

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

**Teoria de Jogos Nebulosos na Resolução de
Problemas de Decisão e Conflito de Interesses**

Wanessa Machado do Amaral
Orientador: Prof. Dr. Fernando Gomide

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação.**

Banca Examinadora

Fernando GomideDCA/FEEC/Unicamp
Laécio Carvalho de Barros.....IMECC/Unicamp
Paulo Augusto Valente Ferreira.....DT/FEEC/Unicamp

Campinas, SP
Julho/2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE -
UNICAMP

Am13t Amaral, Wanessa Machado do
Teoria de jogos nebulosos na resolução de problemas de
decisão e conflito de interesses / Wanessa Machado do
Amaral. --Campinas, SP: [s.n.], 2007.

Orientador: Fernando Antonio Campos Gomide
Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Teoria dos jogos. 2. Conjuntos difusos. 3.
Programação linear. 4. Programação não-linear. 5.
Computação evolutiva. I. Gomide, Fernando Antonio
Campos. II. Universidade Estadual de Campinas. Faculdade
de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Fuzzy game theory to solve decision making and conflicting
problems

Palavras-chave em Inglês: Game theory, Fuzzy sets, Linear programming,
non-linear programming, Evolutionary computation

Área de concentração: Engenharia de Computação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Laécio Carvalho de Barros e Paulo Augusto Valente
Ferreira

Data da defesa: 03/07/2007

Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Wanessa Machado do Amaral

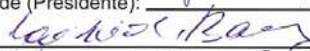
Data da Defesa: 3 de julho de 2007

Título da Tese: "Teoria de Jogos Nebulosos na Resolução de Problemas de Decisão e Conflito de Interesses"

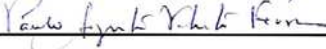
Prof. Dr. Fernando Antônio Campos Gomide (Presidente):



Prof. Dr. Laécio Carvalho de Barros:



Prof. Dr. Paulo Augusto Valente Ferreira:



Resumo

Palavras-chave: Teoria de jogos, conjuntos nebulosos, programação linear, programação não linear, computação evolutiva.

A teoria de jogos é um ramo da teoria da decisão que modela e trata matematicamente situações de conflito de interesses entre entidades, onde o objetivo principal é escolher a melhor estratégia para cada uma delas, ou seja, aquela que se traduz em equilíbrio.

Existem inúmeras áreas em que a teoria de jogos é utilizada. Uma das principais é a microeconomia, onde se aborda questões relativas ao comportamento de empresas e indústrias no mercado competitivo. A teoria de jogos é utilizada para encontrar a estratégia ótima para empresas com objetivos antagônicos, como exige o mercado.

No entanto, os dados dos problemas reais nem sempre são precisos. A teoria de conjuntos nebulosos introduz flexibilidade na formulação desses problemas, pois permite a consideração de parâmetros imprecisos nos modelos.

Esse trabalho aborda a teoria de jogos nebulosos. Estratégias de equilíbrio são analisadas e métodos computacionais desenvolvidos para a resolução dos modelos. É proposto um método baseado em computação evolutiva para obter soluções de equilíbrio de jogos nebulosos. Além disso propõe-se também um método baseado em α -cortes e no algoritmo de decomposição para a solução dos modelos bilineares associados a jogos nebulosos de soma não zero. Exemplos de aplicações são apresentados para ilustrar o potencial prático da teoria de jogos nebulosos.

Abstract

Keywords: Game theory, fuzzy sets, linear programming, non-linear programming, evolutionary computation.

Game theory is a branch of applied mathematics whose aim is to model and study decision making in conflicting situations. In these situations, the main goal is to choose the best strategy for all the players in the game, that is, to find the equilibrium solutions. Game theory can be defined as the study of how self-interested entities interact and make decisions.

There are many applications of game theory in different areas. One of the main applications is in microeconomy, where situations of conflict between companies exist and there is a need to find the optimal strategies in that situation.

In practice however, model parameters are imprecise. Fuzzy set theory allows modeling flexibility because imprecise data can be treated using fuzzy models.

This work concerns Fuzzy Game Theory. Equilibrium strategies are studied and computational methods developed to solve fuzzy game problems. A new method to solve fuzzy games using evolutionary computation is introduced. A method based on α -cuts and on a decomposition algorithm to solve bilinear models also presented to solve fuzzy non zero-sum games. Algorithms were implemented and applications examples are discussed to illustrate the usefulness of fuzzy games in practice.

*Dedico este trabalho a meu
filho, meu esposo, meus pais e meu
irmão, que estão sempre presentes em
minha vida.*

Agradecimentos

Ao meu Prof. Dr. Fernando Antonio Campos Gomide, sou grata pela orientação.

Aos colegas de pós-graduação, pelas críticas e sugestões.

Aos funcionários, pelo auxílio nas questões administrativas.

Aos amigos, por todo o incentivo e companhia, que muito ajudou no decorrer dos últimos anos.

Ao meu esposo Enzo, pelo amor, carinho e apoio incondicional.

A minha família. Meus pais, Regina e José Luis, e meu irmão Wagner, pelo apoio durante esta jornada.

Ao Felipe, meu filho e razão da minha vida.

A Deus, sempre.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xvii
Notação	xix
Introdução	1
1. Teoria de conjuntos nebulosos	9
1.1 Introdução	9
1.2 Conjuntos nebulosos	9
1.3 Teorema da representação	14
1.4 Princípio da extensão	16
1.5 Números nebulosos	19
1.6 Ordenação de números nebulosos	22
1.6.1 Método de Yager	22
1.6.2 Método de α -cortes	23
1.6.3 Método baseado no princípio da extensão	24
1.6.4 Exemplos	24
1.7 Resumo	28
2. Teoria de jogos	29
2.1 Introdução	29
2.2 Elementos de um jogo	29
2.3 Jogos de soma zero com dois jogadores	31
2.4 Jogos de soma não zero com dois jogadores	35
2.4.1 Dilema do prisioneiro	37
2.5 Jogos cooperativos com k Jogadores	39
2.6 Jogos nebulosos	41
2.7 Resumo	45
3. Algoritmos	47
3.1 Introdução	47
3.2 Jogos nebulosos de soma zero	47
3.2.1 Modelo de Campos	48
3.2.2 Modelo de Maeda	54
3.2.3 Modelo de ordenação	59
3.2.4 Modelo de α -cortes	64
3.2.5 Exemplos	65
3.2.6 Comentários sobre os modelos	81
3.2.7 Aplicação em microeconomia	82
3.3 Jogos nebulosos cooperativos com k jogadores	84
3.3.1 Definições	85

3.3.2 Método de solução.....	87
3.3.3 Aplicação à problemas de produção.....	91
3.3.4 Comentários sobre o método.....	94
3.4 Resumo.....	95
4. Algoritmo para resolução de jogos nebulosos de soma não zero	97
4.1 Introdução.....	97
4.2 Algoritmo de Konno.....	98
4.3 Método de α -Cortes.....	101
4.4 Exemplos.....	105
4.5 Aplicações.....	107
4.5.1 Estratégia de investimento preemptivo.....	107
4.5.2 Aplicação no mercado de energia.....	109
4.6 Resumo.....	112
5. Abordagem co-evolutiva para solução de jogos nebulosos	115
5.1 Introdução.....	115
5.2 Algoritmos evolutivos.....	115
5.3 Co-evolução.....	118
5.4 Descrição do algoritmo.....	118
5.4.1 Representação.....	119
5.4.2 Operadores genéticos.....	120
5.4.3 Avaliação e seleção.....	122
5.4.4 Reprodução.....	125
5.5 Implementação e resultados.....	127
5.5.1 Jogos nebulosos de soma zero.....	127
5.5.2 Jogos nebulosos de soma não zero.....	131
5.6 Comparação com os métodos clássicos.....	136
5.7 Resumo.....	137
6. Conclusão	139
6.1 Contribuições.....	139
6.2 Trabalhos futuros.....	140
Referências bibliográficas.....	141

Lista de Figuras

Fig 1.1: Exemplo de conjunto e conjunto nebuloso.	10
Fig 1.2: α -corte de A	14
Fig 1.3: Teorema da representação.....	16
Fig 1.4: Princípio de extensão clássico.....	17
Fig 1.5: Ilustração do princípio da extensão.	18
Fig 1.6: Números nebulosos triangulares	20
Fig 1.7 Número nebuloso triangular simétrico	21
Fig 1.8: Primeiro exemplo de ordenação.....	24
Fig 1.9: Segundo exemplo de ordenação.....	25
Fig 1.10: Terceiro exemplo de ordenação.....	26
Fig 1.11: Quarto exemplo de ordenação.....	26
Fig 1.12: Quinto exemplo de ordenação.	27
Fig 1.13: Sexto exemplo de ordenação.....	27
Fig 3.1 Relação entre $A(\lambda)$ e $\tilde{A}$	55
Fig 3.2: Representação da medida de possibilidade entre dois números nebulosos.....	58
Fig 3.3: Representação da medida de necessidade entre dois números nebulosos.....	58
Fig 3.4: Representação dos pontos de vista constante, otimista e pessimista.....	61
Fig 3.5: Números nebulosos	63
Fig 3.6: Comparação entre dois números nebulosos	63
Fig 3.7 Valor do jogo no exemplo 1 pelo modelo de Maeda	67
Fig 3.8 Valor do jogo para diferentes dispersões na matriz de pagamentos.....	68
Fig 3.9 Valor do jogo para matriz H assimétrica	70
Fig 3.10 Valor do jogo no exemplo 1 pelo modelo de Campos: ordenação 1, 2 e 4.	71

Fig 3.11 Valor do jogo no exemplo 1 pelo modelo de Campos: ordenação 3	72
Fig 3.12 Valor do jogo no exemplo 1 pelo modelo de Campos: ordenação 5	72
Fig 3.13 Valor do jogo no exemplo 2 pelo modelo de Campos: ordenação por 1, 2 e 476	
Fig 3.14 Valor do jogo no exemplo 2 pelo modelo de Campos: ordenação por 3	77
Fig 3.15 Valor do jogo no exemplo 2 pelo modelo de Campos: ordenação por 5	77
Fig 3.16 Valor do jogo para o exemplo 2 no modelo de Maeda.....	78
Fig 3.17: Valor da coalizão $S=\{1, 2, 3, 4,5\}$	93
Fig 3.18: Valor da coalizão $S=\{1\}$	94
Fig 5.1: Representação de um cromossomo em um GA.....	116
Fig 5.2: Cruzamento entre cromossomos binários	117
Fig 5.3: Mutação em um cromossomo binário.....	117
Fig 5.4 Representação de um cromossomo para um jogo com duas opções	120
Fig 5.5 Método de seleção por torneio	125
Fig 5.6 Indivíduos que representam soluções candidatas e a criação de indivíduos filhos.....	126
Fig 5.7. Fitness médio da população	128
Fig 5.8. Alta taxa de mutação proporciona maior diversidade.	130
Fig 5.9. Baixa taxa de mutação proporciona menor diversidade.....	131
Fig 5.10. Evolução do <i>fitness</i> médio da população em um jogo de soma não zero	133
Fig 5.11. <i>Fitness</i> do melhor indivíduo	133

Lista de Tabelas

Tab 1.1: Exemplo do principio de extensão clássica.	18
Tab 2.1: Matriz de pagamentos.....	33
Tab 2.2: Dilema do prisioneiro.....	37
Tab 3.1 Valores da função de ordenação.....	62
Tab 3.2 Valores de ordenação do ponto de vista pessimista.....	62
Tab 3.3: Valores de índice relativo	64
Tab 3.4 Matriz H assimétrica no modelo de Maeda.....	69
Tab 3.5: Exemplo 1 - resultados pelo modelo de Campos	73
Tab 3.6: Exemplo 1 - resultados pelo método de Lee-Kwang e Lee	74
Tab 3.7: Exemplo 1 – alteração na matriz de pagamentos	74
Tab 3.8: Exemplo 1, modelo de α -corte.....	75
Tab 3.9. Exemplo 2 - jogos de soma zero - modelo de Campos.....	76
Tab 3.10. Exemplo 2 - jogos de soma zero - modelo de Lee-Kwang e Lee	78
Tab 3.11: Exemplo 1, modelo de α -corte.....	78
Tab 3.12. Resultado do jogo pelos diferentes métodos de resolução.....	79
Tab 3.13. Ordenação dos resultados do jogo	80
Tab 3.14. Defuzzificação dos diferentes resultados do jogo.....	81
Tab 3.15: Opções de investimento no mercado.....	82
Tab 3.16: Recursos dos decisores	91
Tab 3.17: Valor das coalizões	92
Tab 3.18: Valor do jogo para o decisor 1	93
Tab 4.1: Resultado do Exemplo 1 para o caso clássico.	106
Tab 4.2: Resultado do Exemplo 1, para $\alpha=0.5$, com ponto inicial factível.....	106

Tab 4.3: Resultado do Exemplo 1, para $\alpha=0.5$, com ponto inicial infactível.....	106
Tab 4.4: Resultado do Exemplo 1, para diferentes valores de $\alpha=0.5$.....	107
Tab 4.5: Estratégia de investimento preemptivo	108
Tab 4.6: Saídas das decisões de produção de A e B.....	110
Tab 4.7: Preços correspondentes a cada nível de produção de A e B.....	111
Tab 4.8: Lucro, sem contratos de diferença	111
Tab 4.9: Resultados para o exemplo de mercado de energia	112
Tab 5.1 Estratégias para o jogo de soma zero (5.6).....	129
Tab 5.2 Resultado do exemplo (5.6) com outros métodos.....	129
Tab 5.3 Estratégias obtidas para o jogo de soma não zero	132

Notação

$A(x)$ – função de pertinência de um conjunto nebuloso A

A^L – limite inferior de A

A^C – núcleo de A

A^R – limite superior de A

A_α – α -corte de A

A_α^L – limite inferior do α -corte de A

A_α^R – limite superior do α -corte de A

$\tilde{a}$ – representa um número nebuloso

$\tilde{A}$ – representa uma matriz nebulosa, matriz cujos elementos são números nebulosos

$\inf(A)$ – ínfimo de A

N – conjunto de inteiros positivos

N_n - conjunto $\{1, 2, \dots, n\}$

R^m – universo real de dimensão m

sse – se, e somente se

$\sup(A)$ – supremo de A

$Supp(A)$ – suporte do conjunto nebuloso A

$v(S)$ – valor da coalizão S num jogo cooperativo

x^* – solução ótima (equilíbrio) para um problema

x^*Ay^* – valor do jogo

$\lesssim$ e $\gtrsim$ – restrições nebulosas de desigualdade

$'$ – transposto

Introdução

Motivação

Apesar da evolução significativa da teoria de decisão e sistemas de computação nas últimas décadas, inúmeros e importantes problemas ainda carecem de estudo para considerar aplicações práticas.

A teoria de decisão é uma área amplamente estudada e empregada na resolução de problemas complexos. A teoria de jogos é um ramo da teoria de decisão que trata de uma classe particular de problemas cujo cenário envolve situações de conflito de interesses entre duas ou mais entidades.

A teoria de jogos modela situações de conflito de interesse e analisa possíveis estratégias com o intuito de encontrar uma solução de equilíbrio para todas as entidades envolvidas (Basar e Olsder, 1982).

Existem inúmeras áreas em que a teoria de jogos é utilizada na prática. Uma das mais importantes é a microeconomia, onde existem diversas questões relativas ao comportamento das empresas no mercado competitivo. A teoria de jogos é uma abordagem utilizada para encontrar uma estratégia ótima para cada empresa diante de situações de competição.

Entretanto, os dados relativos a problemas de interesse prático nem sempre podem ser obtidos com precisão. A teoria de conjuntos nebulosos, introduzida por Zadeh (1965), contribui de forma importante no tratamento de imprecisões que ocorrem nos parâmetros dos modelos. Com a utilização dos conceitos de conjuntos nebulosos é possível traduzir informações lingüísticas em dados numéricos, ou equivalentemente, traduzir termos lingüísticos em funções de pertinência. Esta tradução possibilita o tratamento computacional de modelos imprecisos (Pedrycz e Gomide, 1998).

Uma das abordagens utilizadas nesse trabalho é a computação evolutiva. A computação evolutiva é uma heurística de otimização populacional baseada em mecanismos inspirados na evolução biológica. A idéia central da computação evolutiva é aplicar os princípios dos processos de evolução natural propostos por Darwin (1859) na resolução computacional de problemas complexos. Portanto, a computação evolutiva

trabalha diretamente com algoritmos inspirados na teoria da evolução e são amplamente empregados em uma variedade de problemas de otimização. Entre os principais algoritmos evolutivos podemos citar os algoritmos genéticos (Holland, 1975), as estratégias evolutivas (Rechenberg, 1973), a programação evolutiva (Fogel, 1966) e a programação genética (Koza, 1992).

Tanto a computação evolutiva quanto a teoria de conjuntos nebulosos são consideradas, junto com as redes neurais, ramos da inteligência computacional. Atualmente existem inúmeros estudos (Buckley, 1998, Delgado, 2002, Cordón *et al*, 2004) que combinam esses dois conceitos para a resolução de problemas complexos.

Revisão Bibliográfica

Os estudos em teoria de jogos teve início por volta dos anos 1940. A publicação do trabalho de J. von Neumann e O. Morgenstern (1944) chamou a atenção dos pesquisadores para essa área. Muitos outros estudos seguiram-se aos de Neumann.

Desde então a teoria de jogos vem sendo utilizada como ferramenta analítica importante para problemas de decisão de organizações e em sistemas competitivos (Harsanyi, 1977)

Com o avanço na teoria de conjuntos nebulosos introduzidos por Zadeh (1965), surgiram estudos que combinam esta teoria com a teoria de jogos. O poder de análise das situações de conflito da teoria de jogos com a flexibilidade nos parâmetros dos problemas que a teoria de conjuntos nebulosos oferece formam uma ferramenta eficaz na resolução dos problemas de decisão.

