cada jogador $g_i \in G$, existe uma função utilidade

$$\begin{aligned} u_i : S &\longrightarrow \mathbb{R} \\ \mathbf{s} &\longmapsto u_i(\mathbf{s}) \end{aligned}$$

que associa o ganho (payoff) $u_i(s)$ do jogador g_i a cada perfil de estratégia pura $\mathbf{s} \in S$.

Exemplo 1.1.1 :(O dilema do prisioneiro)

A situação é a seguinte: dois ladrões, Pedro e Paulo, são capturados e acusados de um mesmo crime. Presos em selas separadas e sem poderem se comunicar entre si, o delegado de plantão faz a seguinte proposta: cada um pode escolher entre confessar ou negar o crime. Se nenhum deles confessar, ambos serão submetidos a uma pena de 1 ano. Se os dois confessarem, então ambos terão pena de 5 anos. Mas se um confessar e o outro negar, então o que confessou será libertado e o outro será condenado a 10 anos de prisão.

Nesta situação, temos:

$$G = \{Pedro, Paulo\},\ S_{Pedro} = \{confessar, negar\},\ S_{Paulo} = \{confessar, negar\},$$

$$S = \{(confessar, confessar), (confessar, negar), (negar, confessar), (negar, negar)\}$$

As duas funções utilidade

$$u_{Pedro} : S \longrightarrow \mathbb{R} \text{ e } u_{Paulo} : S \longrightarrow \mathbb{R}$$

são dadas por

$$u_{Pedro} = (confessar, confessar) = -5,\ u_{Pedro} = (confessar, negar) = 0,$$
$$u_{Pedro} = (negar, confessar) = -10,\ u_{Pedro} = (negar, negar) = -1$$

(que representam os ganhos (payoffs) de Pedro) e

$$u_{Paulo}(confessar, confessar) = -5,\ u_{Paulo}(confessar, negar) = -10,$$
$$u_{Paulo}(negar, confessar) = 0,\ u_{Paulo}(negar, negar) = -1$$

(que representam os ganhos (payoffs) de Paulo).

Uma maneira conveniente de apresentar os payoffs para um jogo estático de dois jogadores é por meio de uma matriz de payoff. Cada coluna corresponde a uma estratégia de um jogador, denominado jogador coluna, e cada linha corresponde a uma estratégia do outro jogador, denominado jogador linha. Cada entrada, A_{ij}, na matriz contém o payoff obtido quando o jogador linha escolhe a estratégia i e o jogador coluna, a estratégia j. Nossa convenção será listar os payoff da seguinte maneira: (payoff do jogador linha, payoff do jogador coluna).

No dilema do prisioneiro a matriz de payoff é dada por:

		Paulo	
		$Confessar$	*Negar*
$Pedro$	$Confessar$	(-5,-5)	(0,-10)
	Negar	(-10,-0)	(-1,-1)

e o jogador linha é Pedro e o jogador coluna é Paulo. Aqui, os payoffs foram definidos como números menores ou iguais a zero, pois desta maneira, minimizar o tempo de cadeia é equivalente a maximizar o payoff.

1.2 Soluções de um jogo

A solução de um jogo é a determinação ou previsão do perfil de estratégia para o qual este tende, com base no ponto de vista de cada jogador. Existem vários conceitos diferentes de solução. Neste tópico, apresentaremos os dois conceitos mais comuns: dominância e equilíbrio de Nash com destaque para o segundo.

Para determinar a solução do dilema do prisioneiro, ou seja, qual o provável perfil de estratégias que acabará entrando em vigor, precisamos analisar os pontos de vista de cada jogador. No dilema do prisioneiro, sabemos que ambos os jogadores desejam minimizar o tempo de cadeia. Analisando o ponto de vista de cada jogador, vemos a maneira como cada um raciocina, pondera os resultados e, por fim, toma sua decisão. No ponto de vista de Pedro:

"Duas coisas podem acontecer: Paulo pode confessar ou Paulo pode negar. Se Paulo confessar, então é melhor para mim confessar também. Se Paulo não confessar, então eu fico livre se eu confessar. Em qualquer um dos casos, é melhor para mim confessar. Então eu confessarei."

Se analisarmos agora o jogo do ponto de vista de Paulo, podemos aplicar a mesma linha de raciocínio e concluir que Paulo também irá confessar. Assim, ambos confessarão e ficarão presos por 5 anos.

Em termos da teoria dos jogos, dizemos que os dois jogadores possuem uma estratégia dominante, isto é, uma estratégia em especial sempre retorna o melhor resultado, não importando qual estratégia seja escolhida pelo jogador adversário. Neste caso, confessar prevalece sobre as demais estratégias para qualquer configuração de resultados. Este conceito será explicado com mais detalhes a seguir.

Dominância

Iremos discutir perfis de estratégia na qual apenas a estratégia de um único jogador $g_i \in G$ irá variar, enquanto que as estratégias de seus oponentes permanecerão fixas. Denote por

$$\mathbf{s_{-i}} = (s_{1j_1}, ..., s_{(i-1)j_{i-1}}, s_{(i+1)j_{i+1}}, ..., s_{nj_n}) \ \in \ S_{-i} = S_1 \times \cdots \times S_{i-1} \times S_{i+1} \times \cdots \times S_n$$

uma escolha de estratégia para todos os jogadores, menos o jogador g_i. Desta maneira, um perfil de estratégia pode ser convenientemente denotado por

$$\mathbf{s} = (s_{ij_i}, \mathbf{s_{-i}}) = (s_{1j_1}, ..., s_{(i-1)j_{i-1}}, s_{ij_i}, s_{(i+1)j_{i+1}}, ..., s_{nj_n}).$$

Definição 1.2.1 (Estratégia Pura Estritamente Dominada) Uma estratégia pura $s_{ik} \in S_i$ do jogador $g_i \in G$ é <u>estritamente dominada</u> pela estratégia $s_{ik'} \in S_i$ se

$$u_i(s_{ik'}, \mathbf{s_{-i}}) > u_i(s_{ik}, \mathbf{s_{-i}}),$$

para todo $\mathbf{s_{-i}} \in S_i$. A estratégia $s_{ik} \in S_i$ é <u>fracamente dominada</u> pela estratégia $s_{ik'} \in S_i$ se

$$u_i(s_{ik'}, \mathbf{s_{-i}}) \geq u_i(s_{ik}, \mathbf{s_i}),$$

para todo $\mathbf{s_{-i}} \in S_i$.

Equilíbrio de Nash

Equilíbrio de Nash de um jogo é um ponto onde cada jogador não tem incentivo de mudar sua estratégia se os demais jogadores não o fizerem.

Definição 1.2.2 (Equilíbrio de Nash) Dizemos que um perfil de estratégia

$$\mathbf{s}^* = (s_1^*, ..., s_{(i-1)}^*, s_i^*, s_{(i+1)}^*, ..., s_n^*) \in S$$

é um equilíbrio de Nash se

$$u_i(s_i^*, \mathbf{s}_{-\mathbf{i}}^*) \geq u_i(s_{ij_i}, \mathbf{s}_{-\mathbf{i}}^*)$$

para todo $i = 1, ..., n$ e $j_i = 1, ..., m_i \geq 2$.

Exemplo 1.2.1 No jogo da perfuração de poços de petróleo (exemplo 1.1.1) , o perfil de estratégia (largo, largo) é o único equilíbrio de Nash. De fato, se uma das empresas perfurá poços largo e a outra não, aquela que não perfurou poços largo terá prejuízo de 1 milhão ao invés de obter um lucro de 1 milhão.

Exemplo 1.2.2 No dilema do prisioneiro (exemplo 1.1.2) o perfil de estratégia (confessar,confessar) é um equilíbrio de Nash. De fato: se um prisioneiro confessar e o outro não, aquele que não confessou fica preso na cadeia 10 anos, ao invés de 5 anos, se tivesse confessado. Além desse perfil, não existe outros equilíbrios de Nash.

Exemplo 1.2.3 Na batalha dos sexos (exemplo 1.1.3), os perfis de estratégia (futebol, futebol) e (cinema, cinema) são os únicos equilíbrio de Nash do jogo.

Capítulo 2

Resultados Auxiliares

Este capítulo apresenta importantes resultados que servirá como base para a demonstração e o entendimento do resultado principal deste relatório.

2.1 Jogos de soma constante com dois jogadores em estratégias puras

Definição 2.1.1 Um jogo de soma constante com dois jogadores é um jogo com dois jogadores, comumente denominados jogador linha e jogador coluna, com estratégias

$$S_{jogador\ linha} = \{1, ..., m\}$$

e

$$S_{jogador\ coluna} = \{1, ..., n\}$$

e matriz de $payoffs$

		jogador coluna			
		1	2	$\cdots$	n
	1	(a_{11}, b_{11})	(a_{12}, b_{12})	$\cdots$	(a_{1n}, b_{1n})
$jogador$	2	(a_{21}, b_{21})	(a_{22}, b_{22})	$\cdots$	(a_{2n}, b_{2n})
$linha$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	m	(a_{m1}, b_{m1})	(a_{m2}, b_{m2})	$\cdots$	(a_{mn}, b_{mn})

satisfazendo $a_{ij} + b_{ij} = c$, $\forall\ i = 1, ..., m$ e $j = 1, ..., n$. No caso particular em que a constante c é zero, dizemos que o jogo tem soma zero.

Um exemplo de jogo de soma constante é dado pela seguinte situação:

Exemplo 2.1.1 Dois jogadores exibem, ao mesmo tempo, a moeda que cada um esconde em sua mão. Se ambas as moedas apresentam cara ou coroa, o segundo jogador dá sua moeda para o primeiro. Se uma das moedas apresenta cara, enquanto a outra apresenta coroa, é a vez do primeiro jogador dá sua moeda para o segundo. Esse jogo se encontra representado por sua matriz de payoffs dada abaixo.

		g_2	
		s_{21}	s_{22}
g_1	s_{11}	(+1,-1)	(-1,+1)
	s_{12}	(-1,+1)	(+1,-1)

Neste exemplo o jogador linha é g_1 e o jogador coluna é g_2.

2.2 Equilíbrio de Nash em estratégias puras

Definição 2.2.1 (Ponto de sela) Dizemos que um elemento a_{ij} de uma matriz A é um ponto de sela da matriz A se ele for simultaneamente um mínimo em sua linha e um máximo em sua coluna, isto é, se

$$a_{ij} \leq a_{il} \text{ para todo } l = 1, ..., n \text{ e}$$

$$a_{ij} \geq a_{kj} \text{ para todo } k = 1, ..., m.$$

Teorema 2.2.1 O elemento a_{ij} é um ponto de sela da matriz A se, e somente se, o par (i, j) é um equilíbrio de Nash em estratégias puras para o jogo.

Teorema 2.2.2 Se a_{ij} e a_{rs} são dois pontos de sela da matriz A, então a_{is} e a_{rj} também são pontos de sela da matriz A e

$$a_{ij} = a_{rs} = a_{is} = a_{rj}.$$

Teorema 2.2.3 Para toda matriz A, tem-se $v_c(A) \geq v_l(A)$.

Teorema 2.2.4 Uma matriz A tem um ponto de sela se, e somente se, $v_l(A) = v_c(A)$.

Corolário 2.2.1 Um jogo de dois jogadores com soma constante definido pela matriz de payoffs A do jogador linha tem um equilíbrio de Nash em estratégias puras se, e somente se,

$$v_l(A) = v_c(A)$$

.

Capítulo 3

Resultado Principal

Como existem jogos que não possuem equilíbrios de Nash em estratégias puras dedicaremos este capítulo as *estratégias mistas*. Uma alternativa para estes casos é a de considerar o jogo do ponto de vista probabilístico, isto é, ao invés de escolher um perfil de estratégia pura, o jogador deve escolher uma distribuição de probabilidade sobre suas estratégias puras.

3.1 Estratégias mistas

Definição 3.1.1 Uma estratégia mista $\mathbf{p_i}$ para o jogador $g_i \in G$ é uma distribuição de probabilidades sobre o conjunto S_i de estratégias puras do jogador, isto é, $\mathbf{p_i}$ é um elemento do conjunto

$$\Delta_{m_i} = \{(x_1, x_2, ..., x_{m_i}) \in \mathbb{R}^{m_i} \mid x_1 \geq 0, ..., x_{m_i} \geq 0 \text{ e } \sum_{k=1}^{m_i} x_k = 1\}$$

Assim, se $\mathbf{p_i} = (p_{i1}, p_{i2}, ..., p_{im_i})$, então $p_{i1} \geq 0, p_{i2} \geq 0, ..., p_{im_i} \geq 0$ e $\sum_{k=1}^{m_i} p_{ik} = 1$.

Os pontos extremos (vértices) de Δ_{m_i}, isto é, os pontos da forma

$$e_1 = (1, 0, ..., 0, 0), e_2 = (0, 1, ..., 0, 0), ..., e_{m_i} = (0, 0, ..., 0, 1)$$

dão, respectivamente, probabilidade 1 às estratégias puras $s_{i1}, s_{i2}, ..., s_{im_i}$. Desta maneira, podemos considerar a distribuição de probabilidade e_k como a estratégia mista que representa a estratégia pura s_{ik} do jogador g_i.

O espaço de todos os perfis de estratégia mista é o produto cartesiano

$$\Delta = \Delta_{m_1} \times \Delta_{m_2} \times \cdots \times \Delta_{m_n},$$

denominado espaço de estratégia mista. Como no caso de estratégias puras, usaremos a notação $\mathbf{p}_{-i}$ para representar as estratégias de todos os jogadores, com exceção do jogador g_i.

Cada perfil de estratégia mista $\mathbf{p} = (\mathbf{p_1}, ..., \mathbf{p_n}) \in \Delta$ determina um payoff esperado, uma média dos payoffs ponderada pelas distribuições de probabilidades $\mathbf{p_1}, ..., \mathbf{p_n}$. Mais precisamente, se

$\mathbf{p} = (\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_n})$

$= (p_{11}, p_{12}, ..., p_{1m_1}; p_{21}, p_{22}, ..., p_{2m_2}; ...; p_{n1}, p_{n2}, ..., p_{nm_n})$, então

$$EU_i(\mathrm{p}) = \sum_{j_1}^{m_1} \sum_{j_2}^{m_2} \cdots \sum_{j_n}^{m_n} \left(\prod_{k=1}^{n} p_{kj_k} u_i(s_{1j_1}, s_{2j_2}, ..., s_{nj_n}) \right).$$

3.2 Soluções em estratégias mistas

Todos os critérios básicos para soluções de jogos em estratégias puras podem ser estendidos para estratégias mistas, mais nos limitaremos, neste trabalho, apenas em equilíbrio de Nash.