Butnariu (1978) foi o primeiro a estudar jogos não cooperativos com dois jogadores em um cenário nebuloso. Ele considerou em seus trabalhos casos em que o conjunto de estratégias dos jogadores podem ser vistos como conjuntos nebulosos. Em trabalho posterior (Butnariu, 1980) o autor examinou jogos não cooperativos com k jogadores em um cenário nebuloso e apresentou um conceito de solução de equilíbrio para esses jogos.

Buckley (1984) analisou problemas de jogos nebulosos com dois jogadores de maneira similar a Butnariu, envolvendo porem incertezas nas estratégias e múltiplos objetivos nebulosos. Nesse trabalho os jogos são formulados como problemas de decisão de

um jogador com a estratégia do outro jogador sendo uma distribuição de possibilidade de prioridades.

Billot (1992) examinou soluções de equilíbrio de jogos não cooperativos com n jogadores em seu livro.

Ponsard (1986 e 1987) estudou jogos não cooperativos com n jogadores de um ponto de vista diferente no qual ele supõe que os jogadores tem uma preferência imprecisa pelos pagamentos e as ocorrências de incertezas no pagamento do jogo.

Campos (1989) examinou problemas de jogos nebulosos não cooperativos de soma zero onde os elementos da matriz de pagamento são números nebulosos e aplicou métodos de ordenação de números nebulosos para reformular os problemas e encontrar a solução de equilíbrio. De forma similar, Lee-Kwang e Lee (1999), Maeda (2000), Maeda (2003) e Chen e Larbani (2005) também utilizam métodos de programação linear nebulosa para encontrar a solução de equilíbrio para jogos nebulosos não cooperativos de soma zero. Em Lee-Kwang e Lee (1999) porém, é proposto um método de ordenação de números nebulosos antes de se aplicar um método de programação linear. No trabalho de Maeda (2003), o autor investiga as propriedades dos jogos nebulosos e propõe um método que também resolve jogos de soma não zero. Em Chen e Larbani (2005), a nebulosidade do problema é tratada pelo método de α -cortes.

Nishizaki e Sakawa (1995) e Nishizaki e Sakawa (2000) consideram jogos nebulosos não cooperativos de soma zero com múltiplos objetivos e com números nebulosos na matriz de pagamentos.

Além de considerar ambigüidade no julgamento dos decisores com informações imprecisas, Sakawa e Nishizaki (1992) introduziram objetivos nebulosos para o pagamento de uma coalizão. O objetivo nebuloso representa o grau de satisfação que diz respeito a soma dos pagamentos dos jogadores participantes da coalizão. Para os jogos cooperativos com objetivo nebuloso, os autores definiram um conceito de solução baseado na idéia de Bellman e Zadeh (1970).

Todos os trabalhos mencionados consideram jogos não cooperativos. Bergstresser e Yu (1977) consideraram jogos multiobjetivos com n jogadores em um cenário cooperativo onde uma função característica associa subconjuntos de jogadores, uma coalizão, com um

vetor de valores reais e examinaram conceitos de solução como o núcleo utilizando estruturas de dominância.

Derks e Tijs (1986a / 1986b) definiram jogos multiobjetivos cooperativos estendendo a função característica para um mapeamento da coalizão para um conjunto de vetores de pagamento, e considerou o núcleo nos jogos. Os autores se referiram a esse tipo de jogo multiobjetivo como jogos *multi-commodity*, uma vez que eles lidam com situações onde arcos em uma rede pertencem a diferentes jogadores.

Os estudos na área da computação evolutiva envolvendo problemas de teoria dos jogos é assunto relativamente novo. O dilema do prisioneiro iterado foi primeiramente explorado utilizando-se algoritmos genéticos por Axelrod em 1984. Em seu texto, Axelrod (1984) propôs um torneio para que estudiosos e interessados no assunto apresentassem estratégias com memória das jogadas anteriores para o problema do dilema do prisioneiro. Essas estratégias foram comparadas através de jogos entre elas, e a estratégia que obteve o maior número de pontos foi declarada vencedora.

Muitos outros trabalhos seguiram-se ao de Axelrod. Andreoni e Varian (1993) consideraram o dilema do prisioneiro modificado, onde cada jogador pode pagar para o outro jogador cooperar.

Borges *et al.*(1995) trabalha com o dilema do prisioneiro iterado onde os movimentos de cooperação ou não cooperação são considerados valores nebulosos. O texto apresenta um torneio, onde além dos jogadores nebulosos, as estratégias de sucesso já conhecidas como *tit for tat* e Pavlov são apresentadas.

Goldbeck (2002) analisou a tendência de cooperação com demais jogadores cooperativos, através de uma abordagem utilizando algoritmos genéticos.

Hingston e Kendall (2004) combinam aprendizado com evolução. Foram escolhidas duas classes de populações, uma com jogadores não-adaptáveis, que aprendem a jogar entre si apenas por um processo evolutivo, não aprendendo nada durante sua fase de vida, e outra classe de população composta por jogadores adaptáveis que não herdam nada de seus parentes, e devem aprender tudo durante seu próprio período de vida. Os autores concluíram que jogadores adaptáveis são capazes de sobreviver na população em evolução em proporções significativas.

O’Riordan *et al.* (2004) apresentam resultados para o problema do dilema do prisioneiro envolvendo n jogadores. Também é incorporada a idéia de perdão, que visa evitar estratégias baseadas em respostas muito rápidas à delação, tentando restabelecer a cooperação perdendo delações anteriores. Os resultados indicam que a incorporação do perdão aumenta os níveis de cooperação.

Goh *et al.* (2005) investigaram a relação entre aprendizado e evolução, concluindo que o aprendizado dá aos jogadores uma vantagem nos estágios iniciais do jogo, enquanto que a evolução é responsável pela manutenção e preservação desta. Nesse texto os autores fazem ainda uma análise sobre aprendizado em tempo real, de onde se conclui que o aprendizado em tempo real propicia aos jogadores uma maior adaptabilidade em ambientes dinâmicos.

Em Wei-Liu *et al.* (2006) uma abordagem para encontrar um equilíbrio de Nash aproximado em um ambiente nebuloso é apresentada. Esse trabalho trata do conceito de equilíbrio de Nash nebuloso para estratégias mistas e propõe um algoritmo genético para encontrar um equilíbrio de Nash aproximado em um ambiente nebuloso.

O uso de co-evolução para o mercado de oligopólios é considerado por Chen *et al.* (2006) para analisar diferentes modelos de equilíbrio durante a interação estratégica entre agentes de mercado, incluindo Cournot, Bertrand e Stackelberg, entre outros.

Diversos outros trabalhos utilizam a teoria dos jogos como tentativa de explicar e compreender a co-evolução o que, de acordo com (Wiegand *et al.*, 2002), é uma tarefa complexa. Destacam-se os trabalhos de Ficici *et al.* (2000) e Ficici e Pollack (2000 e 2001) sobre co-evolução.

Objetivo

Esse trabalho apresenta duas abordagens para a resolução de jogos nebulosos não cooperativos. A primeira, baseada em α -cortes, utiliza modelos de programação bilinear e um algoritmo de decomposição do modelo bilinear na solução de jogos não cooperativos de soma não zero.

A segunda abordagem considera métodos de computação evolutiva, analisando uma população de soluções candidatas e simulando a co-evolução de estratégias que levem a

soluções de equilíbrio. É possível utilizar a abordagem evolutiva para resolver jogos de soma zero e de soma não zero. As principais vantagens e desvantagens observadas nessa abordagem em relação às abordagens clássicas são discutidas.

O trabalho também discute técnicas de solução propostas na literatura que trata de jogos nebulosos. Estas técnicas foram implementadas e os resultados obtidos são analisados e comparados com os métodos propostos.

Organização do trabalho

Após esta introdução, o Capítulo 1 apresenta os conceitos básicos sobre a teoria de conjuntos nebulosos (Zadeh, 1965). Nesse capítulo define-se o conceito de números nebulosos e como eles podem representar matematicamente valores de variáveis lingüísticas e quantidades imprecisas. Conceitos sobre ordenação de números nebulosos (Yager, 1981) são apresentados neste capítulo e considerados no decorrer do trabalho.

O segundo capítulo resume os principais conceitos da teoria de jogos clássica (Basar e Olsder, 1982). Serão abordados jogos não cooperativos de soma zero, de soma não zero, e jogos cooperativos.

O Capítulo 3 discute os principais métodos propostos na literatura para a solução de jogos nebulosos. Primeiramente, os jogos nebulosos não cooperativos de soma zero serão analisados e técnicas para a resolução desses problemas são apresentadas (Campos, 1989, Maeda, 2003, Lee-Kwang e Lee, 1999). Este capítulo também traça um paralelo entre os jogos nebulosos de soma zero e programação linear. Neste caso, as técnicas de solução diferem na maneira como trabalham com os números nebulosos, os coeficientes do modelo. Os métodos estudados foram implementados e os resultados obtidos são apresentados. Uma comparação é feita considerando funções de ordenação do valor final do jogo obtido por cada algoritmo. Este capítulo também aborda jogos cooperativos e um método para a sua solução (Nishizaki e Sakawa, 1998).

O Capítulo 4 apresenta os jogos nebulosos não cooperativos de soma não zero (Amaral e Gomide, 2007a). É demonstrado como é possível resolver esse tipo de jogo formulando-o como um problema de programação bilinear e utilizando um algoritmo de decomposição clássico para resolução. Uma prova de convergência do algoritmo é também

incluída. Nesse capítulo introduz-se um método de solução que combina o conceito de α -corte e o algoritmo de decomposição para resolver problemas de jogos nebulosos não cooperativos de soma não zero.

O Capítulo 5 desenvolve uma abordagem co-evolutiva para resolver jogos nebulosos não cooperativos (Amaral e Gomide, 2007b). O algoritmo co-evolutivo resolve jogos nebulosos de soma zero e de soma não zero. Exemplos são apresentados e analisados.

Finalmente, o Capítulo 6 conclui o trabalho resumindo as principais contribuições desta dissertação e comentando sobre temas para trabalhos futuros.

Capítulo 1

Teoria de conjuntos nebulosos

1.1 Introdução

A teoria de conjuntos nebulosos foi introduzida como uma abordagem para tratar da imprecisão que ocorre em sistemas e processamento de informação.

Um conjunto nebuloso pode ser definido matematicamente atribuindo-se a cada elemento de um universo de interesse um valor representando o grau de pertinência deste elemento em um conjunto. O grau de pertinência representa o quanto esse elemento é similar ou compatível com um conceito ou classe de objetos representado pelo conjunto nebuloso. Dessa forma, entidades podem pertencer ao conjunto nebuloso em maior ou menor grau, indicado por um maior ou menor grau de pertinência.

O grau de pertinência geralmente é representado por valores reais no intervalo $[0, 1]$. Sendo assim, um elemento que é plenamente compatível com o conceito associado ao conjunto nebuloso possui grau de pertinência igual a 1. De forma análoga, um grau de pertinência 0 indica que o elemento não é compatível com o conceito. Podemos considerar, portanto, um conjunto clássico como um caso particular de conjunto nebuloso no qual apenas a pertinência ou a não pertinência ocorrem. A seguir serão apresentadas as idéias básicas da teoria de conjuntos nebulosos (Zadeh, 1965) e suas características.

1.2 Conjuntos nebulosos

A noção de conjunto ocorre quando há necessidade de organizar, classificar ou generalizar objetos. O conceito de conjunto envolve coleções de objetos que compartilham algo em comum ou não. Um conjunto A em um universo X é descrito por uma função característica $A : X \rightarrow \{0, 1\}$ tal que:

$$A(x) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases}$$

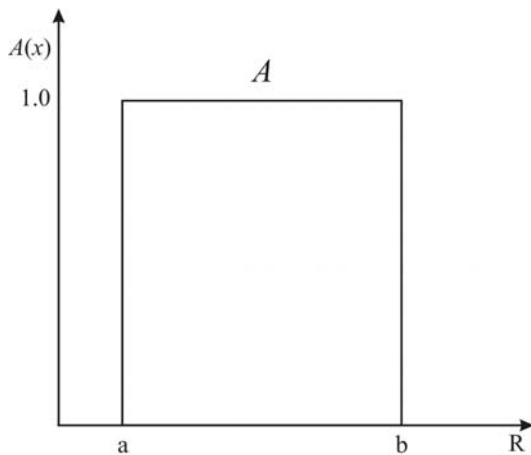
Um exemplo de função característica de um conjunto A é mostrado na Figura 1.1 (a). Neste exemplo, A é o intervalo $[a, b]$ de números reais.

A função $A: X \rightarrow \{0, 1\}$ induz um conjunto com limites bem definidos. O conceito de conjuntos nebulosos (*fuzzy sets*) flexibiliza esses limites e admitir que objetos de um universo X compartilhem parcialmente algo em comum. Isso permite uma representação mais realista de conjuntos, pois freqüentemente um objeto de um universo pode ser compatível com mais de um conjunto, mas com graus distintos.

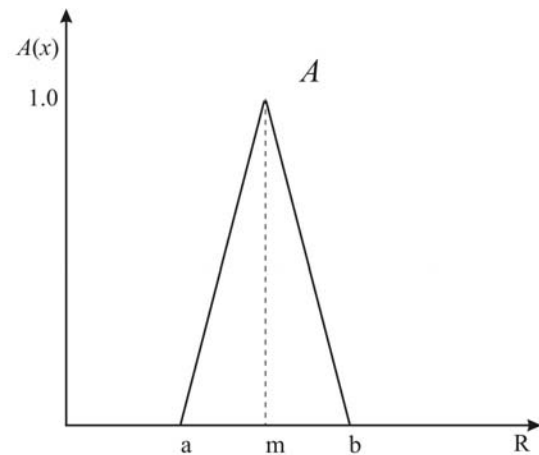
Formalmente, um conjunto nebuloso é caracterizado por uma função de pertinência que mapeia os elementos de um universo X em $[0, 1]$ (Zadeh, 1965), ou seja:

$$A: X \rightarrow [0, 1]$$

Portanto, conjunto nebuloso é sinônimo de função de pertinência.



(a) Exemplo de um conjunto



(b) Exemplo de um conjunto nebuloso

Fig 1.1: Conjunto e conjunto nebuloso.

A Figura 1.1 (b), mostra um exemplo de um conjunto nebuloso cuja função de

pertinência é triangular, representada como (Pedrycz e Gomide, 1998):

$$A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{m-a}, & \text{se } x \in [a, m] \\ \frac{b-x}{b-m}, & \text{se } x \in [m, b] \\ 0, & \text{se } x \geq b \end{cases} \quad (1.1)$$

Duas notações distintas são comumente utilizadas na literatura para representar a função de pertinência de um conjunto nebuloso (Klir e Yuan, 1995). Em uma delas, a função de pertinência de um conjunto nebuloso A é representada por μ_A , ou seja:

$$\mu_A : X \rightarrow [0,1]$$

Em uma outra notação, a função de pertinência é representada por A e tem a forma:

$$A : X \rightarrow [0,1]$$

De acordo com a primeira notação, o símbolo do conjunto nebuloso (A) é diferenciado do símbolo de sua função de pertinência (μ_A). Pela segunda alternativa essa distinção não é feita, porém nenhuma ambigüidade pode resultar dessa notação. Esse trabalho utiliza a segunda notação.

Alguns conceitos importantes da teoria dos conjuntos nebulosos como normalidade, altura, suporte, núcleo e α -corte são apresentados nessa seção.

Definição 1.1 (Pedrycz e Gomide, 1998) *Um conjunto nebuloso é normal se existe um ou mais elementos no universo X cujo valor da função de pertinência é igual a 1, ou seja,*

$$\sup_{x \in X} A(x) = 1$$

onde $\sup$ é o supremo.

O valor $\text{hgt}(A) = \sup_{x \in X} A(x)$ é chamado de altura de A . Portanto, um conjunto nebuloso normal é um conjunto nebuloso cuja altura é a unidade, isto é, $\text{hgt}(A) = 1$.

Definição 1.2 (Pedrycz e Gomide, 1998) *O suporte de um conjunto nebuloso A , denotado por $\text{Supp}(A)$, é o conjunto de todos os elementos de X que pertencem a A com um grau de pertinência maior que zero, ou seja:*

$$\text{Supp}(A) = \{x \mid A(x) > 0\}$$

Definição 1.3 (Pedrycz e Gomide, 1998) *O núcleo de um conjunto nebuloso A , denotado por A^C , é o conjunto de todos os elementos de X que pertencem a A com um grau de pertinência igual a 1, ou seja,*

$$A^C = \{x \mid A(x) = 1\}$$

Definição 1.4 (Pedrycz e Gomide, 1998) *Dado um conjunto nebuloso A em um universo finito X , a cardinalidade de A , denotada por $\text{Card}(A)$, é definida como:*

$$\text{Card}(A) = \sum_{x \in X} A(x)$$

Definição 1.5 (Pedrycz e Gomide, 1998) *Uma t-norma é uma operação binária $t : [0, 1]^2 \rightarrow [0, 1]$ satisfazendo as seguintes condições:*

- *Comutativa:* $x \, t \, y = y \, t \, x$
- *Associativa:* $x \, t \, (y \, t \, z) = (x \, t \, y) \, t \, z$
- *Monotônica:* se $x \leq y$ e $w \leq z$, então $x \, t \, w \leq y \, t \, z$
- *Contorno:* $0 \, t \, x = 0, 1 \, t \, x = x$

Definição 1.6 (Zadeh, 1978; Dubois e Prade, 1980) *A medida de possibilidade de um conjunto nebuloso A em relação ao conjunto nebuloso B , denotado por $Poss(A, B)$ é definida como:*

$$Pos(A, B) = \sup_{x \in X} [\min(A(x), B(x))]$$

Definição 1.7 (Zadeh, 1978; Dubois e Prade, 1980) *A medida de necessidade de um conjunto nebuloso A em relação ao conjunto nebuloso B , denotado por $Nec(A, B)$ é definida como:*

$$Nec(A, B) = \inf_{x \in X} [\max(A(x), 1 - B(x))]$$

Definição 1.8 (Yager 1982) *Seja um universo finito X . A especificidade do conjunto nebuloso A definido em X , denotada por $Sp(A)$, atribui a A um número não negativo que*

- $Sp(A) = 1$, se e somente se existir apenas um elemento de X para o qual A assume um grau de pertinência igual a 1, enquanto para os demais elementos o grau de pertinência é igual a 0.
- $Sp(A) = 0$ se $A(x) = 0$ para todos os elementos de X .
- $Sp(A_1) \leq Sp(A_2)$ se $A_1 \supset A_2$

Yager (1982) sugere uma medida de especificidade como a integral:

$$Sp(A) = \int_0^{\alpha_{\max}} \frac{1}{Card(A_\alpha)} d\alpha$$

onde $\alpha_{\max} = hgt(A)$, ou seja, o maior grau de pertinência obtido entre todos elementos do conjunto. Para um universo finito, o que implica em um número finito de graus de pertinência, a integral é substituída pela soma:

$$\text{Sp}(A) = \sum_{i=1}^n \frac{1}{\text{Card}(A_{\alpha_i})} \Delta \alpha_i$$

onde $\Delta \alpha_i = \alpha_i - \alpha_{i-1}$ e $\alpha_0 = 0$, n é o número de elementos do universo e $\text{Card}(A)$ é a cardinalidade de A .