Equilíbrio de Nash

Definição 3.2.1 (Equilíbrio de Nash) Dizemos que um perfil de estratégia mista

$$\mathbf{p}^* = (\mathbf{p}_1^*, \mathbf{p}_2^*, ..., \mathbf{p}_n^*) \in \Delta = \Delta_{m_1} \times \Delta_{m_2} \times \cdots \times \Delta_{m_n}$$

é um equilíbrio de Nash se

$$EU_i(\mathbf{p}_i^*, \mathbf{p}_{-i}^*) \geq EU_i(\mathbf{p}, \mathbf{p}_{-i}^*)$$

para todo $\mathbf{p} \in \Delta_{m_i}$, isto é, nenhum jogador sente motivação de trocar sua estratégia mista se os demais jogadores não o fizerem.

No dilema do prisioneiro (exemplo 1.1.2), o perfil de estratégia mista

$$\mathbf{p}^* = (\mathbf{p}_1^*, \mathbf{p}_2^*) = (1, 0; 1, 0)$$

é um equilíbrio de Nash, pois

$$EU_1(\mathbf{p}, \mathbf{p}_2^*) = EU_1(p, 1-p; 1, 0) = 5p - 10 \leq 5 = EU_1(1, 0; 1, 0) = EU_1(\mathbf{p}_1^*, \mathbf{p}_2^*)$$

para todo $\mathbf{p} = (p, 1-p) \in \Delta_2$ e

$$EU_2(\mathbf{p}_1^*, \mathbf{q}) = EU_2(1, 0; q, 1-q) = 5q - 10 \leq -5 = EU_2(1, 0; 1, 0) = EU_2(\mathbf{p}_1^*, \mathbf{p}_2^*)$$

para todo $\mathbf{q} = (q, 1-q) \in \Delta_2$. Observe que este equilíbrio corresponde ao equilíbrio em estratégias puras $\mathbf{s}^* = (confessar, confessar)$.

3.3 Jogos de soma constante com dois jogadores em estratégias mistas

Definição 3.3.1 Um jogo de soma constante com dois jogadores é um jogo com dois jogadores, comumente denominados jogador linha e jogador coluna, com estratégias

$$S_{jogador\ linha} = \{1, ..., m\}$$

e

$$S_{jogador\ coluna} = \{1, ..., n\}$$

e matriz de $payoffs$

jogador coluna

		1	2	$\cdots$	n
	1	(a_{11}, b_{11})	(a_{12}, b_{12})	$\cdots$	(a_{1n}, b_{1n})
$jogador$	2	(a_{21}, b_{21})	(a_{22}, b_{22})	$\cdots$	(a_{2n}, b_{2n})
$linha$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	m	(a_{m1}, b_{m1})	(a_{m2}, b_{m2})	$\cdots$	(a_{mn}, b_{mn})

satisfazendo $a_{ij} + b_{ij} = c$, $\forall\ i = 1, ..., m$ e $j = 1, ..., n$. No caso particular em que a constante c é zero, dizemos que o jogo tem soma zero.

Em termos de estratégias mistas, se $\mathbf{p} = (p_1, .., p_m) \in \Delta_m$ é uma distribuição de probabilidades para as estratégias puras do jogador linha e $\mathbf{q} = (q_1, ..., q_n) \in \Delta_n$ é uma distribuíção de probabilidades para tratégias puras do jogador coluna, então o *payoff* esperado para o jogador linha é

$$EU_l(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{m}\sum_{j=1}^{n} p_i a_{ij} q_j = \begin{bmatrix} p_1 & p_2 & \cdots & p_m \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n, \end{bmatrix}$$

isto é

$$EU_l(\mathbf{p}, \mathbf{q}) = \mathbf{p^T} A\mathbf{q},\ \text{com } A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Analogamente, o payoff esperado para o jogador coluna é dado por

$$EU_c(\mathbf{p}, \mathbf{q}) = \mathbf{p^T} B\mathbf{q} \text{ com } B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

Uma vez que o jogo tem soma constante, vemos que

$$A + B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} c & c & \cdots & c \\ c & c & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \cdots & c \end{bmatrix},$$

isto é,

$$A + B = C = \begin{bmatrix} c & c & \cdots & c \\ c & c & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \cdots & c \end{bmatrix} = c \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} = c\bigstar$$

onde $\bigstar$ denota a matriz $m \times n$ formada com 1 em todas as suas entradas. Sendo assim, é fácil de ver que

$$EU_c(\mathbf{p}, \mathbf{q}) = \mathbf{p^T} B\mathbf{q} = \mathbf{p^T}(c\bigstar - A)\mathbf{q} = c\mathbf{p^T}\bigstar\mathbf{q} - \mathbf{p^T} A\mathbf{q} = c - EU_l(\mathbf{p}, \mathbf{q})$$

onde, na última igualdade, usamos que $\mathbf{p^T}\bigstar\mathbf{q} = 1$, pois $\mathbf{p}$ e $\mathbf{q}$ são distribuição de probabilidade e, por isto,

$$\sum_{i=1}^{m} \mathbf{p}_i = 1 \text{ e } \sum_{j=1}^{n} \mathbf{q}_j = 1.$$

3.4 Equilíbrio de Nash em estratégias mistas

Defina

$$v_l(A) = \max_{\mathbf{p}\in\mathbf{\Delta_m}} \min_{\mathbf{q}\in\Delta_n} \mathbf{p^T} A\mathbf{q} \text{ e } v_c(A) = \min_{\mathbf{q}\in\mathbf{\Delta_n}} \max_{\mathbf{p}\in\Delta_m} \mathbf{p^T} A\mathbf{q}.$$

Teorema 3.4.1 Para toda matriz A, tem-se $v_c(A) \geq v_l(A)$.

Demonstração: Temos que para todo $\mathbf{p} \in \Delta_m$,

$$\mathbf{p}^T A\mathbf{q} \geq \min_{y\in\Delta_n} \mathbf{p}^T A\mathbf{y}$$

Assim,

$$\max_{\mathbf{p}\in\mathbf{\Delta_m}} \mathbf{p}^T A\mathbf{q} \geq \max_{\mathbf{p}\in\Delta_m} \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^T A\mathbf{y} = v_l(A).$$

Consequetemente,

$$v_c(A) = \min_{\mathbf{q}\in\Delta_n} \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q} \geq \max_{\mathbf{p}\in\Delta_m} \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^T A\mathbf{y} = v_l(A).\blacksquare$$

O próximo teorema caracteriza a existência de equilíbrios de Nash em estratégias mistas em termos das funcões v_l e v_c .

Teorema 3.4.2 Um perfil de estratégia mista $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash de um jogo de dois jogadores com soma constante definido pela matriz de payoffs A do jogador linha se, e somente se,

$$v_l(A) = v_c(A) = \mathbf{p}^{*T} A\mathbf{q}^*$$

.

Demonstração:

$(\Longrightarrow)$ Se $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash, então

$$\mathbf{p}^{*T} A\mathbf{q}^* = EU_l(\mathbf{p}^*, \mathbf{q}^*) \geq EU_l(\mathbf{p}, \mathbf{q}^*) = \mathbf{p}^T A\mathbf{q}^*,$$

para todo $\mathbf{p} \in \Delta_m$. Em particular,

$$\mathbf{p}^{*T} A\mathbf{q}^* = \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q}^* \geq \min_{\mathbf{q}\in\Delta_n} \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{y} = v_c(A).$$

Vale também que

$$\mathbf{p}^{*T} A\mathbf{q}^* = c - EU_c(\mathbf{p}^*, \mathbf{q}^*) \leq c - EU_c(\mathbf{p}^*, \mathbf{q}) = \mathbf{p}^{*T} A\mathbf{q},$$

para todo $\mathbf{q} \in \Delta_n$. Em particular

$$\mathbf{p}^{*T} A\mathbf{q}^* = \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^{*T} A\mathbf{q} \leq \max_{\mathbf{x}\in\Delta_m} \min_{\mathbf{q}\in\Delta_n} \mathbf{x}^T A\mathbf{q} = v_l(A).$$

Desta maneira, $v_l(A) \geq v_c(A)$. Como, pelo teorema anterior, $v_l(A) \leq v_c(A)$, concluímos que $v_l(A) = v_c(A)$.

($\Longleftarrow$) Como $v_l(A) = \max_{\mathbf{p}\in\Delta_m} \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^T A\mathbf{q}$, existe $\mathbf{p}^* \in \mathbf{\Delta_m}$ tal que

$$v_l(A) = \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^{*T} A\mathbf{q}.$$

Analogamente, como $v_c(A) = \min_{\mathbf{q}\in\Delta_n} \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q}$, existe $\mathbf{q}^* \in \Delta_m$ tal que

$$v_c(A) = \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q}^* .$$

Uma vez que, por hipótese, $v_l(A) = v_c(A)$, temos que

$$\min_{\mathbf{q}\in\Delta_n} \mathbf{p}^{*T} A\mathbf{q} = v_l(A) = v_c(A) = \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q}^* .$$

Afirmamos que $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash do jogo. Com efeito,

$$EU_l(\mathbf{p}^*, \mathbf{q}^*) = \mathbf{p}^{*T} A\mathbf{q}^* \geq \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^{*T} A\mathbf{q} = \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q}^* \geq \mathbf{x}^T A\mathbf{q}^* = EU_l(\mathbf{x}, \mathbf{q}^*),$$

para todo $\mathbf{x} \in \Delta_m$. Por outro lado,

$$EU_c(\mathbf{p}^*, \mathbf{q}^*) = c - \mathbf{p}^{*T} A\mathbf{q}^* \geq c - \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q}^* = c - \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^{*T} A\mathbf{q} \geq c - \mathbf{p}^{*T} A\mathbf{y} = EU_c(\mathbf{p}^*, \mathbf{y}),$$

para todo $\mathbf{y} \in \Delta_n$. Desta maneira, $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash do jogo. ■

3.5 Teorema minimax de Von Neumann

O próximo teorema estabelece que, para jogos de dois jogadores com soma zero, $v_l(A) = v_c(A)$ sempre. Sendo assim, pelo teorema 3.4.2, segue-se que, para esta classe de jogos, sempre existe pelo menos um equilíbrio de Nash em estratégias mistas.

Teorema 3.5.1 (**minimax de Von Neumann**) Para todo jogo de soma zero com dois jogadores, representado pela matriz de payoffs A do jogador linha, sempre existe um perfil de estratégia mista $(\mathbf{p}^*, \mathbf{q}^*) \in \Delta_m \times \Delta_n$ satisfazendo

$$v_l(A) = \max_{\mathbf{p}\in\Delta_m} \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^T A\mathbf{q} = \mathbf{p}^{*T} A\mathbf{q}^* = \min_{\mathbf{q}\in\Delta_n} \max_{\mathbf{p}\in\Delta_m} \mathbf{p}^T A\mathbf{q} = v_c(A).$$

Em particular, $(\mathbf{p}^*, \mathbf{q}^*)$ e um equilíbrio de Nash do jogo.

Daremos uma demonstração deste teorema minimax de Von Neumann usando o teorema de dualidade da teoria de programação linear. Lembramos que um problema de programação linear é um problema de otimização com função objetivo e restrições lineares:

(**problema primal**)

$$\text{maximizar } \mathbf{b}^T\mathbf{y}$$

$$\text{sujeito a } A\mathbf{y} \leq c,$$

$$\mathbf{y} \geq 0,$$

onde as desigualdades devem ser interpretadas componente a componente. A cada problema de programação linear (problema primal) podemos associar um outro problema de otimização (problema dual):

(**problema dual**)

$$\text{minimizar } \mathbf{c}^T\mathbf{x}$$

$$\text{sujeito a } \mathbf{x}^T A \geq \mathbf{b}^T,$$

$$\mathbf{x} \geq 0.$$

Teorema 3.5.2 (**DA DUALIDADE EM PROGRAMAÇÃO**)

(a) O problema primal possui uma solução se, e somente se, o problema dual possui uma solução.

(b) Se $\mathbf{y}^*$ é solução do problema primal e $\mathbf{x}^*$ é solução do problema dual, então $\mathbf{c^T x}^* = \mathbf{b}^T\mathbf{y}^*$.

Demonstração do teorema minimax: 3.5.1 sem perda de generalidade, podemos assumir que todas as entradas da matriz de payoffs A do jogador linha são positivas. Caso contrário, basta substituir A por $\overline{A} = A + D$ e $B = -A$ por $\overline{B} = -D + B$, onde $D = d\bigstar$, com $d > \max\limits_{1\leq i\leq m} \max\limits_{1\leq j\leq n} |a_{ij}|$. Observe que $\overline{A} + \overline{B} = 0$ (isto é, o jogo definido pelas matrizes $\overline{A}$ e $\overline{B}$ tem soma zero) e que $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash para o jogo definido pela matriz A se, e somente se, $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash para o jogo definido pela matriz $\overline{A}$.

Sejam $\mathbf{c} = (1, 1, ..., 1)^T$ e $\mathbf{b} = (1, 1, ..., 1)^T$. Considere os problemas de programação linear:

(**problema primal**)

$$\text{maximizar } \mathbf{b^T y}$$

$$\text{sujeito a } A\mathbf{y} \leq c,$$

$$\mathbf{y} \geq 0,$$

(**problema dual**)

$$\text{minimizar} \mathbf{c}^T\mathbf{x}$$

$$\text{sujeito a } \mathbf{x}^T A \geq \mathbf{b}^T,$$

$$\mathbf{x} \geq 0.$$

Passo 1: o problema dual possui uma solução.

Como $A > 0$, o conjunto admissível $X = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x}^T A \geq \mathbf{b}^T \text{ e } \mathbf{x} \geq 0\}$ é não vazio. Por outro lado, como $\mathbf{c} = (1, 1, ..., 1)^T$, a função objetivo do problema é escrita como $\mathbf{x} = (x_1, x_2, ..., x_m) \longmapsto \mathbf{c}^T\mathbf{x} = x_1 + x_2 + \cdots + x_m$. Assim, o problema dual consiste em encontrar o ponto do conjunto X mais próximo da origem segundo a norma da soma $|| \cdot ||_1$, um problema que certamente possui uma solução pois, se $\mathbf{p} \in$ X, então podemos "compactificar"o conjunto admissível incluindo a restrição $||\mathbf{x}||_1 \leq ||\mathbf{p}||_1$ e, com isso, podemos usar o teorema de Weierstrass para garantir a existência de um mínimo.