1.3 Teorema da representação

Seja A um conjunto nebuloso em X e considere $\alpha \in [0, 1]$. O conjunto $A_\alpha = \{x \mid A(x) \geq \alpha\}$ é chamado de α -corte de A . O conjunto α -corte forte é definido como (Klir e Yuan, 1995) $A_\alpha^+ = \{x \mid A(x) \geq \alpha\}$.

O α -corte pode ser representado também por $[A_\alpha^L, A_\alpha^R]$, onde $A_\alpha^L = \inf(A_\alpha)$ e $A_\alpha^R = \sup(A_\alpha)$, onde $\sup(A)$ e $\inf(A)$ são, respectivamente, o supremo e o ínfimo de A_α .

A Figura 1.2 ilustra o α -corte de um conjunto nebuloso A , com função de pertinência triangular.

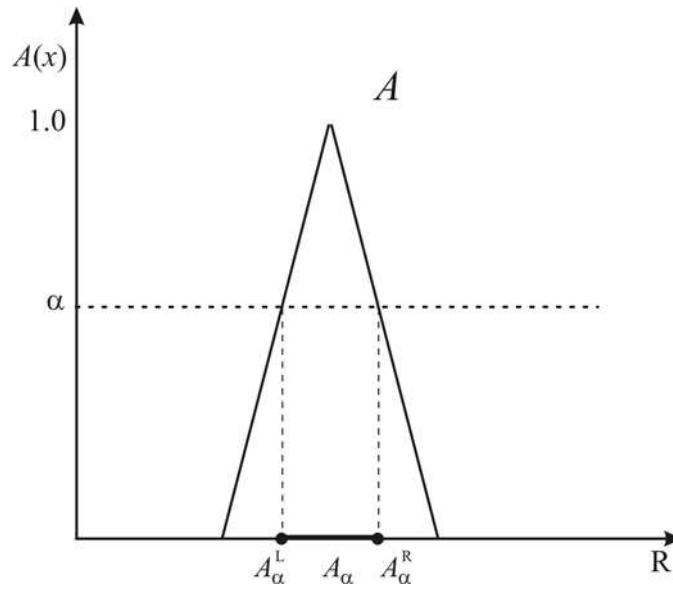


Fig 1.2: α -corte de A .

Em particular, o nível mais alto, ou seja α igual a 1, determina o conjunto de X que é um subconjunto próprio do $\text{Supp}(A)$. Quanto menor o valor de α , maior a cardinalidade do α -corte. Em geral, se $\alpha_1 > \alpha_2$ então $A_{\alpha_1} \subset A_{\alpha_2}$.

Um papel importante dos α -cortes é a capacidade de representar conjuntos nebulosos. Nessa seção será demonstrado que qualquer conjunto nebuloso pode ser representado por uma família de α -cortes. Esta é a idéia do teorema da representação.

O teorema da representação estabelece que qualquer conjunto nebuloso A pode ser decomposto em α -cortes, isto é,

$$A = \bigcup_{\alpha \in [0,1]} (\alpha A_\alpha) \quad (1.2)$$

Prova: (Klir e Yuan, 1995) Para cada $x \in X$, seja $a = A(x)$ e ${}^\alpha A(x) = \alpha A_\alpha(x)$. Então,

$$\begin{aligned} \left(\bigcup_{\alpha \in [0,1]} {}^\alpha A \right)(x) &= \sup_{\alpha \in [0,1]} {}^\alpha A(x) \\ &= \max \left[\sup_{\alpha \in [0,a]} {}^\alpha A(x), \sup_{\alpha \in [a,1]} {}^\alpha A(x) \right]. \end{aligned}$$

Para cada $\alpha \in (a, 1]$, temos $A(x) = a < \alpha$ e, sendo assim, ${}^\alpha A = 0$. Por outro lado, para cada $\alpha \in [0, a]$, temos $A(x) = a \geq \alpha$ e ${}^\alpha A = \alpha$. Então,

$$\left(\bigcup_{\alpha \in [0,1]} {}^\alpha A \right)(x) = \sup_{\alpha \in [0,a]} \alpha = a = A(x).$$

Uma vez que os mesmos argumentos são válidos para cada $x \in X$, a validade de (1.2) é estabelecida. $\square$

O teorema da representação descrito acima é ilustrado na Figura 1.3. Resultado análogo pode ser amostrado para α -corte fortes.

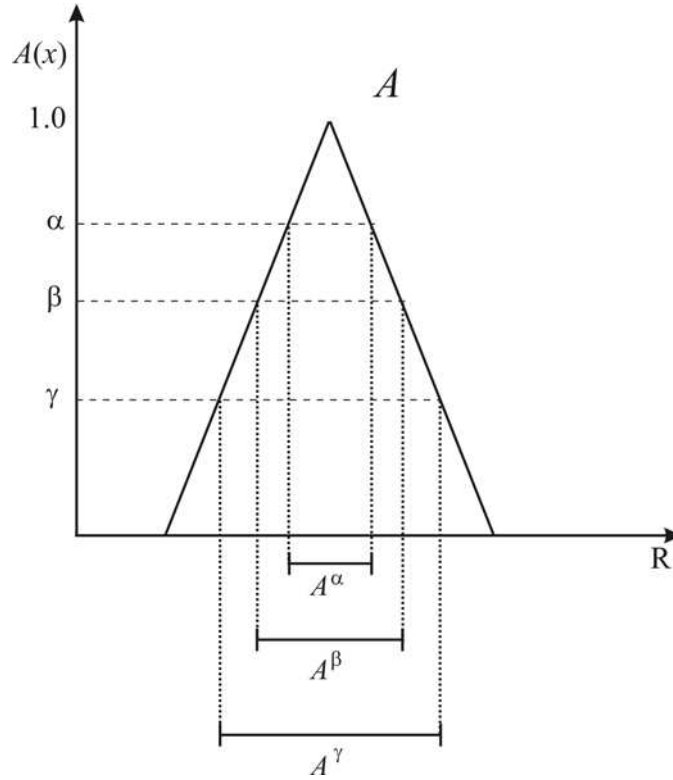


Fig 1.3: Teorema da representação.

1.4 Princípio da extensão

O princípio da extensão é utilizado para transformar conjuntos nebulosos. Sejam X e Y dois conjuntos e f uma função da forma (Klir e Yuan, 1995):

$$f : X \rightarrow Y$$

Seja A um conjunto nebuloso em X . O princípio da extensão estabelece que a imagem de A nesse mapeamento é um conjunto nebuloso $B = f(A)$ em Y , tal que, para cada $y \in Y$,

$$B(y) = \sup_x A(x), \text{ sujeito a } x \in X \text{ e } y = f(x) \quad (1.3)$$

O princípio da extensão é uma generalização do caso clássico onde, expressando os conjuntos $f(A)$ e $f^{-1}(B)$ por suas funções características, obtemos:

$$[f(A)](y) = \sup_x A(x) \text{ sujeito a } x \in X \text{ e } y = f(x)$$

$$[f^{-1}(B)](x) = B(f(x))$$

$$f^{-1} : Y \rightarrow X$$

onde f^{-1} é a inversa de f definida por:

$$f^{-1}(B) = \{x \mid f(x) \in B\}$$

Para exemplificar o princípio considere o caso clássico. Seja $X = \{a, b, c\}$ e $Y = \{1, 2\}$, Figura 1.4 e Tabela 1.1. Considere a função f : $f(a) = 1$, $f(b) = 1$ e $f(c) = 2$. Assim, temos:

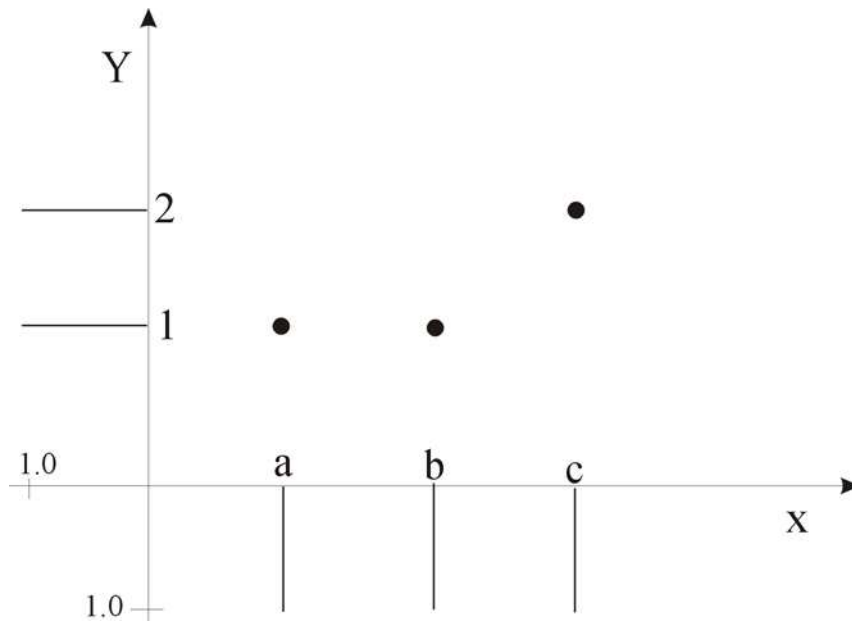


Fig 1.4: Princípio de extensão clássico.

A	$f(A)$	B	$f'(B)$
$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$
$\{a\}$	$\{1\}$	$\{1\}$	$\{a, b\}$
$\{b\}$	$\{1\}$	$\{2\}$	$\{c\}$
$\{c\}$	$\{2\}$	$\{1,2\}$	$\{a,b,c\}$
$\{a,b\}$	$\{1\}$		
$\{a,c\}$	$\{1,2\}$		
$\{b,c\}$	$\{1,2\}$		
$\{a,b,c\}$	$\{1,2\}$		

Tab 1.1: Exemplo do princípio de extensão clássico.

No caso de transformação de um conjunto nebuloso A em X , o princípio da extensão estabelece que a imagem de A é um conjunto nebuloso $B = f(A)$ em Y , cuja função de pertinência é dada por (1.3), conforme ilustra a Figura 1.5.

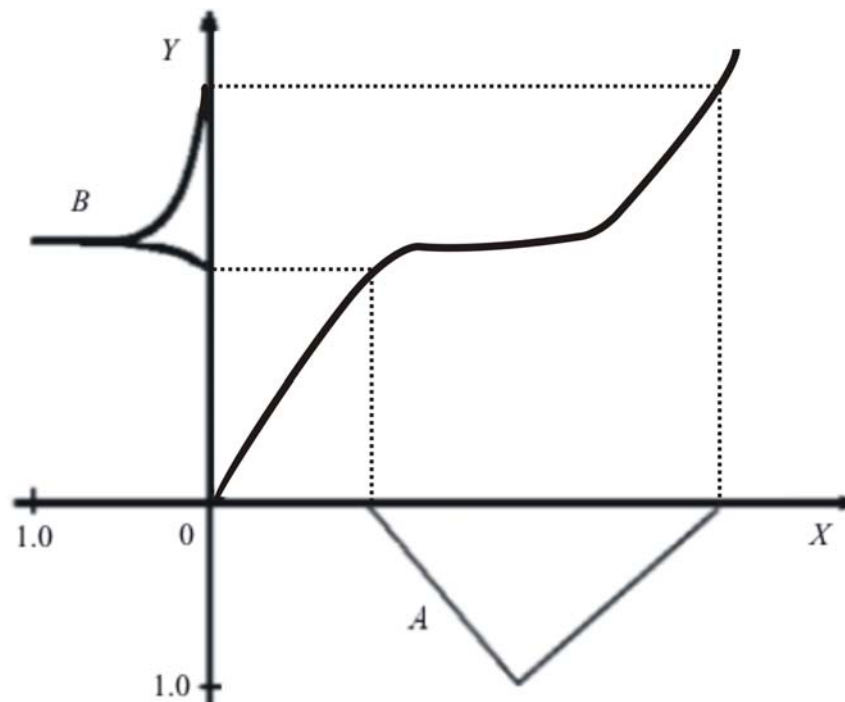


Fig 1.5: Ilustração do princípio da extensão.

1.5 Números nebulosos

Entre os vários tipos de conjuntos nebulosos, um de grande relevância são os números nebulosos. Números nebulosos representam quantidades imprecisas e são particularmente importantes para representar valores imprecisos de parâmetros de modelos.

Definição 1.9 (Nguyen e Walker, 1999) *Um número nebuloso é um conjunto nebuloso A definido no universo dos números reais $\mathbb{R}$ cuja função de pertinência $A : \mathbb{R} \rightarrow [0, 1]$ satisfaz as seguintes condições:*

- (i) *Deve existir um único número real cujo valor da função de pertinência é igual a 1, isto é, A é um conjunto normal, $A(x)=1$;*
- (ii) *A_α é um intervalo fechado para todo $\alpha \in [0,1]$;*
- (iii) *O suporte de A é limitado.*

O exemplo mais simples de número nebuloso é o número nebuloso triangular. A função de pertinência do número nebuloso triangular é aquela mostrada na Figura 1.6(a).

Um número nebuloso triangular pode ser representado por $A = (A^L, A^C, A^R)$, onde A^C é o núcleo, e A^L e A^R são o limite inferior e superior, respectivamente.

O valor $d = A^R - A^C$ e $e = A^C - A^L$ são chamados de dispersão à direita e à esquerda, respectivamente. Para um número nebuloso triangular simétrico, $d = e$. A Figura 1.6 (b) mostra uma função de pertinência para um número nebuloso triangular simétrico.

Alternativamente, quando for necessário enfatizar, um número nebuloso pode ser representado utilizando o símbolo $\sim$, ou seja, $\tilde{a}$.

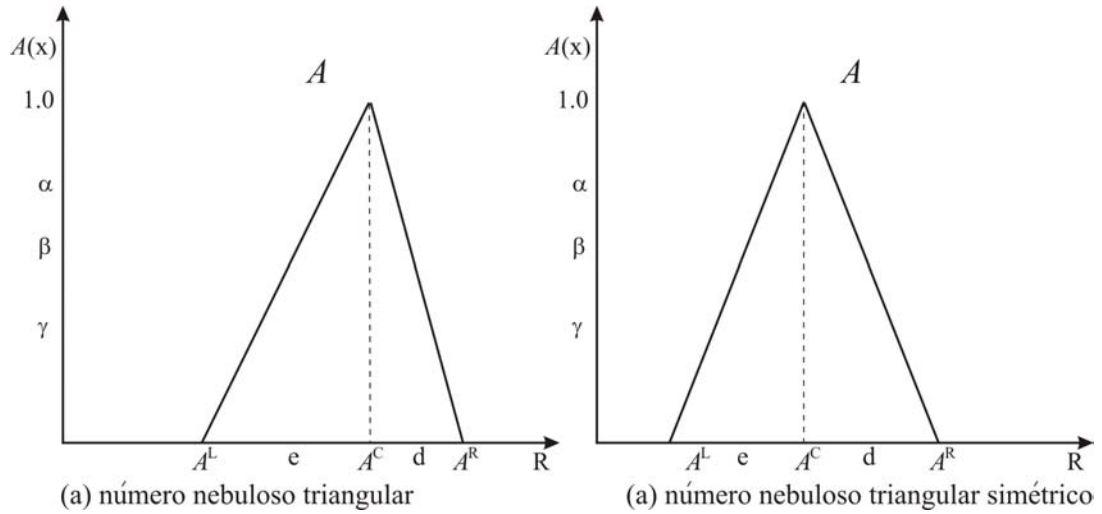


Fig 1.6: Números nebulosos triangulares

Definição 1.10 (Maeda, 2003) *Seja m um número real e h um número positivo. Um número nebuloso A cuja função de pertinência é dada por:*

$$A(x) = \begin{cases} 1 - \left| \frac{x - m}{h} \right|, & \text{se } m - h \leq x \leq m + h \\ 0, & \text{caso contrário} \end{cases} \quad (1.4)$$

é chamado de número nebuloso triangular simétrico. Ou seja, a dispersão a direita e a esquerda são iguais, $d = e = h$.

Podemos representar um número nebuloso triangular simétrico através de seu núcleo m e de sua dispersão h . Se h é a dispersão de um número nebuloso triangular simétrico, então ele pode ser representado por $\tilde{a}$ conforme (1.5):

$$\tilde{a} = (m, h) \quad (1.5)$$

A Figura 1.7 ilustra um exemplo de número nebuloso triangular simétrico.

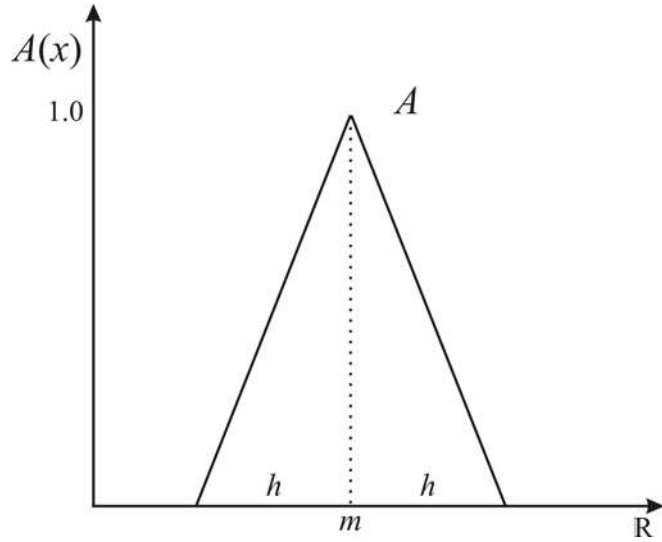


Fig 1.7 Número nebuloso triangular simétrico

Segundo a definição clássica na literatura de conjuntos nebulosos (Nguyen e Walker, 1999), os números nebulosos são conjuntos nebulosos convexos.