Passo 2: construção do equilíbrio de Nash. Dado que o problema dual possui uma solução, pelo teorema de dualidade, o problema primal também possui. Mais ainda: se $\mathbf{x}^*$ é solução do problema primal e $\mathbf{y}^*$ é solução do problema dual, então

$$\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*.$$

Seja $\theta = \mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$ (que é > 0, pois $(0, 0, ..., 0)$ não é admissível) e defina

$$\mathbf{p}^* = \frac{\mathbf{x}^*}{\theta} \qquad \text{e} \qquad \mathbf{q}^* = \frac{\mathbf{y}^*}{\theta}.$$

Afirmamos que $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash do jogo. Com efeito: claramente $\mathbf{p} \in \Delta_m$ e $\mathbf{q} \in \Delta_n$, pois $\mathbf{p}^* \geq 0$ (já que $\mathbf{x}^* \geq 0$ e $\theta > 0$) $\mathbf{q}^* \geq 0$ (já que $\mathbf{y}^* \geq 0$ e $\theta > 0$),

$$\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} \frac{\mathbf{x_i}^*}{\theta} = \frac{\mathbf{c^T x}^*}{\theta} = \frac{\theta}{\theta} = 1 \quad \text{e} \quad \sum_{j=1}^{n} q_j = \sum_{j=1}^{n} \frac{\mathbf{y_j}^*}{\theta} = \frac{\mathbf{b^T y}^*}{\theta} = \frac{\theta}{\theta} = 1$$

Agora, como $\mathbf{x}^{*T} A \geq \mathbf{b}^T$, temos que para todo $\mathbf{q} \in \Delta_n$,

$$\mathbf{x}^{*T} A\mathbf{q} \geq \mathbf{b}^T\mathbf{q} = \sum_{j=1}^{n} \mathbf{q}_j = 1$$

Mas $\mathbf{x}^* = \dfrac{\mathbf{p}^*}{\theta}$. Desta maneira, $\mathbf{p}^* A\mathbf{q} \geq \theta = \mathbf{p}^* A\mathbf{q}^*$, para todo $\mathbf{q} \in \Delta_n$. Consequentemente,

$$EU_c(\mathbf{p}^*, \mathbf{q}^*) = -\mathbf{p}^{*T} A\mathbf{q}^* \geq -\mathbf{p}^{*T} A\mathbf{q} = EU_c(\mathbf{p}^*, \mathbf{q})$$

para todo $\mathbf{q} \in \Delta_n$. Mostramos então que o jogador coluna não pode aumentar o seu payoff esperado trocando $\mathbf{q}^*$ por $\mathbf{q}$, se o jogador linha mantiver a escolha $\mathbf{p}^*$. Analogamente, como $A\mathbf{y} \leq c$, temos que para todo $\mathbf{p} \in \Delta_m$,

$$\mathbf{p}^T A\mathbf{y}^* \leq \mathbf{p}^T\mathbf{c} = \sum_{i=1}^{m} \mathbf{p}_i = 1$$

. Mas $\mathbf{y}^* = \dfrac{\mathbf{q}^*}{\theta}$. Desta maneira, $\mathbf{p}A\mathbf{q}^* \leq \theta = \mathbf{p}^{*T} A\mathbf{q}^*$, para todo $\mathbf{p} \in \Delta_m$. Consequentemente,

$$EU_l(\mathbf{p}^*, \mathbf{q}^*) = \mathbf{p}^{*T} A\mathbf{q}^* \geq \mathbf{p}^T A\mathbf{q}^* = EU_l(\mathbf{p}, \mathbf{q}^*),$$

para todo $\mathbf{p} \in \Delta_m$. Mostramos então que o jogador linha não pode aumentar o seu payoff esperado trocando $\mathbf{p}^*$ por $\mathbf{p}$, se o jogador coluna mantiver a escolha $\mathbf{q}^*$. Concluímos, portanto, que $(\mathbf{p}^*, \mathbf{q}^*)$ é um equilíbrio de Nash do jogo. ■

Além de estabelecer a existência de equilíbrios de Nash, a demonstração que demos sugere uma maneira de calculá-los: resolvendo-se dois problemas de programação linear.

Exemplo 3.5.1 O governo deseja vacinar seus cidadãos contra um certo vírus da gripe. Este vírus possui dois sorotipos, sendo que é desconhecida a proporção na qual os dois sorotipos ocorrem na população do vírus. Foram desenvolvidas duas vacinas onde a eficácia da vacina 1 é de 85% contra o sorotipo 1 e de 70% contra o sorotipo 2. A eficácia da vacina 2 é de 60% contra o sorotipo 1 e de 90% contra o sorotipo 2. Qual política de vacinação deveria ser tomada pelo governo?

Esta situação pode ser modelada como um jogo de soma zero com dois jogadores, onde o jogador linha L (o governo) deseja fazer a compensação (a fração dos cidadãos resistentes ao vírus) o maior possível e o jogador coluna C (o vírus) deseja fazer a compensação a menor possível. A matriz de payoffs é a seguinte:

vírus

sorotipo 1 sorotipo 2

$gogerno$	$vacina1$	$(85/100, -85/100)$	$(70/100, -70/100)$
	$vacina2$	$(60/100, -60/100)$	$(90/100, -90/100)$

Para encontrar um equilíbrio de Nash, devemos resolver os seguintes problemas de programação linear

(problema primal)

$$\begin{aligned} &\text{maximizar } y_1 + y_2 \\ &\text{sujeito a } \begin{bmatrix} 85/100 & 70/100 \\ 60/100 & 90/100 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \\ &\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \end{aligned}$$

(problema dual)

$$\begin{aligned} &\text{minimizar } x_1 + x_2 \\ &\text{sujeito a } \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 85/100 & 70/100 \\ 60/100 & 90/100 \end{bmatrix} \geq \begin{bmatrix} 1 & 1 \end{bmatrix}, \\ &\begin{bmatrix} x_1 & x_2 \end{bmatrix} \geq \begin{bmatrix} 0 & 0 \end{bmatrix}, \end{aligned}$$

isto é

(problema primal)

$$\begin{aligned} &\text{maximizar } y_1 + y_2 \\ &\text{sujeito a } 17y_1 + 14y_2 \leq 20, \\ &6y_1 + 9y_2 \leq 10, \\ &y_1 \geq 0, \\ &y_2 \geq 0, \end{aligned}$$

(problema dual)

$$\begin{aligned} &\text{minimizar } x_1 + x_2 \\ &\text{sujeito a } 17x_1 + 12x_2 \geq 20, \\ &7x_1 + 9x_2 \geq 10, \\ &x_1 \geq 0, \\ &x_2 \geq 0. \end{aligned}$$

A solução do problema dual é $\mathbf{x}^* = (20/23, 10/23)$ (figura 3.1) e a solução do problema primal é $\mathbf{y}^* = (40/69, 50/69)$, com

$$\theta = x_1^* + x_2^* = y_1^* + y_2^* = \frac{30}{23}.$$

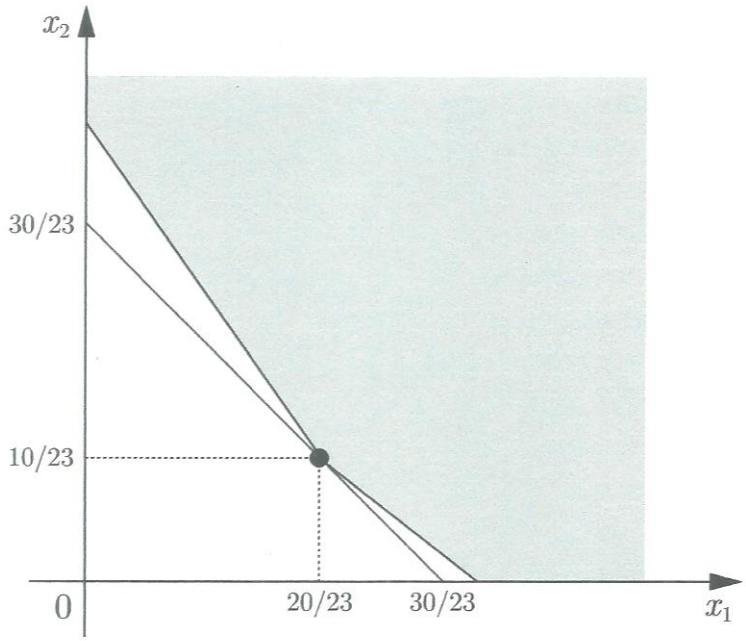

Figura 3.1: Solução do problema dual.