Um conjunto nebuloso A é convexo se sua função de pertinência satisfaz (Pedrycz e Gomide, 1998):

$$A(\lambda x_1 + (1 - \lambda)x_2) \geq \min[A(x_1), A(x_2)] \quad (1.5)$$

para qualquer $x_1, x_2 \in X$ e $\lambda \in [0, 1]$. Um conjunto nebuloso A é convexo se A_α é convexo para qualquer $\alpha \in [0, 1]$.

Prova: Assuma que A é convexo, e seja $\alpha = A(x_1) \leq A(x_2)$. Então $x_1, x_2 \in A_\alpha$ e, além disso, $\lambda x_1 + (1 - \lambda)x_2 \in A_\alpha$ para qualquer $\lambda \in [0, 1]$, pela convexidade de A . Consequentemente:

$$A(\lambda x_1 + (1 - \lambda)x_2) \geq \alpha = A(x_1) = \min[A(x_1), A(x_2)].$$

Assuma que A satisfaz (1.5). É preciso provar que para qualquer $\alpha \in [0, 1]$, A_α é convexo. Agora, para qualquer $x_1, x_2 \in A_\alpha$, com $A(x_1) \geq \alpha$, $A(x_2) \geq \alpha$, e para qualquer $\lambda \in [0, 1]$, por (1.5):

$$A(\lambda x_1 + (1-\lambda)x_2) \geq \min[A(x_1), A(x_2)] \geq \min[\alpha, \alpha] = \alpha$$

Então $\lambda x_1 + (1-\lambda)x_2 \in A_\alpha$. Sendo assim, A_α é convexo para qualquer $\alpha \in [0,1]$. Logo, A é convexo. $\square$

Portanto pode-se afirmar ainda que A é convexo se A_α é convexo para todo $\alpha \in [0,1]$. Observar que (1.5) indica que, se A é convexo, então sua função de pertinência $A(x)$ é quasi-concava (Bazaraa et al., 1979).

1.6 Ordenação de números nebulosos

Em tomada de decisão, números nebulosos aparecem em vários casos, como por exemplo na matriz de pagamentos em problemas de jogos, nos coeficientes da função objetivo e restrições em problemas de otimização.

Pelo fato de números nebulosos representarem valores imprecisos, geralmente traduzindo termos lingüísticos, como em torno de cinco ou menor que dez, o problema de ordenar números nebulosos é delicado. Em muitos casos é difícil, mesmo intuitivamente, dizer se um número nebuloso é maior ou menor que outro.

O problema de ordenar dois números nebulosos já foi estudado e diversas alternativas foram propostas (Zhu e Lee, 1992). Essa seção apresenta três métodos de ordenação. Uma variedade de outros propostos na literatura são variações desses três métodos básicos.

1.6.1 Método de Yager

O método de ordenação proposto por Yager (1981) consiste em mapear cada número nebuloso A na reta real $\mathbb{R}$ utilizando a função:

$$F(A) = \frac{\int_{A^L}^{A^R} g(x)A(x)dx}{\int_{A^L}^{A^R} A(x)dx}$$

onde A^R e A^L são os limites superior e inferior do suporte de A e $g(x)$ é uma função de ponderação (peso) que representa a importância do valor x .

O Capítulo 3 descreve o método de Lee-Kwang e Lee (1999) para jogos nebulosos com dois jogadores, método este baseado na ordenação de Yager, onde $g(x)$ é um número triangular nebuloso que representa o ponto de vista dos jogadores.

1.6.2 Método de α -cortes

O segundo método de ordenação de números nebulosos utiliza o conceito de α -cortes. Nesse método, para comparar dois números nebulosos A e B seleciona-se um valor particular para $\alpha \in [0, 1]$ e determina-se os respectivos α -cortes $A_\alpha = [A_\alpha^L, A_\alpha^R]$ e $B_\alpha = [B_\alpha^L, B_\alpha^R]$, e define-se:

$$A \leq B \text{ se } A_\alpha^R \leq B_\alpha^R$$

onde A_α^L é o limite inferior do α -corte e A_α^R , o limite superior.

O método de ordenação por α -cortes é interessante pelo ponto de vista didático. No entanto, percebe-se que esse método pode ordenar números nebulosos de forma distinta, dependendo de como o valor de α é escolhido, não sendo portanto o método mais eficaz. Para se obter um resultado mais satisfatório, a literatura recomenda (Klir e Yuan, 1995) um valor para α maior que 0.5. Contudo, para resultados mais consistentes, faz-se necessário o uso de outros métodos de ordenação para comprovar os resultados.

1.6.3 Método baseado no princípio da extensão

O terceiro método baseia-se no princípio da extensão. Esse método pode ser usado para ordenar vários números nebulosos $A_1, A_2, \dots, A_n$ de uma só vez. A idéia básica é construir um conjunto nebuloso P em $\{A_1, A_2, \dots, A_n\}$ chamado conjunto de prioridades, em que $P(A_i)$ é o grau em que A_i é ordenado como o maior número nebuloso. Usando o princípio da extensão, P é definido para cada $i \in N_n$ pela fórmula:

$$P(A_i) = \sup \min_{k \in N_n} A_k(r_k)$$

onde o supremo é escolhido entre todos os vetores $(r_1, r_2, \dots, r_n) \in \mathbb{R}^n$ sendo que $r_i > r_j$, para todo $j \in N_n$, e N_n é o conjunto de inteiros positivos, $N_n = \{1, 2, \dots, n\}$.

1.6.4 Exemplos

Essa seção ilustra os três métodos utilizando números nebulosos triangulares e compara os resultados fornecidos por cada um deles.

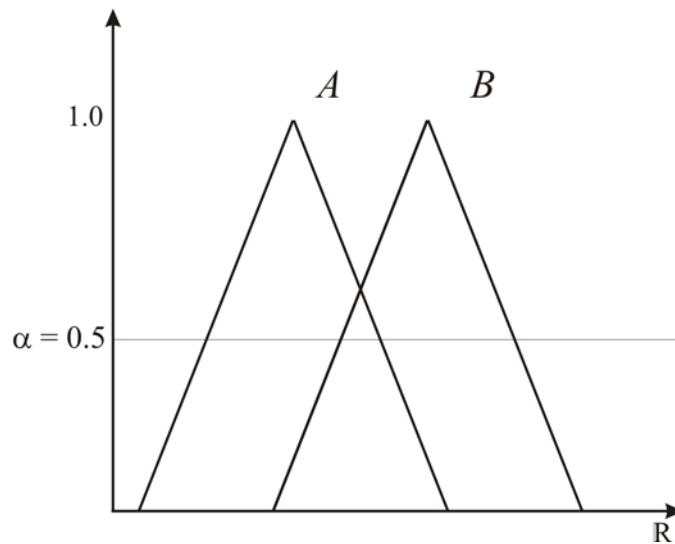


Fig 1.8: Primeiro exemplo de ordenação.

No exemplo da Figura 1.8, intuitivamente o número nebuloso B é maior que A . Os três métodos concordam com a intuição. Em particular o método do α -corte indica que, para qualquer valor de α , o número nebuloso B é maior que A , ou seja, $A_\alpha^R \leq B_\alpha^R$.

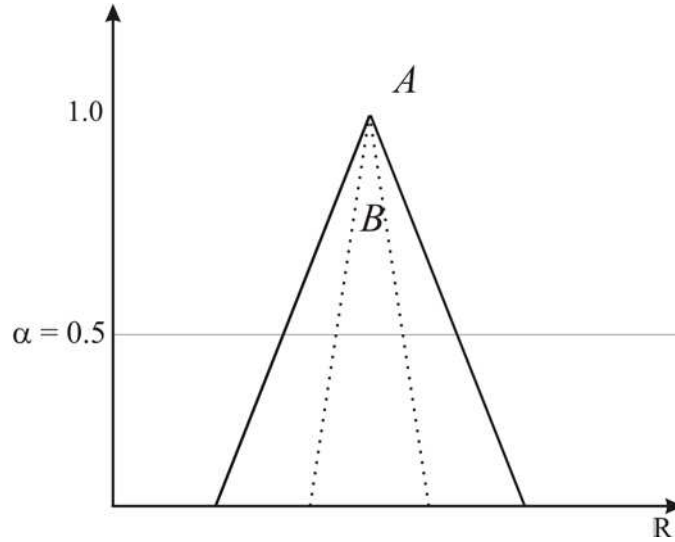


Fig 1.9: Segundo exemplo de ordenação.

O segundo exemplo já não é tão intuitivo (Figura 1.9). O método de Yager fornece resultados diferentes de acordo com a função $g(x)$. Por exemplo, sob o ponto de vista de Lee-Kwang e Lee (1999), se a preferência for para números menores, ou seja, se o ponto de vista for pessimista, o número B é maior que o número A , porque A contém valores menores que o menor valor de B . Já se o ponto de vista for otimista, ou constante, A é maior que B . Isso porque A contém valores maiores que o maior valor de B . Já para o método de α -cortes A é maior que B , independentemente do valor escolhido para α . No caso do terceiro método, baseado no princípio da extensão, A é igual a B .

Para o terceiro exemplo, Figura 1.10, para qualquer valor de α , $B_\alpha^R \leq A_\alpha^R$, ou seja, A é maior que B , pelo método de α -cortes. Também pelo método de Yager A é maior que B , mas o método baseado no princípio da extensão diz que A é igual a B :

$$P(A) = \sup_{x \geq y} \min[A(x), B(y)] = 1 \quad \text{e} \quad P(B) = \sup_{y \geq x} \min[A, B] = 1$$

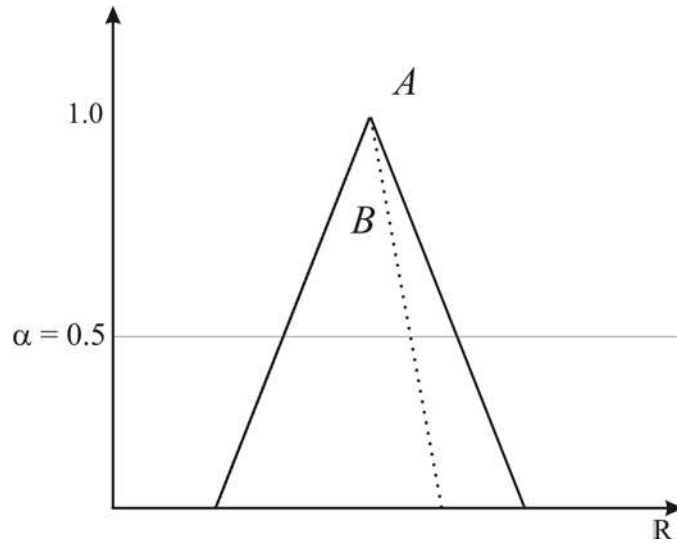


Fig 1.10: Terceiro exemplo de ordenação.

No quarto exemplo, Figura 1.11, o método de Yager indica que B é maior do que A , sob um ponto de vista pessimista, mas sob um ponto de vista otimista, ou constante, A é igual a B .

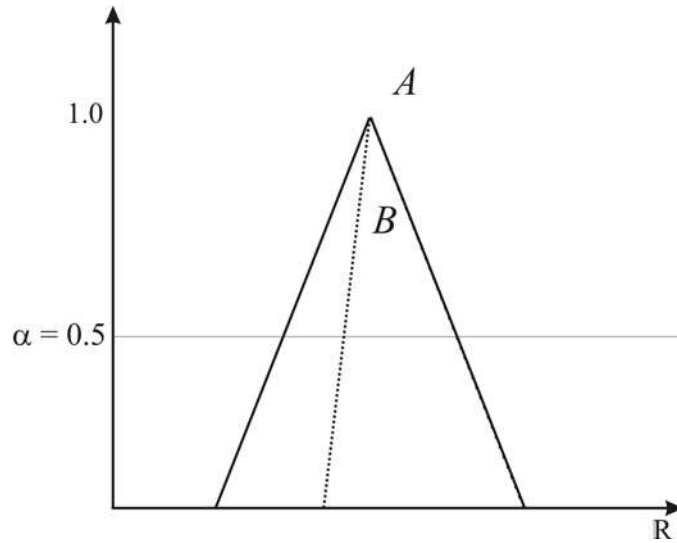


Fig 1.11: Quarto exemplo de ordenação.

Já pelo método de α -cortes A é igual a B . Isso porque $B_\alpha^R = A_\alpha^R$ para qualquer valor de α . Utilizando o método baseado no princípio, A é igual a B :

$$P(A) = \sup_{x \geq y} \min[A, B] = 1 \quad \text{e} \quad P(B) = \sup_{y \geq x} \min[A, B] = 1$$

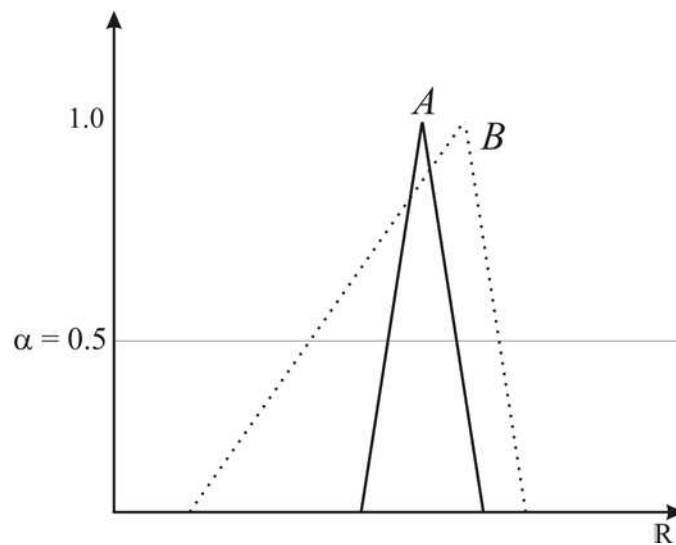


Fig 1.12: Quinto exemplo de ordenação.

Para o exemplo da Figura 1.12, os três métodos sugerem que B é maior do que A .

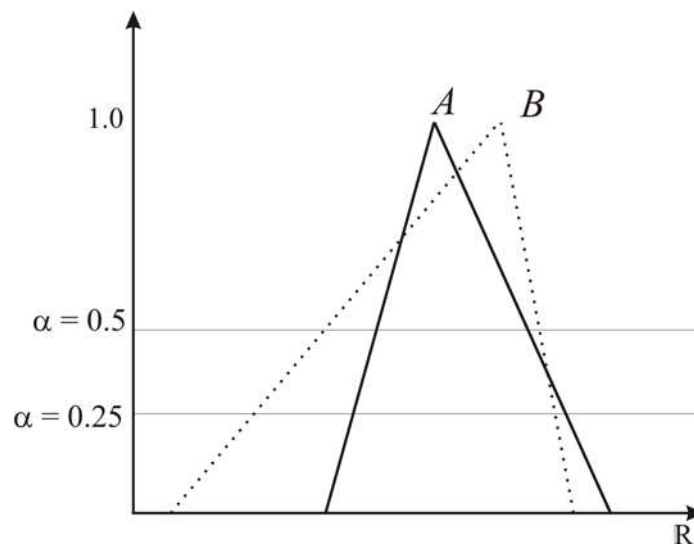


Fig 1.13: Sexto exemplo de ordenação.

No último exemplo, Figura 1.13, pelo método de Yager, B é maior que A . Esse exemplo mostra como a escolha do valor de α pode influenciar no método de α -cortes. Para α igual a 0.25 A é maior que B . Já para um valor de α igual a 0.5, B é maior que A . Pelo método baseado no princípio da extensão B é maior que A :

$$P(A) = \sup_{x \geq y} \min[A, B] = 0.9 \quad \text{e} \quad P(B) = \sup_{y \geq x} \min[A, B] = 1$$

Os exemplos acima mostram que os três métodos apresentados podem ser utilizados para ordenar números nebulosos. A escolha de qual método utilizar depende do problema a ser resolvido. Pelos resultados apresentados nos exemplos vimos que o método de α -cortes é bastante simples e também pode ser utilizado para ordenar números nebulosos mas nem sempre é adotado pois pode apresentar resultados diferentes dependendo do valor de α .

1.7 Resumo

Esse capítulo apresentou os conceitos básicos da teoria de conjuntos nebulosos, que são particularmente relevantes para este trabalho. A teoria de conjuntos nebulosos permite tratar imprecisões em problemas de decisão e aproxima modelos do mundo real.

Naturalmente, a teoria de conjuntos nebulosos é um assunto muito mais abrangente, envolvendo conceitos de operações com conjuntos nebulosos, lógica nebulosa e sistemas baseados em regras nebulosas por exemplo. Esse capítulo enfatizou apenas os conceitos básicos da teoria de conjuntos nebulosos que serão utilizados para tratar a imprecisão de valores em matrizes de jogos e parâmetros de problemas de otimização utilizando números nebulosos.

Capítulo 2

Teoria de jogos

2.1 Introdução

Um jogo é uma situação entre n entidades (grupos, agentes, etc), chamados de jogadores, caracterizada por um conjunto de regras com forma conhecida de pagamento (ganhos ou perdas). As regras definem decisões elementares, ou lances, do jogo. Jogadores diferentes podem tomar decisões diferentes, mas tipicamente assume-se as decisões de cada jogador é informação pública.

A teoria de jogos pode ser definida como um tratamento matemático de situações que envolvem conflito de interesses, buscando indicar as melhores opções que, sob determinadas condições, conduzirão ao objetivo desejado.