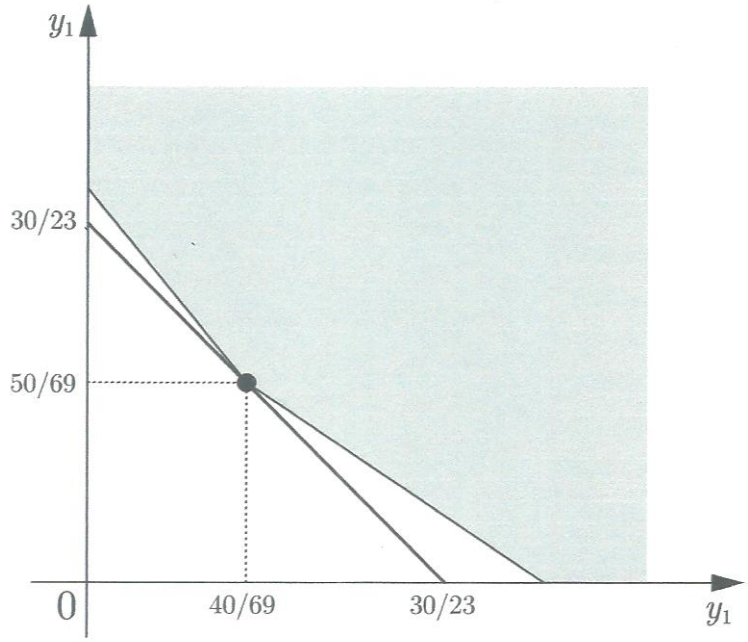

Figura 3.2: Solução do problema primal.

Desta maneira, o único equilíbrio de Nash para o problema é dado pelo ponto $(\mathbf{p}^*, \mathbf{q}^*)$, onde

$$\mathbf{p}^* = \frac{\mathbf{x}^*}{\theta} = \left(\frac{2}{3}, \frac{1}{3}\right) \text{ e } \mathbf{q}^* = \frac{\mathbf{y}^*}{\theta} = \left(\frac{4}{9}, \frac{5}{9}\right). \blacksquare$$

3.6 Resoluções em Matlab

Agora serão implementadas soluções para jogos de soma zero através de códigos realizados em Matlab. Para que isso seja possível, é necessário adaptar algumas das relações do problema primal e do problema dual para o contexto do Matlab. Lá, a função linprog só oferece a opção de minimização sujeito a $A\mathbf{y} \leq b$. Isso significa que para o problema primal teremos que multiplicar $\mathbf{b}^T$ por -1, já que minimizar $-\mathbf{b}^T\mathbf{y}$ é equivalente a maximizar $\mathbf{b}^T\mathbf{y}$. Para o problema dual, faremos a matriz de payoff's igual a sua transposta negativa e multiplicaremos $\mathbf{c}^T$ por -1, de forma a compensarmos o sinal de desigualdade trocado no Matlab. O comando para a solução do problema das vacinações do exemplo 3.5.1 é o seguinte:

$c = [1 \quad 1]$; % vetor da desigualdade

$b = [-1 \quad -1]$; % b'y a ser maximizado

$lb = [0 \quad 0]$; % lower bound: limite inferior definido anteriormente como zero

$A = [0.85 \quad 0.7; \quad 0.6 \quad 0.9]$ % matriz de payoff's

$y = linprog(b, A, c, [\quad], [\quad], lb, [\quad]$ % vetor solução y

$A = -A'$ % transposta de A multiplicada por -1;

$x = linprog(c, A, b, [\quad], [\quad], lb, [\quad])$ %vetor solução x e troca de b com c

$t = sum(x)$; % soma dos elementos de x ou y, que é igual a θ.

$p = x/t$

$q = y/t$

O resultado é, naturalmente, o mesmo obtido anteriormente na seção 3.5.

Para jogos com maior número de estratégias por jogador, é necessário uma pequena adaptação. Seguem agora exemplos de resoluções que ilustram essa situação. Partiremos das matrizes de pay offs, não sendo necessária a criação de uma problema para a contextualização na vida real.

Exemplo 3.6.1 Encontrar o equilíbrio de nash para a seguinte matriz de pay off's

$$\begin{bmatrix} 5 & -1 & 4 & 7 \\ -3 & 1 & -2 & -5 \\ 1 & -4 & 8 & 3 \end{bmatrix}$$

$A = [5 \ -1 \quad 4 \quad 7; \ -3 \quad 1 \ -2 \ -5; \quad 1 \ -4 \quad 8 \quad 3]$

$c = [1 \quad 1 \quad 1]$;

$b = [-1 \ -1 \ -1 \ -1]$;

$lby = [0 \quad 0 \quad 0 \quad 0]$; %note que neste caso, é necessária definição de um novo limite inferior para cada uma das soluções

$y = linprog(b, A, c, [\quad], [\quad], lby, [\quad])$

$A = -A'$;

$lbx = [0 \quad 0 \quad 0]$;

$x = linprog(c, A, b, [\quad], [\quad], lbx, [\quad])$

$t = sum(x)$;

$p = y/t$

$q = x/t$

Obtemos o seguinte resultado

y= $\begin{bmatrix} 0 \\ 6 \\ 0 \\ 1 \end{bmatrix}$

x= $\begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}$

p= $\begin{bmatrix} 0.0000 \\ 0.8571 \\ 0.0000 \\ 0.1429 \end{bmatrix}$

q= $\begin{bmatrix} 0.4286 \\ 0.5714 \\ 0.0000 \end{bmatrix}$

Exemplo 3.6.2 Encontrar o equilíbrio de nash para a seguinte matriz de pay off's

$$\begin{bmatrix} -1 & 1 & 2 & 0 \\ 4 & -2 & -3 & 2 \\ 0 & 3 & 1 & -1 \end{bmatrix}$$

Novamente, seguindo o mesmo código do exercício anterior e somente alterando os elementos da matriz obtemos o resultado:

y= $\begin{bmatrix} 0.2 \\ 1.2 \\ 0.0 \\ 1.3 \end{bmatrix}$

x= $\begin{bmatrix} 1.8 \\ 0.7 \\ 0.2 \end{bmatrix}$

p= $\begin{bmatrix} 0.0741 \\ 0.4444 \\ 0.0000 \\ 0.4815 \end{bmatrix}$

q= $\begin{bmatrix} 0.6667 \\ 0.2593 \\ 0.0741 \end{bmatrix}$

Que é a solução esperada.

Capítulo 4

Encontrando Equilíbrios De Nash Com Um Modelo De Otimização

4.1 Modelo De Otimização

Considere o jogo finito n , cujo o número de jogadores é representado pelo conjunto $N = \{1, ..., n\}$. Cada jogador possui um conjunto de estratégias $S_i = \{s_{i1}, ..., s_{iJ_i}\}$, com um número J_i de estratégias puras.

Definindo-se Δ_i como o conjunto de valores de probabilidade sobre os elementos de S_i, cujo os elementos são da forma $p_i : S_i \longrightarrow \mathbb{R}$ e $\sum_{s_{ij} \in S_i} p_i(s_{ij}) = 1$ com $p_i(s_{ij}) \geq 0$ para todo $s_{ij} \in S_i$. Nós usaremos a notação p_{ij} = $p_i(s_{ij})$. Escreveremos $\Delta = \prod_{i \in N} \Delta_i$ e faremos $J = \sum_{i \in N} J_i$. Nós denotaremos estes pontos em Δ por $p = (p_1, ..., p_n)$, onde $p_i = (p_{i1}, ..., p_{iJi}) \in \Delta_i$. Usaremos, de forma abusiva, o termo s_{ij} para representar a estratégia cuja a probabilidade p_{ij} é igual a 1. A notação $p = (s_{ij}, p_{-i})$ simbolizará a estratégia na qual o jogador i assumirá uma estratégia pura enquanto os demais jogadores utilizam quaisquer componentes de p.

A função utilidade u_i, que foi definida como tendo domínio S, será extendida para que tenha Δ como domínio através da definição $u_i(p) = \sum_{s \in S} p(s) u_i(s)$ aonde $p(s) = \prod_{i \in N} p_i(s_i)$. Um vetor $p = (p_1, ..., p_n) \in \Delta$ será um equilíbrio de Nash se, e somente se, para todo elemento $i \in N$, e para todo $p_i \in \Delta_i$, $u_i(p_i, p_{-i}) \leq u_i(p)$.

Criaremos uma função $v : \Delta \longrightarrow \mathbb{R}$ definida como:

$$v(p) = \sum_{i \in N} \sum_{s_{ij} \in S_i} \{max\,[u_i(s_{ij}, p_{-i}) - u_i(p), 0]\}$$

O modelo de otimização consiste em minimizar v(p) com o seguinte modelo de otimização:

$$\begin{array}{ll} min & v(p) \\ s.a. & \sum_j p_{ij} = 1 \\ & p_{ij} \geq 0. \end{array}$$

Para fins de computação, nós imporemos essas condições sob a forma de funções de penalidade. Sendo assim, v(p) será redefinida em uma nova função, a qual denominaremos w(p):

$$w(p) = v(p) + \sum_j min\,[p_{ij}, 0]^2 + \sum_{i \in N} (1 - \sum_j p_{ij})^2$$

Assim, é equivalente minimizarmos $w(p)$ sem restrições.

4.2 Algoritmo para calcular w(p)

Nesta seção será apresentado um algoritmo escrito na linguagem C para o cálculo de w(p) em um jogo descrito por uma matriz 2 por 2, ou seja, com 2 jogadores e 2 estratégias mistas para cada. As principais rotinas que levam aos resultados de v(p) e de w(p) são as que retornam valores de funções utilidade: M, M1X e M2X. O algoritmo para o cálculo delas foi realizado com base nas funções de payoff esperado para jogador linha e coluna desenvolvidos na seção 3.3. O código é o que se segue:

```
#include <stdio.h>


float max(float a, float b){

  if (a>=b)
    return a;
  else
    return b;
}

float min(float a, float b){

  if (a<=b)
    return a;
  else
    return b;
}

imprime_matriz(float mat[2][2]){

  int a,b;

  for (a=0; a<=1 ; a++){

    for(b=0; b<=1; b++)
    printf("%f ", mat[a][b]);
    printf("\n");
  }
}

imprime_vet(float vet[2]){

  int a;

  for (a=0; a<=1; a++)

    printf("%f ", vet[a]);
}

//produto escalar vetor x vetor
float pe_vet(float vet1[2], float vet2[2]){

  float a=0;
  int i;

  for(i=0; i<2; i++)

    a+=vet1[i]*vet2[i];
```

```
    return a;
}



//produto escalar matriz x vetor
void pe_vetmat(float mat[2][2], float vet[2], float vet1[2]){

  int i , j;

  vet1[0]=0;
  vet1[1]=0;

  for (i=0; i<2; i++){

          for (j=0; j<2; j++)

          vet1[i]=mat[i][j]*vet[j]+vet1[i];

     }

}
//lê vetor
void scan_vet(float vet[2]){

  int i;

  for(i=0; i<=1; i++)

    scanf("%f", &vet[i]);
}

//lê matriz
void scan_mat(float mat[2][2]){

  int a, b;

  for (a=0; a<=1 ; a++)

    for(b=0; b<=1; b++)

      scanf("%f", &mat[a][b]);
}

//calcula função utilidade M
float M(float payoff[2][2], float p1[2], float p2[2]){


float M, vetaux[2]={};

  pe_vetmat(payoff, p2, vetaux);
  M=pe_vet(p1, vetaux);

  return M;
}
```

```
//calcula M1(S11, P-1) e  M1(S12, P-1)
void  M1X(float  payoff[2][2],  float  p2[2],  float  vetM1[2]  ){

  float  vetaux2[2],  vetaux[2]={},  M11,  M12;


  //calcula  M1(S11, P-1)
  vetaux2[0]=1;
  vetaux2[1]=0;

  pe_vetmat(payoff,  p2,  vetaux);
  M11=pe_vet(vetaux2,  vetaux);


  //calcula M1(S12, P-1)
  vetaux2[0]=0;
  vetaux2[1]=1;
  vetaux[0]=0;
  vetaux[1]=0;


      pe_vetmat(payoff,  p2,  vetaux);
  M12=pe_vet(vetaux2,  vetaux);


  vetM1[0]=M11;
  vetM1[1]=M12;
}


//calcula M2(S21, P-2) e  M2(S22, P-2)
void  M2X(float  payoff2[2][2],  float  p1[2],  float  vetM2[2]  ){

  float  vetaux2[2]={},  vetaux[2]={},  M21,  M22;

  //calcula e imprime M2(S21, P-2)
  vetaux2[0]=1;
  vetaux2[1]=0;

  pe_vetmat(payoff2,  vetaux2,  vetaux);
  M21=pe_vet(p1,  vetaux);


  //calcula e imprime M2(S22, P-2)
  vetaux2[0]=0;
  vetaux2[1]=1;
  vetaux[0]=0;
  vetaux[1]=0;


      pe_vetmat(payoff2,   vetaux2,   vetaux);
  M22=pe_vet(p1,  vetaux);


  vetM2[0]=M21;
  vetM2[1]=M22;
}
```

```
/*Aqui vão ser lidas as entradas do programa e retornada a funcao v.
Aqui tambem serão realizadas chamadas a todas as rotinas acima.*/

float v(float p1[2], float p2[2]){


  float vp, aux, payoff[2][2]={}, payoff2[2][2]={},
   M1, M1s[2]={}, M2, M2s[2]={}, vetv[4]={};
  int a;

  //lê payoff do jogador 1
  printf("\n\nInsira os elementos da matriz de payoff ");
  printf("em estrategias puras do jogador 1,\n");
  printf("linha, por linha.\n");
  scan_mat(payoff);
  printf("matriz digitada: \n\n");
  imprime_matriz(payoff);


  //lê payoff do jogador 2
  printf("\n\nInsira os elementos da matriz de payoff ");
  printf("em estrategias puras do jogador 2,\n");
  printf("linha, por linha.\n");
  scan_mat(payoff2);
  printf("matriz digitada: \n\n");
  imprime_matriz(payoff2);


  //lê vetor de prob do jogador 1
  printf("\n\nInsira os elementos do vetor de probabilidades de p1\n");
  scan_vet(p1);
  printf("\nvetor digitado:\n");
  imprime_vet(p1);


  //lê vetor de prob do jogador 2
  printf("\n\nInsira os elementos do vetor de probabilidades de p2\n");
  scan_vet(p2);
  printf("\nvetor digitado:\n");
  imprime_vet(p2);


  //calcula funcao utilidade do jogador 1
  M1 = M(payoff, p1, p2);
  printf("\n_____________________________");
  printf("\n\nM1: %f", M1);


  //calcula funcao utilidade do jogador 2
  M2 = M(payoff2, p1, p2);
  printf("\nM2: %f", M2);
  printf("\n_____________________________");


  //Imprime um vetor que contém M1(S11, P-1) e  M1(S12, P-1)
  printf("\n\nM1(S11, P-1) e  M1(S12, P-1):");
```

```
  M1X(payoff,  p2,  M1s);
  imprime_vet(M1s);



  //Imprime um vetor que contém M2(S21, P-2) e  M2(S22, P-2)
  printf("\n\nM2(S21, P-2) e  M2(S22, P-2): ");
  M2X(payoff2,  p1,  M2s);

  imprime_vet(M2s);
  printf("\n_____________________________");


  /*Cálculo dos elementos entre chaves no somatório da função v(p).

 Utiliza-se os elementos dos dois vetores impressos acima e
 os valores das funções utilidade M1 e M2.

 Cada um dos 4 elementos do somatorio é armazenado no vetor vetv[4].
 Posteriormente, eles sao somados, obtendo-se v(p)*/

  for  (a=0;  a<=1;  a++){

    aux=(M1s[a]-M1);
    vetv[a]=max(aux,  0);
    vetv[a]*=vetv[a];
  }


  for  (a=0;  a<=1;  a++){

    aux=(M2s[a]-M2);
    vetv[a+2]=max(aux,  0);
    vetv[a+2]*=vetv[a+2];
  }



  /*Aqui são imprimidos os elementos do vetor vetv, que são
 os elementos do somatório de vp.

 Além disso eles são somados para a obtenção e retorno de v(p)*/

  vp=0;
  printf("\n\nelementos do somatório de v(p): ");

  for  (a=0;  a<=3;  a++){


    printf("%f ",  vetv[a]);
    vp+=vetv[a];
  }
  return  vp;
}


//retorna a função w a partir de v e dos vetores de probabilidade
float  w  (float  vp,  float  p1[2],  float  p2[2]){
```

```
  int  a,  b  ;
  float  wp,  aux,  somap1,  somap2;

  //primeira parte da função, que antecede os somatórios, composta pela função v

  wp=vp;

  //primeiro somatório
  for  (a=0;  a<=1;  a++){

    somap1+=p1[a];
    aux  =  min(p1[a],  0);
    aux*=aux;
    wp+=aux;
  }

  for  (a=0;  a<=1;  a++){

    somap2+=p2[a];
    aux  =  min(p2[a],  0);
    aux*=aux;
    wp+=aux;
  }

  //segundo somatório
  aux  =  1-somap1;
  aux*=aux;
  wp+=aux;


  aux  =  1-somap2;
  aux*=aux;
  wp+=aux;


  return  wp;
}


int  main(){

  /*p1 e p2 são os vetores de probabilidades, que vo ser lidos na rotina v.
 São iniciados aqui na Main, para que fiquem visíveis tanto para v quanto para w
  float  vp,  wp,  p1[2]={},  p2[2]={};



  vp  =  v(p1,  p2);
  printf("\n\nv(p)=%f",  vp);


  wp  =  w(vp,  p1,  p2);
  printf("\nw(p)=%f",  wp);
  printf("\n");

  return  0;
}
```