A noção de conflitos de interesses é antiga, porém seu estudo científico teve início recentemente. As publicações de J. von Neumann e O. Morgenstern em 1944 atraiu a atenção de pesquisadores para a teoria de jogos. Situações de conflito são comuns em aplicações de economia, engenharia, aeronáutica, sociologia, política, biologia e computação.

Na teoria de jogos, as decisões são interpretadas através de modelos de jogos de estratégia, onde há várias opções, cada estratégia possuindo um grau de risco e um pagamento que pode ser obtido, dependendo da estratégia e decisões escolhidas pelos outros jogadores.

A seguir serão apresentados os elementos da teoria de jogos.

2.2 Elementos de um jogo

Um jogo, em sua forma geral, é definido por um conjunto de jogadores, de ações (decisões elementares) e de funções de pagamento (Basar e Olsder, 1982). O jogador é

quem determina as decisões, tendo como objetivo otimizar seus pagamentos. A estratégia é o conjunto de ações que estão disponíveis para os jogadores, auxiliando-o na tomada de decisão e na escolha da ação a ser tomada diante das ações dos demais jogadores. O pagamento dos jogadores decorre do resultado final do jogo. O pagamento vai depender das estratégias escolhidas pelos jogadores.

A forma mais comum de representar jogos finitos é através de uma matriz de pagamentos. Por exemplo, em um jogo com dois jogadores, uma matriz de pagamento $A=[a_{ij}]$ de dimensões $m \times n$, onde o número de linhas m corresponde às m ações associadas à estratégia do primeiro jogador e o número de colunas n corresponde às n ações associadas a estratégia do segundo jogador. Assim, na matriz de pagamentos cada jogador tem um conjunto finito de estratégias simples, as decisões elementares. Cada jogador conhece as ações elementares do outro, mas não pode assegurar que ação cada um deles tomará. Assim uma caracterização completa de jogo é dada pela matriz de pagamento onde a_{ij} é o valor (pagamento) que o primeiro jogador obtém quando este utiliza sua i -ésima ação elementar (estratégia pura) e o segundo sua j -ésima ação elementar. Em jogos de soma zero, a matriz de pagamentos de um jogador é a negativa da matriz do outro. Em jogos de soma não zero as matrizes de pagamento de cada jogador possuem valores diferentes para cada ação elementar (adotamos os termos ação, decisão elementar ou decisão, simplesmente, como sinônimos).

Os jogos podem ser classificados em cooperativos ou não cooperativos. Jogos não cooperativos são aqueles em que não há comunicação entre os jogadores, ou os jogadores não fazem acordos, mesmo que haja comunicação entre eles. Já nos jogos cooperativos é permitido aos jogadores compartilharem decisões para garantir que o pagamento beneficie todos os jogadores. O jogo é chamado de dinâmico se for considerada a seqüência em que cada decisão é tomada.

Conforme antecipado acima, existem dois tipos de jogos, em relação ao resultado de pagamentos: os jogos de soma zero e os de soma não zero. Nos jogos de soma zero, a soma dos ganhos dos jogadores resulta em zero. Isso significa necessariamente que, para um jogo com dois jogadores, o que um jogador ganha o outro perde. Já nos jogos de soma não zero o que um jogador perde não é necessariamente quanto o outro jogador ganha. Esse tipo de

jogo também é conhecido como bimatricial, pois possui matrizes de pagamento distintas, uma para cada jogador.

É importante lembrar que na teoria de jogos pode haver mais do que dois jogadores. Quanto maior o número de jogadores mais complexo é o problema.

Um elemento importante na teoria de jogos é a noção de uma solução de equilíbrio. A solução de equilíbrio pode ser entendida como aquela que otimiza os ganhos (pagamento esperado) de cada jogador. Soluções de equilíbrio significam que um jogador não melhora seu desempenho alterando sua decisão unilateralmente, isto é, adotando decisões que não sejam aquelas de equilíbrio. Esta noção é chamada de *solução de equilíbrio de Nash*, ou apenas *solução de Nash*.

O par de estratégias que proporcionam soluções de equilíbrio para um jogo e que não ocasionam uma melhora no resultado de um dos jogadores, sem piorar o resultado do outro, é uma *solução de Pareto* (Basar e Olsder, 1982). Se existir uma solução de equilíbrio de Nash ela é considerada uma solução Pareto ótima no contexto de jogos.

2.3 Jogos de soma zero com dois jogadores

Os jogos de soma zero com dois jogadores são aqueles em que o que um jogador ganha, o outro jogador perde. Por convenção, chamaremos os dois jogadores de Jogador I e II, respectivamente.

Definição 2.1 (Nishizaki e Sakawa, 2001) Sejam $I = \{1, \dots, m\}$ e $J = \{1, \dots, n\}$ conjuntos de decisões e f_1 e f_2 as funções de pagamentos do jogador I e II, respectivamente. Quando o jogador I escolhe uma decisão $i \in I$ e o jogador II escolhe uma decisão $j \in J$, $f_1(i, j)$ e $f_2(i, j)$ são os pagamentos dos jogadores I e II, respectivamente. Dessa forma, um jogo é de soma zero se e somente se a função de pagamento satisfaz:

$$f_1(i, j) + f_2(i, j) = 0, \forall i \in I, \forall j \in J \quad (2.1)$$

Definindo $a_{ij} = f_1(i, j) = -f_2(i, j)$, o jogo de dois jogadores de soma zero pode ser representado pela matriz:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad (2.2)$$

onde m e n são o número de decisões (decisões elementares ou estratégias puras) do jogador I e II, respectivamente. A matriz A é chamada de matriz de pagamentos (*payoff matrix*).

Definição 2.2 (Nishizaki e Sakawa, 2001) *Uma estratégia mista $x=(x_1, \dots, x_m)$ para o jogador I é uma distribuição de probabilidade sobre suas estratégias puras. O conjunto de estratégias mistas para o jogador I é representado por:*

$$X = \{x = (x_1, \dots, x_m) \in \mathbb{R}^m \mid \sum_{i=1}^m x_i = 1, x_i \geq 0, i = 1, \dots, m \} \quad (2.3)$$

onde $\mathbb{R}^m$ é um conjunto de números reais de dimensão m . De forma similar, o conjunto de estratégias mistas para o jogador II é representado por:

$$Y = \{y = (y_1, \dots, y_n) \in \mathbb{R}^n \mid \sum_{j=1}^n y_j = 1, y_j \geq 0, j = 1, \dots, n \} \quad (2.4)$$

Definição 2.3 (Nishizaki e Sakawa, 2001) *O pagamento esperado do jogo (expected payoff), após a escolha das estratégias pelos jogadores, em um jogo de soma zero, é dado por:*

$$P(x, y) = \sum_{i=1}^m \sum_{j=1}^n x_i a_{ij} y_j = xAy \quad (2.5)$$

Em um jogo de soma nula, o pagamento mínimo esperado pelo jogador I é $v_1(x) = \min_{y \in Y} xAy$. Portanto, o objetivo do jogador I é escolher uma estratégia que maximize seu pagamento mínimo esperado:

$$v_I = \max_{x \in X} \min_{y \in Y} xAy \quad (2.6)$$

O par de estratégias (x, y) que satisfazem (2.6) é chamado de solução ótima. O valor de v_I é o valor do jogo caracterizado pela matriz A para o jogador I. Da mesma forma, para o jogador II, o valor do jogo será:

$$v_{II} = \min_{y \in Y} \max_{x \in X} xAy \quad (2.7)$$

Observe que estamos evitando o uso de x' , o transposto do vetor x em (2.5), (2.6) e (2.7), assim com em outras ocasiões. Isto será adotado para simplificar a notação, sempre que não causar confusão.

Por exemplo, seja a matriz de pagamentos da Tabela 2.1:

		II			
		1	2	3	4
I	1	3	0	-4	-4
	2	2	3	1	2
	3	-4	2	-1	3

Tab 2.1: Matriz de pagamentos

O jogador I segundo (2.6) precisa escolher a linha i da matriz de pagamentos cujo menor valor seja maior que o menor valor das outras linhas. Já o jogador II, agindo racionalmente, deve escolher a coluna j em que o maior valor dessa coluna seja menor que o maior valor das demais colunas, segundo (2.7). Percebe-se que, para o exemplo em questão, $i = 2$ e $j = 3$. Essa estratégia é a chamada de estratégia de equilíbrio, onde nenhum jogador pode melhorar o valor do jogo para si próprio modificando sua estratégia unilateralmente.

Teorema 2.1 (Nishizaki e Sakawa, 2001) *Para um jogo A de soma zero e com dois jogadores, temos:*

$$\max_{x \in X} \min_{y \in Y} xAy = \min_{y \in Y} \max_{x \in X} xAy \quad (2.8)$$

e o par de estratégias (x, y) que satisfaz essa condição é a solução de equilíbrio.

O Teorema 2.1 é o teorema fundamental da teoria dos jogos e é devido a von Neumann (Neumann e Morgenstern, 1944). Uma prova desse teorema mostra a existência de soluções de equilíbrio e sugere um método computacional para encontrar a solução através de um problema de programação linear (Nishizaki e Sakawa, 2001).

Assumindo $a_{ij} > 0, i \in I, j \in J$ e sendo $v(x) = \min_{y \in Y} xAy = \min_{j \in J} xA.j$,

$$v_I = \max_{x \in X} \min_{y \in Y} xAy = \max_{x \in X} \min_{j \in J} xA.j = \max_{x \in X} \min_{j \in J} \sum_{i=1}^m a_{ij}x_i \quad (2.9)$$

onde $xA.j = a^j = [a_{1j}, a_{2j}, \dots, a_{mj}]'$ é a j -ésima coluna da matriz A .

Dessa forma, para encontrar v_I podemos formular o seguinte problema de programação matemática:

$$\begin{aligned} \max \quad & \min_{j \in J} \sum_{i=1}^m a_{ij}x_i \\ \text{sa : } & x_1 + \dots + x_m = 1 \\ & x_i \geq 0, i = 1, \dots, m. \end{aligned} \quad (2.10)$$

Introduzindo uma variável auxiliar z , o problema pode ser transformado no problema de programação linear a seguir:

$$\begin{aligned}
 \max \quad & z \\
 \text{sa : } & a_{1j}x_1 + \dots + a_{mj}x_m \geq z, \quad j = 1, \dots, n \\
 & x_1 + \dots + x_m = 1 \\
 & x_i \geq 0, \quad i = 1, \dots, m.
 \end{aligned} \tag{2.11}$$

Seja $s_i = x_i/z$, $i = 1, \dots, m$. Maximizar z é o equivalente a minimizar $1/z = s_1 + \dots + s_m$ e $a_{ij} > 0$, $i \in I, j \in J$, o programa acima pode ser reescrito como:

$$\begin{aligned}
 \min \quad & s_1 + \dots + s_m \\
 \text{sa : } & a_{1j}s_1 + \dots + a_{mj}s_m \geq 1, \quad j = 1, \dots, n \\
 & s_i \geq 0, \quad i = 1, \dots, m
 \end{aligned} \tag{2.12}$$

Para o jogador II temos, analogamente,

$$\begin{aligned}
 \max \quad & r_1 + \dots + r_n \\
 \text{sa : } & a_{1j}r_1 + \dots + a_{in}r_n \leq 1, \quad i = 1, \dots, m \\
 & r_i \geq 0, \quad i = 1, \dots, n
 \end{aligned} \tag{2.13}$$

Os problemas (2.12) e (2.13) são dois problemas lineares duais entre si.

2.4 Jogos de soma não zero com dois jogadores

Os jogos vistos até agora são de soma zero, pois satisfazem a condição (2.1). Os jogos que não satisfazem essa condição são jogos de soma não zero. Sejam $a_{ij} = f_1(i, j)$ e $b_{ij} = f_2(i, j)$ o pagamento de um jogo de soma não zero pode ser representado por duas matrizes $A = [a_{ij}]$ e $B = [b_{ij}]$ $m \times n$:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mn} \end{bmatrix} \tag{2.14}$$

Devido a (2.14) os jogos com dois jogadores de soma não zero também são conhecidos como jogos bimatriciais denotados por $(A, B) = [a_{ij}, b_{ij}]$.

Assim como para jogos de soma zero, jogos de soma não zero possuem estratégias mistas, e a definição é similar a (2.3) e (2.4).

Definição 2.4 (Basar e Olsder, 1982) *Para um jogo de soma não zero com dois jogadores, uma solução de equilíbrio é um par de estratégias com um vetor m -dimensional x^* e um vetor n -dimensional y^* se, para qualquer outra estratégia $x \in X$ e $y \in Y$,*

$$\begin{aligned} x^* A y^* &\geq x A y^* \\ x^* B y^* &\geq x^* B y \end{aligned} \tag{2.15}$$

O par de estratégias (x^*, y^*) é conhecido como uma solução de equilíbrio de Nash para um jogo não cooperativo bimatriacial.

Os jogos com dois jogadores de soma não zero podem ser formulados como um problema de programação bilinear.

Teorema 2.2 (Nishizaki e Sakawa, 2001) *Um par de estratégias (x^*, y^*) constitui uma solução de equilíbrio para um jogo bimatriacial (A, B) se, e somente se, (x^*, y^*, p^*, q^*) é uma solução ótima do seguinte problema de programação bilinear:*

$$\begin{aligned} \max \quad & x A y + x B y - p - q \\ \text{sa : } & A y \leq p e^m \\ & B' x \leq q e^n \\ & \sum_{i=1}^m x_i = 1 \\ & \sum_{j=1}^n y_j = 1 \\ & x_i \geq 0, i = 1, \dots, m. \\ & y_j \geq 0, j = 1, \dots, n. \end{aligned} \tag{2.16}$$

onde e^m e e^n são vetores m - e n -dimensionais cujos elementos são todos a unidade, e p e q são constantes.

2.4.1 Dilema do prisioneiro

Talvez um dos exemplos de jogos de soma não zero mais conhecidos e discutidos na literatura é o dilema do prisioneiro. O dilema do prisioneiro é um exemplo simples e útil para ilustrar jogos de soma não zero e o efeito da cooperação e competição nestas circunstâncias.

O dilema do prisioneiro consiste em dois prisioneiros que serão interrogados e tem, ambos, a opção de ficar calado ou culpar o outro. Dependendo de suas atitudes eles terão um “pagamento”, em anos de prisão, conforme a Tabela 2.2:

	Prisioneiro II fica calado	Prisioneiro II culpa o colega
Prisioneiro I fica calado	(1, 1)	(3, 0)
Prisioneiro I culpa o colega	(0, 3)	(2, 2)

Tab 2.2: Dilema do prisioneiro

onde os valores (a_{ij}, b_{ij}) correspondem aos a_{ij} anos de prisão se o prisioneiro I escolher a opção i e os b_{ij} anos de prisão do jogador II se este escolher a opção j . Na situação da Tabela 2.2 o melhor resultado possível para o primeiro jogador é culpar o outro e este ficar calado. Com esta escolha mesmo se o outro também o culpar, o prisioneiro pega dois anos de prisão. Ou seja, racionalmente, a melhor escolha de ambos, independente da escolha do outro prisioneiro e, admitindo que não pode haver cooperação entre os jogadores, é um culpar o outro. Ao agirem dessa maneira ambos terão dois anos de prisão. Esse exemplo é simples mas ilustra a noção de equilíbrio na teoria dos jogos de soma não zero e a importância da cooperação quando for possível implantá-la. Observe que, se for permitido a cooperação, a melhor estratégia é ambos ficarem calados, pois nesse caso a punição é de um ano de prisão para cada um. Problemas com a característica da Tabela 2.2, onde os pagamentos incentivam a não cooperação para a obtenção de um ganho máximo, podem ser formulados como jogos similares ao dilema do prisioneiro.

Apesar de ilustrativo, o dilema do prisioneiro é um problema de interesse prático, com aplicações em áreas distintas, como por exemplo, em ciências sociais, economia, política, engenharia, relações internacionais, sociologia e biologia.

Dilema do prisioneiro iterativo

O dilema do prisioneiro iterativo é uma extensão do problema do dilema do prisioneiro (Axelrod, 1984), onde os participantes jogam várias vezes mantendo a memória das jogadas anteriores, criando-se uma estratégia para o jogador em cada situação. Axelrod solicitou, a vários acadêmicos, estratégias para o problema visando testá-las em um torneio.

Os resultados do torneio de estratégias para o dilema do prisioneiro encontrados por Axelrod foram surpreendentes. Concluiu-se que uma estratégia muito simples foi a vencedora. A melhor estratégia determinística encontrada foi o *Tit for Tat*, denominação inglesa equivalente à retaliação. Uma expressão aproximada no português seria “olho por olho, dente por dente”. Ou seja, a estratégia que se mostrou mais eficaz no torneio foi àquela baseada na retaliação ao outro jogador. O jogador deve repetir o que o outro jogador fez no movimento anterior. Ou seja, cooperar quando houver cooperação do outro jogador e vice-versa. Axelrod conclui alguns padrões envolvidos nas melhores estratégias, como:

1. seja leal: se o oponente cooperou nas últimas jogadas não deixe de cooperar na jogada atual. Ou seja, continue cooperando depois de várias cooperações mútuas.
2. seja vingativo: não coopere quando seu oponente o trair.
3. aceite desculpas: quando a cooperação for restaurada, continue cooperando.
4. não seja invejoso: não tente ganhar mais do que o seu oponente. Isso leva a traição e contraria a estratégia de ser leal.

Diversos estudos seguiram-se ao torneio de Axelrod, entre eles trabalhos envolvendo jogos com mais de dois jogadores (Yao e Darwen, 2001).

2.5 Jogos cooperativos com k Jogadores

Essa seção apresenta os conceitos sobre jogos cooperativos com n jogadores que são relevantes para esse trabalho. Esse tipo de jogo possui inúmeras aplicações, sobretudo na área de economia. Em um cenário competitivo, pode ser vantajoso que as entidades formem coalizões entre si para conseguirem um resultado superior aquele que conseguiriam sozinhas.

Jogos cooperativos com k jogadores são definidos por um conjunto de jogadores e uma função associando uma coalizão entre os jogadores a um número real.