4.3 Trabalhos Futuros

Por conta do término do período para a conclusão do trabalho, alguns tópicos não puderam ser abordados, embora a parte mais importante do estudo tenha sido realizada. Tais tópicos ficam como sujestão para serem abordados em uma possível continuação do trabalho. Dentre estes, destaca-se a implementação de um método de otimização para encontrar os minimizadores da função w. Um método adequado é o de gradientes conjugados, pois este necessita somente do gradiente da função w. No artigo A Liapunov Function For Nash Equilibria (McKelvery, 1998) é dado explicitamente o gradiente da função w, mas é possível também aproximar o grandiente usando diferenças finitas. Deve-se notar que nem todos os pontos mínimos na função w são mínimos globais. Existem mínimos locais e deve-se estar atento para a possibilidade de ser levado até um. Por isso, seja qual método for adotado, deve ser realizada uma verificação para determinar se o minimizador é global ou não.

Conclusão

Com os estudos feitos neste trabalho foi possível entender a teoria dos jogos com a finalidade de prever os movimentos dos outros jogadores, sejam eles concorrentes ou aliados, através dessa teoria os jogadores se posicionam da melhor forma para obter o resultado desejado.

O estudo nos revelou também que a teoria dos jogos nos proporciona o entendimento da lógica na hora de uma decisão e nos ajuda a responder se é possível haver colaboração entre os jogadores, em quais circunstâncias o mais racional é não colaborar e quais estratégias devem ser adotadas para garantir a colaboração entre os jogadores.

Foi estudada a maneira de computar o equilíbrio de Nash para a situação de jogos de soma zero. Os algoritmos desenvolvidos em Matlab foram apresentados como uma maneira alternativa de se obter o resultado final.

Finalmente, foi estudada uma função, que se minimizada, leva ao equilíbrio de Nash. Esta função foi definida de forma que pode ser desenvolvido um algoritmo escrito em C para o cálculo da mesma. Esta etapa ajudou a entender o problema de forma computacional e caracterizou um passo inicial para a criação de algoritmos que resolvam jogos mais complexos.

Cronograma

1ª ETAPA - 2º SEMESTRE - 2013

Atividades	Ago	Set	Out	Nov	Dez
Leitura de artigos sobre teoria dos jogos	X				
Apresentação de seminário para o orientador		X			
Preparação da apresentação parcial			X		
Apresentação oral parcial				X	
Elaboração do Relatório Parcial					X

2ª ETAPA - 1º SEMESTRE - 2014

Atividades	Jam	Fer	Mar	Abr	Mai	Jun	Jul
Leitura de artigos sobre teoria dos jogos	X						
Apresentação de seminário para o orientador		X					
Preparação da apresentação final			X				
Preparação da apresentação final				X			
Elaboração do Relatório Final					X		
Elaboração do Relatório Final						X	
Elaboração do Relatório Final							X
Apresentação oral final							X

Referências Bibliográficas

[1] SARTINI, B.A., GARBUGIO, G., BORTOLOSSI, H. J., SANTOS, P. A. e BARRETO, L. Uma Introdução à Teoria dos Jogos. 1ª ed. Salvador: II Bienal da SBM, UFB, 2004.

[2] BIERMAN, H. S. e FERNANDEZ, L. Teoria dos jogos. 2ª ed. São Paulo: Pearson Prentice Hall, 2011.

[3] THIE, P. R., KEOUGH, G.E. An Introduction to Linear Programming and Game Theory. 3ª ed. Jhon Wiley & Sons, 2008.