Definição 2.5: (Nishizaki e Sakawa, 2001) *Seja o conjunto de todos os jogadores denominado por $K = \{1, \dots, k\}$, onde k é um inteiro não negativo. Qualquer sub-conjunto não vazio S de K , incluindo o próprio K ou conjuntos com apenas um elemento, é chamado de coalizão.*

Definição 2.6: (Nishizaki e Sakawa, 2001) *A função v , chamada de função característica de um jogo, é uma função de valor real que associa qualquer coalizão $S \subseteq K$ número real $v(S)$, sendo que $v(\emptyset) = 0$, onde $\emptyset$ representa o conjunto vazio.*

A cooperação é descrita pela função característica do jogo v e $v(S)$ é chamado de valor da coalizão S , é interpretado como o montante total de um pagamento que os jogadores que fazem parte de S podem obter sem ajuda dos demais jogadores, os que não fazem parte de S . Sendo assim, um jogo cooperativo é definido pelo par (K, v) .

Definição 2.7: (Nishizaki e Sakawa, 2001) *Um jogo (K, v) é dito ser superaditivo se, para qualquer par de S e T , tal que $S \cap T = \emptyset$,*

$$v(S \cup T) \geq v(S) + v(T) \quad (2.27)$$

Percebe-se que o valor de v é maior quando a coalizão é formada pela união de S e T do que o valor de v para a coalizão S somado ao valor de v para a coalizão T .

Definição 2.8: (Nishizaki e Sakawa, 2001) Para um jogo (K, v) , um vetor de entrada (inputation) é um vetor de pagamento $x = (x_1, \dots, x_n)$ que satisfaz:

$$x_i \geq v(\{i\}), i = 1, \dots, k$$

(2.28)

$$\sum_{i \in K} x_i = v(K) \quad (2.29)$$

Não é possível para nenhum jogador aumentar seu ganho sem diminuir o pagamento de outros jogadores quando a condição (2.29) é respeitada. Assim, os vetores de pagamento que satisfazem essa condição são chamados de soluções Pareto ótimas.

Definição 2.9: (Nishizaki e Sakawa, 2001) Para um jogo (K, v) , sejam x e y dois vetores de entrada (inputations) e seja $S \subseteq K$ uma coalizão. Dizemos que x domina y , $x \text{ dom}_S y$ na coalizão S se:

$$x_i \geq y_i, \forall i \in S \quad (2.30)$$

$$\sum_{i \in S} x_i = v(S) \quad (2.31)$$

Definição 2.10: (Nishizaki e Sakawa, 2001) O conjunto de todos os vetores de entrada (inputations) não dominados de um jogo (K, v) é chamado de núcleo do jogo $C(K, v)$.

Teorema 2.3: (Nishizaki e Sakawa, 2001) Se um jogo (K, v) é superaditivo, o núcleo do jogo é o conjunto de todos os vetores de pagamentos x que satisfazem as seguintes condições:

$$\sum_{i \in S} x_i \geq v(S), S \subset K, \quad (2.32)$$

$$\sum_{i \in K} x_i = v(K) \quad (2.33)$$

Para um jogo superaditivo (K, v) , pelo fato do núcleo $C(K, v)$ ser representado pelas desigualdades (2.32) e pelas igualdades (2.33), ele é um conjunto convexo e pode ser vazio.

Teorema 2.4: *(Nishizaki e Sakawa, 2001) Para um jogo (K, v) , seja $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ uma solução ótima do problema:*

$$\begin{aligned} \min \quad & \sum_{i \in K} x_i \\ \text{sa} \quad & \sum_{i \in S} x_i \geq v(S), \forall S \subset K \end{aligned} \quad (2.34)$$

Então x^* satisfaz a condição:

$$\sum_{i \in S} x_i^* \leq v(K) \quad (2.35)$$

se e somente se o núcleo $C(K, v)$ não for vazio.

A prova desse Teorema pode ser encontrada em Nishizaki e Sakawa, 2001.

2.6 Jogos nebulosos

A teoria dos jogos modela situações de conflito de interesse e é utilizada na prática em áreas distintas. Entretanto, os dados relativos a problemas de interesse prático nem sempre podem ser adquiridos com precisão. A teoria de conjuntos nebulosos, introduzida por Zadeh (1965) e revista no Capítulo 1, contribui de forma importante no tratamento de imprecisões dos parâmetros dos modelos.

Utilizando os conceitos de conjuntos nebulosos do Capítulo 1 é possível traduzir informações lingüísticas em dados numéricos e isso possibilita o processamento numérico do problema (Pedrycz e Gomide, 1998).

Os jogos estudados nesse trabalho apresentam nebulosidades na matriz de pagamento do jogo, isto é, os elementos da matriz de pagamento são números nebulosos.

Um jogo nebuloso de soma zero com dois jogadores é representado pela matriz nebulosa $m \times n$:

$$\tilde{A} = \begin{bmatrix} \tilde{a}_{11} & \cdots & \tilde{a}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{a}_{m1} & \cdots & \tilde{a}_{mn} \end{bmatrix} \quad (2.36)$$

A matriz $\tilde{A}$ é chamada de matriz de pagamentos nebulosa, e seus elementos $\tilde{a}_{ij}$ são números nebulosos. Uma matriz nebulosa $\tilde{A}$ pode ser representada como $\tilde{A} = [A^L, A, A^R]$, as matrizes dos limites inferior, núcleo, e limite superior, respectivamente.

Para um jogo nebuloso de soma zero com dois jogadores com uma matriz de pagamentos nebulosa $\tilde{A}$, temos :

$$\max_{x \in X} \min_{y \in Y} x \tilde{A} y = \min_{y \in Y} \max_{x \in X} x \tilde{A} y \quad (2.37)$$

e o par de estratégias (x, y) que satisfaz essa condição é uma solução de equilíbrio do jogo nebuloso (Nishizaki e Sakawa, 2001).

Para encontrar um par de estratégias (x, y) que satisfaz a condição (2.37), de forma análoga aos problemas (2.12) e (2.13) para o caso clássico, para o jogo nebuloso temos os seguintes problemas. Para o jogador I:

$$\begin{aligned} \min \quad & s_1 + \dots + s_m \\ \text{sa} : \quad & \tilde{a}_{1j}s_1 + \dots + \tilde{a}_{mj}s_m \geq 1, j = 1, \dots, n \\ & s_i \geq 0, i = 1, \dots, m \end{aligned} \quad (2.38)$$

e para o jogador II, analogamente,

$$\begin{aligned}
 \max \quad & r_1 + \dots + r_n \\
 \text{sa : } & \tilde{a}_{1j}r_1 + \dots + \tilde{a}_{in}r_n \leq 1, i = 1, \dots, m \\
 & r_i \geq 0, i = 1, \dots, n
 \end{aligned} \tag{2.39}$$

onde os sinais $\geq$ e $\leq$ podem ser desigualdades nebulosas.

O valor nebuloso do jogo P é obtido por:

$$P = x\tilde{A}y$$

Diferentemente do caso clássico, para o problema nebuloso não é possível garantir a dualidade entre os problemas (2.38) e (2.39), por se tratar de problemas nebulosos onde, dependendo do método utilizado para resolver os problemas acima, o resultado ótimo para a função objetivo de (2.38) pode nem sempre ser igual ao resultado ótimo de (2.39).

Métodos para a solução dos problemas (2.38) e (2.39) são objetos do Capítulo 3.

De forma análoga, um jogo nebuloso de soma não zero com dois jogadores é representado por duas matrizes nebulosas $\tilde{A} = [a_{ij}]$ e $\tilde{B} = [b_{ij}]$ $m \times n$:

$$\tilde{A} = \begin{bmatrix} \tilde{a}_{11} & \dots & \tilde{a}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{a}_{m1} & \dots & \tilde{a}_{mn} \end{bmatrix} \quad \text{e} \quad \tilde{B} = \begin{bmatrix} \tilde{b}_{11} & \dots & \tilde{b}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{b}_{m1} & \dots & \tilde{b}_{mn} \end{bmatrix} \tag{2.40}$$

e a solução de equilíbrio é um par de estratégias com um vetor m -dimensional x^* e um vetor n -dimensional y^* se, para qualquer outra estratégia $x \in X$ e $y \in Y$ (Nishizaki e Sakawa, 2001). :

$$\begin{aligned}
 x^* \tilde{A} y^* &\geq x \tilde{A} y^* \\
 x^* \tilde{B} y^* &\geq x^* \tilde{B} y
 \end{aligned} \tag{2.41}$$

Segundo Maeda(2000), um ponto $(x^*, y^*) \in X \times Y$ é uma estratégia de equilíbrio de Nash nebulosa se e somente se x^* for uma solução ótima para o seguinte problema linear

nebuloso com parâmetro $y \in Y$:

$$\begin{aligned} \max_x \quad & x \tilde{A} y^* \\ \text{sa : } & x \in X \end{aligned} \quad (2.42)$$

e y^* for uma solução ótima para o seguinte problema linear nebuloso com parâmetro $x \in X$:

$$\begin{aligned} \max_y \quad & x^* \tilde{B} y \\ \text{sa : } & y \in Y \end{aligned} \quad (2.43)$$

Uma vez que os valores nos problemas (2.42) e (2.43) são nebulosos, não existe nenhum conceito de solução ótima, no sentido clássico. Dessa forma, associado aos problemas acima, considere os seguintes problemas com parâmetros $v, w \in \mathbb{R}$ e $x \in X, y \in Y$:

$$\begin{aligned} \max_x \quad & \text{Pos}(x \tilde{A} y^* \geq v) \\ \text{sa : } & x \in X \end{aligned} \quad \text{e} \quad \begin{aligned} \max_y \quad & \text{Pos}(x^* \tilde{B} y \geq w) \\ \text{sa : } & y \in Y \end{aligned} \quad (2.44)$$

Para esse trabalho, utilizamos a relação abaixo (Ramik e Rimanek, 1985) para definir uma desigualdade nebulosa, como a encontrada nos problemas (2.42) e (2.43):

$$\tilde{a} \geq \tilde{b} \text{ sse } (a_\alpha^L, a_\alpha^R) \geq (b_\alpha^L, b_\alpha^R) \quad \forall \alpha \in [0,1] \quad (2.45)$$

Para encontrar a solução de equilíbrio num jogo de soma não zero nebuloso o seguinte problema de programação bilinear deve ser resolvido:

$$\begin{aligned}
\max \quad & x\tilde{A}y + x\tilde{B}y - p - q \\
sa : \quad & \tilde{A}y \leq pe^m \\
& \tilde{B}'x \leq qe^n \\
& \sum_{i=1}^m x_i = 1 \\
& \sum_{j=1}^n y_j = 1 \\
& x_i \geq 0, i = 1, \dots, m. \\
& y_j \geq 0, j = 1, \dots, n.
\end{aligned} \tag{2.46}$$

Um método para a solução do problema (2.46) é sugerido no Capítulo 4. Este método será posteriormente utilizado para obter um par (x^*, y^*) que satisfaça (2.41).

Um jogo nebuloso cooperativo é um jogo cujas matrizes de pagamentos são matrizes nebulosas e onde é permitida a cooperação entre os jogadores. No Capítulo 3 é feita uma análise mais detalhada de um jogo nebuloso cooperativo onde um método de solução (Nishizaki e Sakawa, 1998) é discutido.

2.7 Resumo

Esse capítulo apresentou os principais conceitos sobre a teoria de jogos. De um modo geral, um jogo pode ser classificado em cooperativo ou não cooperativo. Jogos cooperativos são aqueles em que é possível haver parceria entre os jogadores para se obter um resultado melhor para todas as partes. Nos jogos não cooperativos não há comunicação nem acordo entre os jogadores.

Nesse trabalho a ênfase é a jogos não cooperativos, embora nesse capítulo introduziu-se noções gerais sobre jogos cooperativos e no Capítulo 3 discute-se um método de solução.

Jogos não cooperativos podem ocorrer em diferentes formas, dependendo da matriz de pagamentos. Jogos de soma zero com dois jogadores são aqueles em que o que um jogador ganha o outro perde. Jogos de soma não zero não atendem esse requisito. Esse tipo de jogo também é conhecido como jogo bimatricial, quando ocorre entre dois jogadores

Jogos nebulosos, os problemas de interesse desse trabalho, são extensões de jogos clássicos. Jogos nebulosos admitem números nebulosos como elementos da matriz de pagamentos. O próximo capítulo apresenta métodos de solução para jogos nebulosos.

Capítulo 3

Algoritmos

3.1 Introdução

Esse capítulo considera algoritmos para a solução de jogos nebulosos não cooperativos de soma zero e jogos nebulosos cooperativos. Os Capítulos 4 e 5 sugerem algoritmos para jogos nebulosos não cooperativos de soma não zero.

Como vimos no capítulo anterior, a teoria de jogos nebulosos é uma extensão da teoria de jogos clássica em que os valores da matriz de pagamentos e o valor do jogo são números nebulosos. Como o Capítulo 1 mostra, um número nebuloso A é um conjunto nebuloso convexo e normal cujo universo é o conjunto $\mathbb{R}$ dos números reais:

$$A : \mathbb{R} \rightarrow [0,1]$$

Esse capítulo enfatiza jogos nebulosos com dois jogadores onde todos os elementos da matriz de pagamentos são números nebulosos.

3.2 Jogos nebulosos de soma zero

Nessa sessão são estudados quatro modelos para a resolução de jogos nebulosos não cooperativos com dois jogadores de soma zero. Uma comparação entre os modelos é elaborada através da análise de resultados obtidos na implementação e execução dos métodos para a resolução de exemplos numéricos.

Como visto no Capítulo 2, os jogos nebulosos com dois jogadores de soma zero podem ser formulados como (3.1):

$$\begin{aligned}
 \min \quad & \sum_{i=1}^m s_i & \max \quad & \sum_{j=1}^n r_j \\
 sa : \quad & \sum_{i=1}^m \tilde{a}_{ij} s_i \gtrless \tilde{b}, j = 1, \dots, n & sa : \quad & \sum_{j=1}^n \tilde{a}_{ij} r_j \gtrless \tilde{b}, i = 1, \dots, m \\
 & s_i \geq 0, i \in I & & r_j \geq 0, j \in J
 \end{aligned} \tag{3.1}$$

onde $\gtrless$ e $\gtrless$ podem ser desigualdades nebulosas ou não.

Encontrar a solução de equilíbrio do jogo consiste em resolver os problemas (3.1) e encontrar as estratégias ótimas x^* e y^* , conforme (3.2).

$$\begin{aligned}
 s_i &= x_i / z, i \in I \\
 r_j &= y_j / w, j \in J
 \end{aligned} \tag{3.2}$$

3.2.1 Modelo de Campos

Diversos métodos foram propostos para a solução de jogos nebulosos de soma zero, entre eles, a programação linear. O modelo de Campos (1989) utiliza a programação linear para encontrar a estratégia ótima do jogo.

O modelo de Campos considera o problema em (3.1) com incertezas tanto na matriz de pagamentos, como no vetor $\tilde{b}$ e nas desigualdades.

Consideremos primeiro a nebulosidade das restrições. Campo trata a nebulosidade nas desigualdades substituindo as restrições nebulosas por um conjunto convexo:

$$\begin{aligned}
 \tilde{a}s &\geq \tilde{b} + \tilde{p}(1 - \alpha) \\
 \tilde{a}r &\leq \tilde{b} + \tilde{q}(1 - \alpha)
 \end{aligned}$$

onde $\alpha \in [0, 1]$ e $\tilde{p}$ e $\tilde{q}$ são números nebulosos e expressam a violação máxima que os jogadores permitem no atendimento das restrições. Por exemplo, quando $\tilde{p}$ for igual a 0, o valor de α não terá influência alguma na desigualdade e o vetor do lado das restrições será igual a $\tilde{b}$, ou seja, nesse caso 0 os jogadores não permitirão violação no

atendimento das restrições.

A relação $\leq$ é escolhida pelo usuário, desde que preserve a ordenação se os números nebulosos forem multiplicados por um escalar positivo.

Assim, os problemas (3.1) tornam-se:

$$\begin{aligned} \min \quad & \sum_{i=1}^m s_i & \max \quad & \sum_{j=1}^n r_j \\ \text{sa : } & \sum_{i=1}^m \tilde{a}_{ij} s_i \geq \tilde{b} + \tilde{p}(1-\alpha), j = 1, \dots, n & \text{sa : } & \sum_{j=1}^n \tilde{a}_{ij} r_j \leq \tilde{b} + \tilde{q}(1-\alpha), i = 1, \dots, m \\ & s_i \geq 0, i \in I, \alpha \in [0,1] & & r_j \geq 0, j \in J, \alpha \in [0,1] \end{aligned} \quad (3.3)$$

O modelo de Campos (1989) trata os números nebulosos da matriz de pagamentos através da ordenação dos números nebulosos. Qualquer número triangular nebuloso pode ser utilizado, independentemente de ser simétrico ou não. Os elementos nebulosos são definidos através dos limites inferior (L), superior (R) e núcleo.

Conforme o Capítulo 1, existem diferentes métodos para a ordenação de números nebulosos. Campos (1989) sugere cinco métodos de ordenação diferentes, baseados nos métodos citados no Capítulo 1. São eles:

- (a) $[a_i^R + a_i^L + a_i] x \leq b_i^R + b_i^L + b_i + [t_i^R + t_i^L + t_i] (1-\alpha)$
- (b) $[a_i^R + a_i^L + 2a_i] x \leq b_i^R + b_i^L + 2b_i + [t_i^R + t_i^L + 2t_i] (1-\alpha)$
- (c) $[ka_i + (1-k)a_i^R] x \leq kb_i + (1-k)b_i^R + [kt_i + (1-k)t_i^R] (1-\alpha)$
- (d) $a_i x \leq b_i + t_i(1-\alpha)$
- (e) $[a_i + a_i^L] x \leq b_i + b_i^L + [t_i + t_i^L] (1-\alpha)$

Considerando os cinco métodos de ordenação obtemos os seguintes modelos auxiliares para a solução de (3.3). Do método de ordenação (a), para o jogador I temos:

$$\begin{aligned}
 & \min \sum_i s_i \\
 & sa : \sum_i [a_{ij}^R + a_{ij}^L + a_{ij}] s_i \geq 3 - (p^R + p^L + p)(1 - \alpha) \\
 & s_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.4}$$

e para o jogador II:

$$\begin{aligned}
 & \max \sum_i r_i \\
 & sa : \sum_i [a_{ij}^R + a_{ij}^L + a_{ij}] r_i \leq 3 + (q^R + q^L + q)(1 - \alpha) \\
 & r_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.5}$$

Do método de ordenação (b), para o jogador I temos:

$$\begin{aligned}
 & \min \sum_i s_i \\
 & sa : \sum_i [a_{ij}^R + a_{ij}^L + 2a_{ij}] s_i \geq 4 - (p^R + p^L + 2p)(1 - \alpha) \\
 & s_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.6}$$

e para o jogador II:

$$\begin{aligned}
 & \max \sum_i r_i \\
 & sa : \sum_i [a_{ij}^U + a_{ij}^L + 2a_{ij}] r_i \leq 4 + (q^U + q^L + 2q)(1 - \alpha) \\
 & r_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.7}$$

Do método de ordenação (c), para o jogador I temos:

$$\begin{aligned}
 & \min \sum_i s_i \\
 & sa : \sum_i [ka_{ij} + (1-k)a_{ij}^U] s_i \geq 1 - (kp + (1-k)p^U)(1-\alpha) \\
 & s_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.8}$$

e para o jogador II:

$$\begin{aligned}
 & \max \sum_i r_i \\
 & sa : \sum_i [ka_{ij} + (1-k)a_{ij}^U] r_i \leq 1 + (kq + (1-k)q^U)(1-\alpha) \\
 & r_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.9}$$

onde $k \in [0, 1]$ é um valor constante previamente escolhido pelo jogador.

Do método de ordenação (d), para o jogador I temos:

$$\begin{aligned}
 & \min \sum_i s_i \\
 & sa : \sum_i a_{ij} s_i \geq 1 - p(1-\alpha) \\
 & s_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.10}$$

e para o jogador II:

$$\begin{aligned}
 & \max \sum_i r_i \\
 & sa : \sum_i a_{ij} r_i \leq 1 + q(1-\alpha) \\
 & r_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.11}$$

Do método de ordenação (e), para o jogador I temos:

$$\begin{aligned}
 \min \quad & \sum_i s_i \\
 sa : \quad & \sum_i [a_{ij} + a_{ij}^L] s_i \geq 2 - (p + p^L)(1 - \alpha) \\
 & s_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.12}$$

e para o jogador II:

$$\begin{aligned}
 \max \quad & \sum_i r_i \\
 sa : \quad & \sum_i [a_{ij} + a_{ij}^L] r_i \leq 2 + (q + q^L)(1 - \alpha) \\
 & r_i \geq 0, \alpha \in [0, 1]
 \end{aligned} \tag{3.13}$$

Após essas modificações o jogo pode ser resolvido como um problema de programação linear clássico.

Resumindo, o algoritmo correspondente ao modelo de Campos (1989) é o seguinte:

Passo 1: Inicializar

entrar: $\alpha, \tilde{A}, \tilde{b}, \tilde{p}$ e $\tilde{q}$.

Passo 2: Escolher o tipo de ordenação

entrar: tipoOrdenacao

Passo 3: Preparar o modelo de programação linear

se tipoOrdenacao = a então:

$$A = A^R + A^L + A$$

$$b = b^R + b^L + b$$

$$p = p^R + p^L + p$$

$$q = q^R + q^L + q$$

senão se tipoOrdenacao = b então:

$$A = A^R + A^L + 2A$$

$$b = b^R + b^L + 2b$$

$$p = p^R + p^L + 2p$$

$$q = q^R + q^L + 2q$$

senão se tipoOrdenacao = c então:

entrar: k

$$A = k A + (1 - k) A^R$$

$$b = k b + (1 - k) b^R$$

$$p = k p + (1 - k) p^R$$

$$q = k q + (1 - k) q^R$$

senão se tipoOrdenacao = d então:

$$A = A$$

$$b = b$$

$$p = p$$

$$q = q$$

senão se tipoOrdenacao = e então:

$$A = A^L + A$$

$$b = b^L + b$$

$$p = p^L + p$$

$$q = q^L + q$$

Passo 4: Resolver o problema (3.3), substituindo os valores de $\tilde{A}$, $\tilde{b}$, $\tilde{p}$ e $\tilde{q}$ por A , b , p e q encontrados no passo anterior. Esse passo vai resultar nos valores s e r para as variáveis do problema e nos valores z e w para função objetivo, para o jogador I e II, respectivamente

Passo 5: Encontrar as estratégias ótimas, de acordo com (3.2)

$$x^* = s \times z$$

$$y^* = r \times w$$

Passo 6: Calcular o valor do jogo P

$$P = x^* A y^*$$

Podemos perceber que o algoritmo acima transforma, através das funções de ordenação, um problema de programação linear nebuloso em uma família de problemas de programação linear.

3.2.2 Modelo de Maeda

Outro modelo proposto para a resolução de jogos nebulosos de soma zero é o modelo de Maeda (2003). Nele, diferentemente de Campos(1989), os jogos possuem incertezas apenas na matriz de pagamento, cujos elementos são números nebulosos. O autor considera números nebulosos triangulares e simétricos (1.5).

Maeda considera a matriz de pagamentos nebulosa representada por uma matriz M , a matriz dos centros, e a matriz H , com os valores de dispersão, e os parâmetros λ e μ . Para se obter a matriz de pagamentos A a partir de M e H , utiliza:

$$\begin{aligned} A(\lambda) &= M + (I - 2\lambda) H \\ A(\mu) &= M + (I - 2\mu) H \end{aligned} \quad (3.14)$$

onde λ e μ são números reais, determinados pelo usuário, podendo variar de 0 a 1. $A(\lambda)$ é a matriz de pagamentos do jogador I e $A(\mu)$ a matriz de pagamentos do jogador II.

No modelo de Maeda os problemas em (3.1) são reformulados como (3.15):

$$\begin{aligned} \min \quad & \sum_{i=1}^m s_i & \max \quad & \sum_{j=1}^n r_j \\ \text{sa : } & \sum_{i=1}^m a_{ij}(\mu) s_i \geq b, j = 1, \dots, n & \text{sa : } & \sum_{j=1}^n a_{ij}(\lambda) r_j \leq b, i = 1, \dots, m \\ & s_i \geq 0, i \in I & & r_j \geq 0, j \in J \end{aligned} \quad (3.15)$$

Para deixar claro qual a relação entre as matrizes $A(\lambda)$ e $A(\mu)$ com a matriz $\tilde{A}$ considere o seguinte exemplo:

Seja o elemento $\tilde{a}_{ij}$ da matriz de pagamentos nebulosa $\tilde{A}$. Seja $m_{ij} = 5$, $h_{ij} = 1$ e $\tilde{a}_{ij} = (m_{ij}, h_{ij})$. Para $\lambda = 0.2$:

$$a_{ij}(\lambda) = m_{ij} + (1 - 2\lambda)h_{ij}$$

$$a_{ij}(\lambda) = 5 + (1 - 2(0.2))1$$

$$a_{ij}(\lambda) = 5 + 0.6 = 5.6$$

Considere $\lambda = 0.8$:

$$a_{ij}(\lambda) = 5 + (1 - 2(0.8))1$$

$$a_{ij}(\lambda) = 5 - 0.6 = 4.4$$

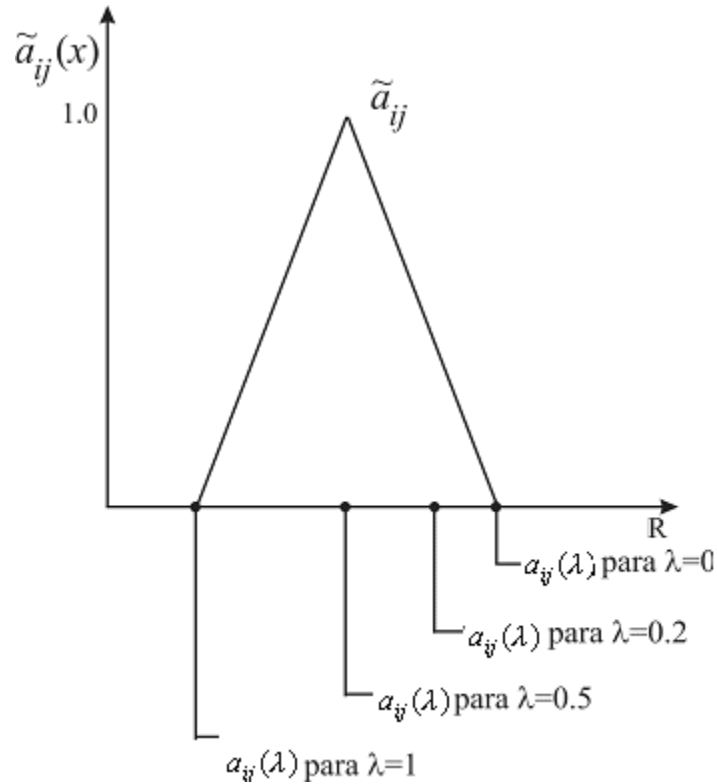


Fig 3.1 Relação entre $A(\lambda)$ e $\tilde{A}$

Fica claro que para valores de λ menores que 0.5, o valor de $a_{ij}(\lambda)$ será maior que m_{ij} . De forma análoga, para valores de λ maiores que 0.5, o valor de $a_{ij}(\lambda)$ será menor que m_{ij} e quando $\lambda = 0.5$, $a_{ij}(\lambda)$ é igual ao núcleo do numero nebuloso $\tilde{a}_{ij}$. A Figura 3.1 ilustra a relação entre $A(\lambda)$ e $\tilde{A}$.

Dessa forma Maeda (2003) transforma o problema original, que agora pode ser resolvido através de qualquer método de programação linear. A definição de equilíbrio do jogo é dada por:

Definição 3.1: (Maeda, 2003) *Seja $\lambda, \mu \in [0,1]$ números reais. A estratégia $(x^*, y^*) \in X \times Y$ é uma estratégia de equilíbrio de Nash para o jogo se:*

$$\begin{aligned} xA(\lambda)y^* &\leq x^*A(\lambda)y^* \quad \forall x \in X \\ x^*A(\mu)y^* &\leq x^*A(\mu)y \quad \forall y \in Y \end{aligned}$$

Como se pode perceber, no modelo de Maeda (2003) as incertezas são tratadas através da parametrização da matriz de pagamentos. Ou seja, é possível ajustar a nebulosidade para cada jogador individualmente, mesmo em um jogo de soma zero. Quando $\lambda = \mu$ temos um jogo com uma única matriz de pagamentos e com isso, os dois jogadores são simétricos e o modelo torna-se semelhante ao de Campos.

O algoritmo de Maeda (2003) pode ser resumido da seguinte forma:

Passo 1: Inicializar

entrar com os valores de M, H, λ e μ .

Passo 2:

calcular as matrizes de pagamentos como em (3.14)

Passo 3:

resolver os problemas (3.15). Esse passo vai resultar nos valores s e r para as variáveis do problema e nos valores z e w para função objetivo, para o jogador I e II,

respectivamente

Passo 4: Encontrar as estratégias ótimas, de acordo com (3.2)

$$x^* = s \ z$$

$$y^* = r \ w$$

Passo 5: Calcular o valor do jogo P

$$P_I = x^* A(\lambda) y^*$$

$$P_{II} = x^* A(\mu) y^*$$

Propriedades dos jogos nebulosos

Maeda (2003) investiga as propriedades dos valores dos jogos nebulosos usando medidas de possibilidade e de necessidade.

Definição 3.2: (Dubois e Prade, 1983) Sejam A e B números nebulosos quaisquer. Define-se as relações abaixo:

- (i) $Pos(A \geq B) = \sup \{ \min(A(x), B(y)) \mid x \geq y \}$,
- (ii) $Pos(A > B) = \sup \{ \inf \{ \min(A(x), 1 - B(y)) \mid x \leq y \} \}$,
- (iii) $Nec(A \geq B) = \inf \{ \sup \{ \max(1 - A(x), B(y)) \mid x \geq y \} \}$,
- (iv) $Nec(A > B) = \inf \{ \max(1 - A(x), 1 - B(y)) \mid x \leq y \}$.

Teorema 3.1: (Sakawa e Yano, 1991) Seja A e B números nebulosos triangulares quaisquer e $\alpha \in (0, 1]$ um número real. Então temos as seguintes relações:

- (i) $Pos(A \geq B) \geq \alpha$ se e somente se $A_\alpha^R \geq B_\alpha^L$,
- (ii) $Pos(A \geq B) \leq \alpha$ se e somente se $A_\alpha^R \leq B_\alpha^L$,
- (iii) $Nec(A \geq B) \geq \alpha$ se e somente se $A_{1-\alpha}^L \geq B_\alpha^L$,
- (iv) $Nec(A \geq B) \leq \alpha$ se e somente se $A_{1-\alpha}^L \leq B_\alpha^L$.

As Figuras 3.2 e 3.3 ilustram, respectivamente, medidas de possibilidade e de

necessidade entre dois números nebulosos.

Exemplo 3.1: (i) $Pos(A \geq B) \geq \alpha$ sse $A_{\alpha}^R \geq B_{\alpha}^L$

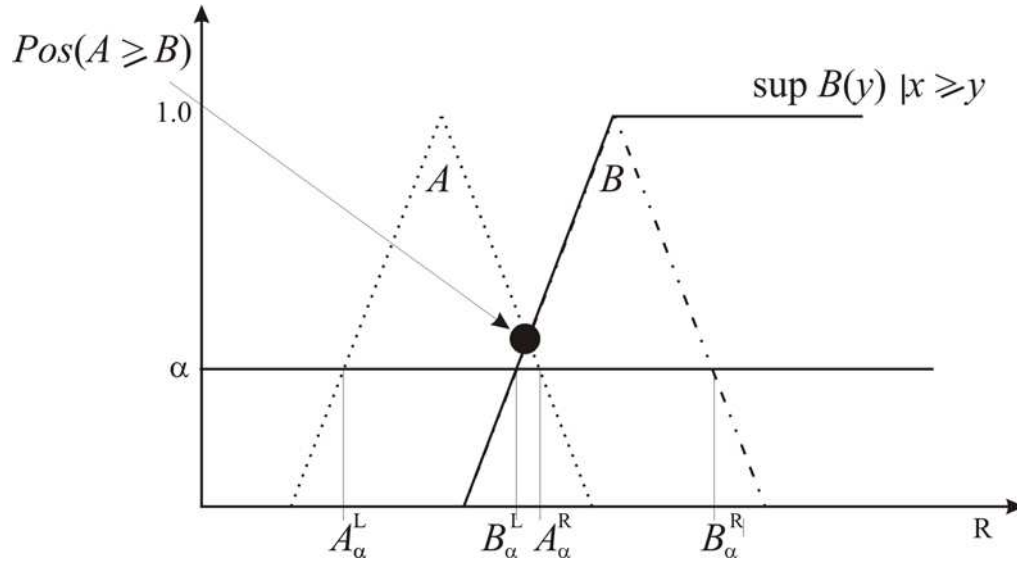


Fig 3.2: Representação da medida de possibilidade entre dois números nebulosos

Exemplo 3.2: (iii) $Nec(A \geq B) \geq \alpha$ sse $A_{1-\alpha}^L \geq B_{\alpha}^L$,

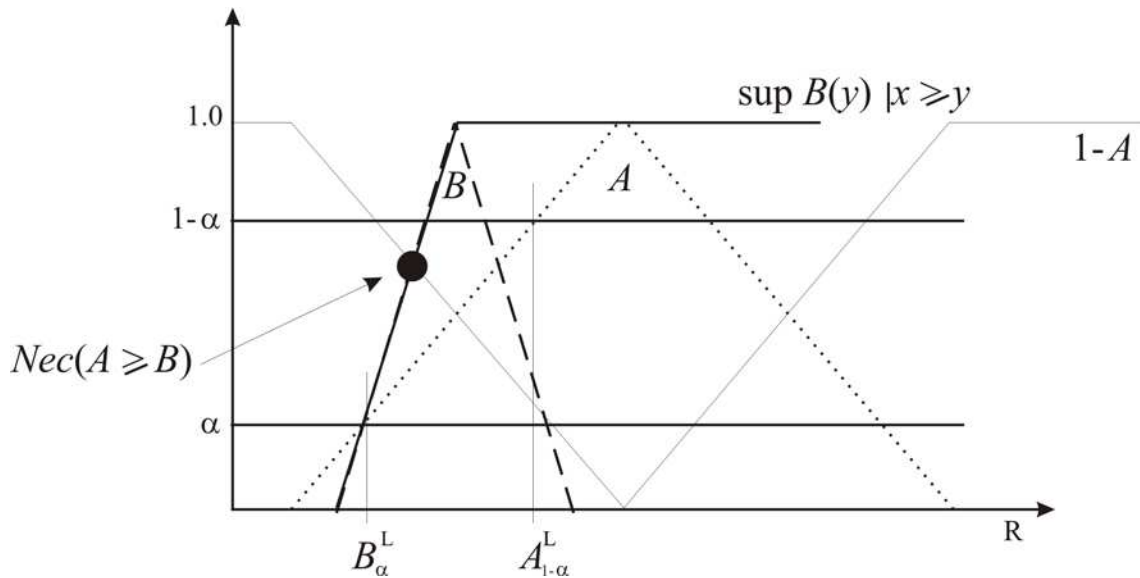


Fig 3.3: Representação da medida de necessidade entre dois números nebulosos

Seja $(x^*, y^*) \in X \times Y$ uma estratégia de equilíbrio para o jogo. Pela definição 3.1

existem números reais $\lambda, \mu \in [0,1]$ tal que:

$$\begin{aligned} x^*(A + (1 - 2\lambda) H)y^* &\geq x(A + (1 - 2\lambda) H)y^* \quad \forall x \in X \\ x^*(A + (1 - 2\mu) H)y^* &\leq x^*(A + (1 - 2\mu) H)y \quad \forall y \in Y \end{aligned}$$

Fazendo $v^* = x^*(A + (1 - 2\lambda) H)y^*$ e $w^* = x^*(A + (1 - 2\mu) H)y^*$. O problema de encontrar a estratégia de equilíbrio pode ser formulado em função das medidas de possibilidade e necessidade, da seguinte forma:

$$\begin{array}{ll} \max & Pos(xAy^* \geq v^*) \\ sa : & x \in X \end{array} \quad \begin{array}{ll} \min & Pos(x^*Ay \geq w^*) \\ sa : & y \in Y \end{array}$$

Temos, portanto, que a estratégia x^* maximiza a possibilidade que o pagamento nebuloso esperado xAy^* seja maior ou igual a v^* , dada a estratégia y^* do jogador II. Por outro lado, a estratégia y^* minimiza a possibilidade que o pagamento nebuloso esperado de x^*Ay seja maior ou igual a w^* , dada a estratégia do jogador I.

3.2.3 Modelo de ordenação

Uma outra maneira de encontrar uma solução de equilíbrio de um jogo nebuloso de soma zero baseia-se no método de ordenação de números nebulosos proposto por Lee-Kwang e Lee (1999).

Pelo fato de números nebulosos representarem valores imprecisos, o problema de ordenar números nebulosos torna-se delicado, conforme discutido no Capítulo 1. Em muitos casos é impossível dizer, mesmo intuitivamente, se um número nebuloso é maior ou menor que outro.

Lee-Kwang e Lee (1999) propõem uma maneira diferente de ordenar números nebulosos considerando diferentes pontos de vista. Dessa maneira, é possível acrescentar à ordenação pontos de vista, que traduzem preferências. Além disso, Lee-Kwang e Lee (1999) sugerem que jogos nebulosos sejam resolvidos através desse método de ordenação.

Nesse modelo, os números nebulosos são ordenados pelo método de Yager (1981), acrescentando-se a noção de ponto de vista, que pode ser otimista ou pessimista.

Conforme Capítulo 1, o método de ordenação proposto por Yager consiste em mapear um número nebuloso na reta real de acordo com:

$$F(A) = \frac{\int_{A^L}^{A^R} g(x)A(x)dx}{\int_{A^L}^{A^R} A(x)dx}$$

onde A^R e A^L são os limites superior e inferior do suporte de A e $g(x)$ é uma função de peso que descreve a importância do valor x .

Os pontos de vista considerados podem ser constante, otimistas ou pessimistas, conforme a Figura 3.4, onde $g(x) = v(x)$.

Percebe-se que o ponto de vista mostrado na Figura 3.4 como otimista é referente a uma entidade que deseja maximizar seus ganhos. Ou seja, para o ponto de vista otimista, quanto maior for o valor de x , melhor. Para uma entidade que deseje minimizar despesas, o ponto de vista otimista seria o oposto, ou seja, o interesse é maior para valores menores.

Dessa forma, o ponto de vista é também representado por um número nebuloso e entra no cálculo da ordenação dos números nebulosos através de uma operação envolvendo uma t-norma e as funções de pertinência dos números nebulosos.

O modelo de Lee-Kwang e Lee faz a ordenação dos números nebulosos utilizando uma função de satisfação que compara dois números nebulosos e retorna um número real entre 0 e 1 como resultado da comparação:

$$F(A < B) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^y A(x) \otimes B(y) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(x) \otimes B(y) dx dy}$$

$$F(A > B) = \frac{\int_{-\infty}^{\infty} \int_y^{\infty} A(x) \otimes B(y) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(x) \otimes B(y) dx dy}$$

onde $\otimes$ é uma t-norma.

A função $F(A < B)$ representa o quanto o número nebuloso A é menor que B . De maneira similar, a função $F(A > B)$ representa o quanto A é maior que B . A primeira função é a integração da t-norma dos números nebulosos na área em que A é menor que B dividido pela área total da relação. A segunda função é similar, para a área em que A é maior que B .

Pontos de Vista

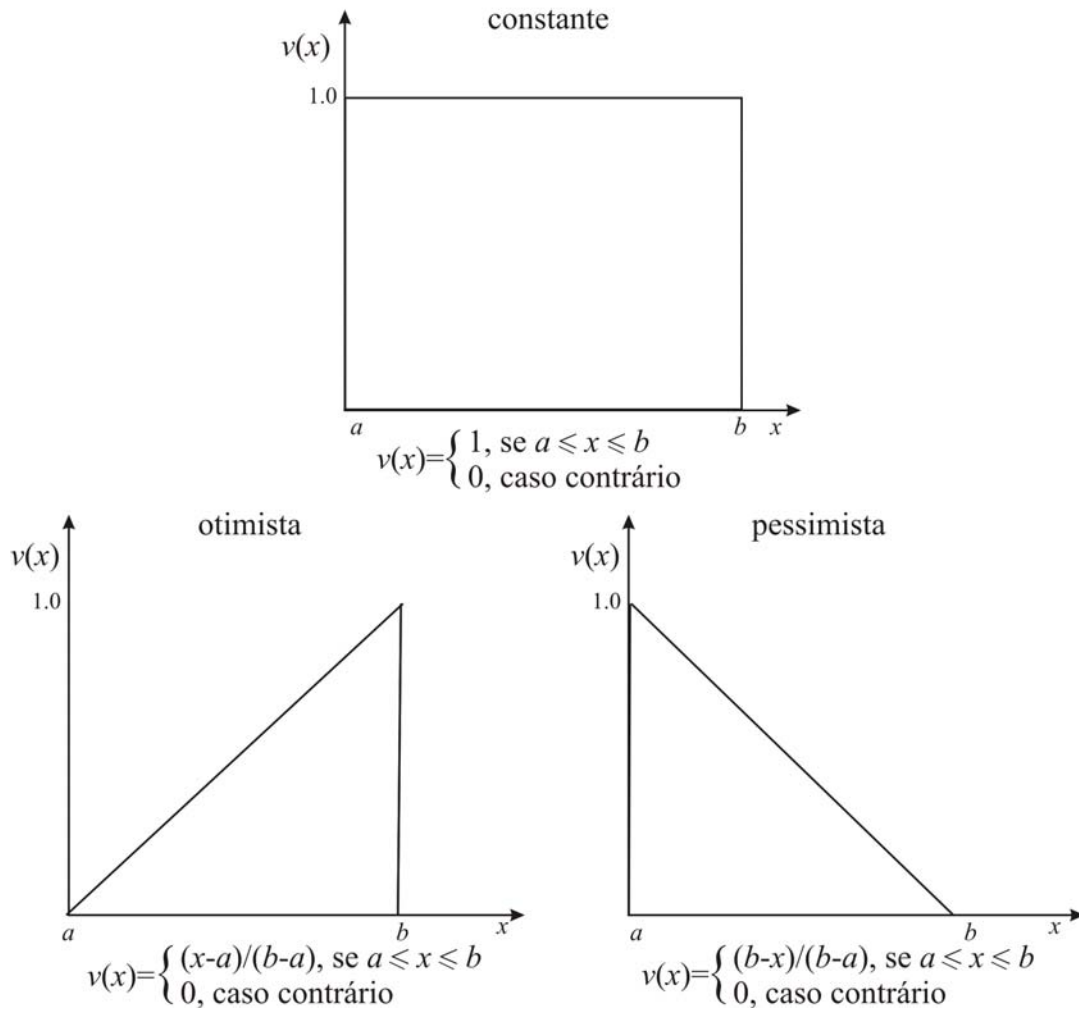


Fig 3.4: Representação dos pontos de vista constante, otimista e pessimista.

Para um ponto de vista v e um conjunto de números nebulosos $\mathfrak{F}$, o valor de avaliação é definido como:

$$E_v(A) = F(A > v)$$

Para um ponto de vista v e um conjunto de números nebulosos $\mathfrak{F}$ o índice relativo de $A \in \mathfrak{F}$ sob o ponto de vista v , $R_v(A)$ é definido por:

$$R_v(A) = \frac{E_v(A)}{\max_{F \in \mathfrak{F}}(E_v(F))}$$

Para ilustrar o método de ordenação, considere os números nebulosos da Figura 3.5. Levando-se em conta o ponto de vista constante, os valores da função de avaliação e do índice relativo ficam como mostra a Tabela 3.1.

	F_1	F_2	F_3	F_4
Ev_1	0.9	0.55	0.65	0.35
Rv_1	1	0.61	0.72	0.38

Tab 3.1 Valores da função de ordenação

O maior número é F_1 , com um índice relativo igual a 1. F_4 é o menor número, com um índice relativo de 0.38. Ou seja, comparando F_1 e F_4 , nesse mesmo ponto de vista, F_4 pode ser considerado 38% do tamanho de F_1 .

Se o ponto de vista for otimista, como mostra a Figura 3.5, os valores da função de avaliação e do índice relativo são os apresentados na Tabela 3.2.

	F_1	F_2	F_3	F_4
Ev_2	0.81	0.34	0.42	0.16
Rv_2	1	0.42	0.52	0.20

Tab 3.2 Valores de ordenação do ponto de vista pessimista

Sob o ponto de vista otimista, valores maiores tem preferência sobre os menores. Comparando novamente F_1 e F_4 sob este ponto de vista, F_4 pode ser considerado 20% do tamanho de F_1 .

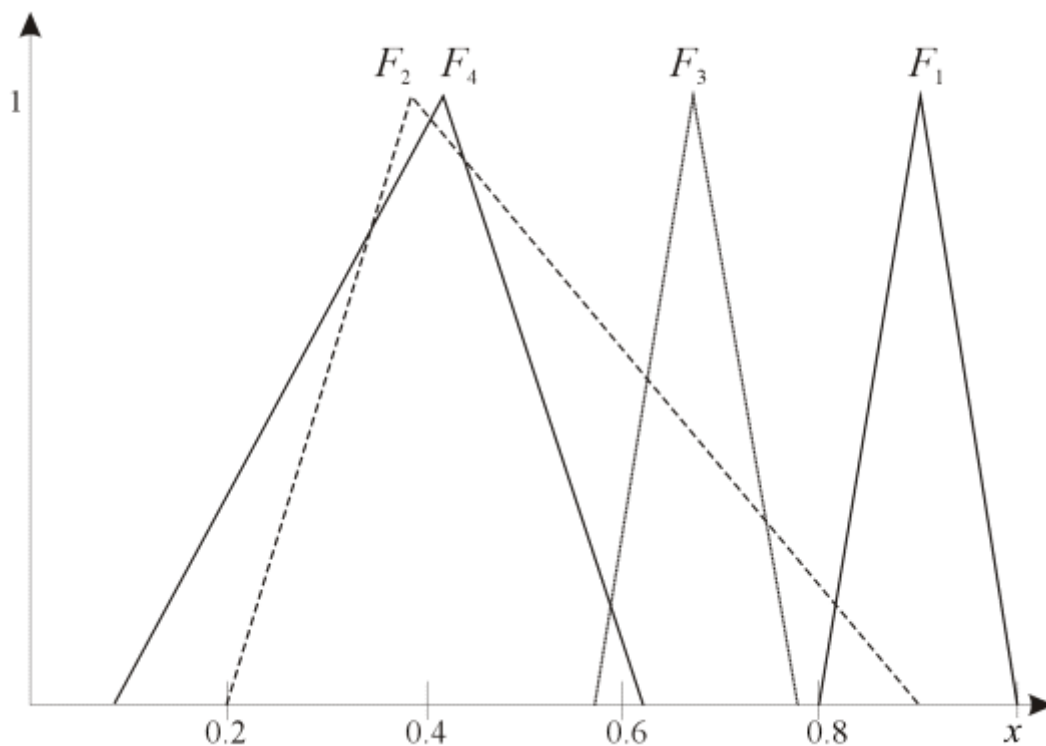


Fig 3.5: Números nebulosos

Considere agora os números nebulosos F_1 e F_2 , Figura 3.6.

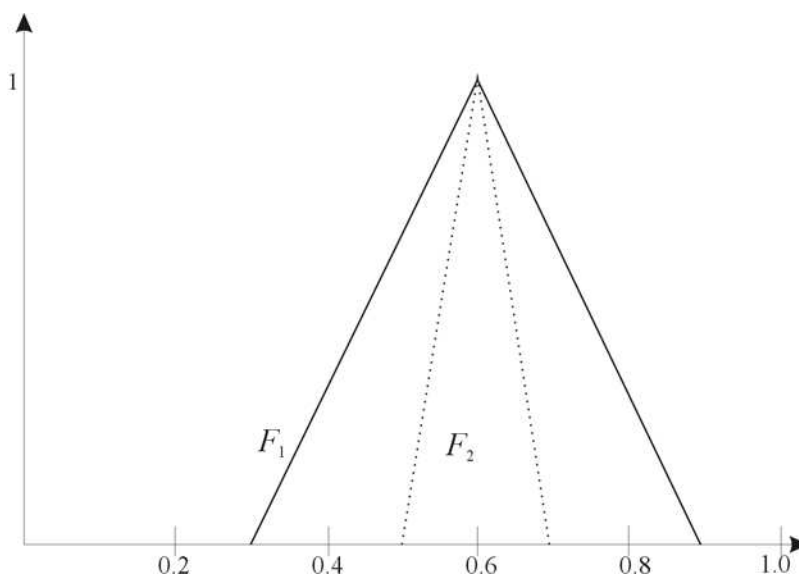


Fig 3.6: Comparação entre dois números nebulosos

Os respectivos valores de índice relativo são mostrados na Tabela 3.3.

	F_1	F_2
$Rv_{otimista}$	1	0.77
$Rv_{pessimista}$	0.89	1
Rv_{neutro}	1	1

Tab 3.3: Valores de índice relativo

Nesse exemplo F_2 está incluído em F_1 e, portanto não fica claro qual número é maior que o outro. Em um ponto de vista pessimista, F_2 é considerado como maior que F_1 porque F_1 inclui valores menores que F_2 . No entanto, do ponto de vista otimista F_1 é considerado como maior número porque contém valores maiores que F_2 .

Com o exemplo da Figura 3.6 é possível perceber o papel importante da inclusão do ponto de vista do usuário na ordenação de números nebulosos.

O método de Lee-Kwang e Lee (1999) consiste, portanto em ordenar os números nebulosos da matriz de pagamentos e criar uma matriz de índices relativos, que será a matriz utilizada para encontrar a solução de equilíbrio do jogo.

3.2.4 Modelo de α -cortes

Outro modelo estudado nesse capítulo para a solução de jogos nebulosos de soma zero é baseado em α -cortes (Chen e Larbani, 2005).

Esse modelo é adaptado neste trabalho para resolver também jogos nebulosos de soma não zero, e o algoritmo proposto é apresentado no Capítulo 4.

Como visto no Capítulo 1, o α -corte de um número nebuloso $\tilde{a}$ é definido por:

$$\tilde{a}_\alpha = \{x \mid \tilde{a}(x) \geq \alpha\}$$

que por sua vez pode ser representado por $[a_\alpha^L, a_\alpha^R]$.

Dessa forma, é possível representar a matriz nebulosa $\tilde{A}$ em função de seu α -corte. Dessa maneira, $\tilde{A}_\alpha = [A_\alpha^L, A_\alpha^R]$, onde A_α^L e A_α^R são matrizes cujos elementos são o ínfimo e o supremo dos números nebulosos $\tilde{a}_{ij}$, elementos da matriz nebulosa $\tilde{A}$.

O algoritmo de α -cortes consiste em transformar os problemas (3.1) nos problemas (3.16):

$$\begin{aligned}
 \min \quad & \sum_{i=1}^m s_i & \max \quad & \sum_{j=1}^n r_j \\
 sa : \quad & \sum_{i=1}^m A_{\alpha}^R s_i \geq 1 & sa : \quad & \sum_{j=1}^n A_{\alpha}^R r_j \leq 1 \\
 & \sum_{i=1}^m A_{\alpha}^L s_i \geq 1 & & \sum_{j=1}^n A_{\alpha}^L r_j \leq 1 \\
 & s_i \geq 0, i \in I & & r_j \geq 0, j \in J
 \end{aligned} \tag{3.16}$$

Os problemas (3.16) não possuem mais a nebulosidade em suas matrizes de pagamentos podendo, portanto, ser resolvidos através de um método de programação linear.

3.2.5 Exemplos

Os modelos de Campos, de Maeda, de Ordenação e de α -corte foram implementados utilizando-se o Matlab e o Xpress. Os exemplos numéricos aqui considerados são encontrados em Campos (1989) e Chen e Larbani (2005).

Exemplo 1

O primeiro exemplo foi estudado por Campos (1989) e também utilizado por Maeda (2003). Inicialmente, supomos a matriz de pagamentos (3.17).

$$A = \begin{bmatrix} 180 & 156 \\ 90 & 180 \end{bmatrix} \tag{3.17}$$

Considerando que os elementos da matriz A (3.17) são números reais, o jogo clássico pode ser formulado como:

$$\begin{array}{ll}
 \min & s_1 + s_2 \\
 \text{sa :} & 180s_1 + 90s_2 \geq 1 \\
 & 156s_1 + 180s_2 \geq 1 \\
 & s_i \geq 0 \\
 \text{e} & \\
 \max & r_1 + r_2 \\
 \text{sa :} & 180r_1 + 156r_2 \leq 1 \\
 & 90r_1 + 180r_2 \leq 1 \\
 & r_i \geq 0
 \end{array}$$

e o resultado do jogo é $x^* = (0.7895, 0.2105)$, $y^* = (0.2105, 0.7895)$ e $P = 161.0526$.

Como vimos, o modelo de Maeda (2003) requer, como entrada, a matriz de núcleos A e a matriz de dispersão H . Primeiramente, considere a matriz M igual a (3.17) e a matriz de dispersão seguinte (3.18)

$$H = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} \quad (3.18)$$

Vários valores foram utilizados para os parâmetros de λ e μ . Em todos os casos o algoritmo convergiu para a solução de equilíbrio, $x^* = (0.7895, 0.2105)$ e $y^* = (0.2105, 0.7895)$ e P em torno de 161 (Figura 3.7).

Os valores de λ e μ foram testados de 0 a 1 e em todos os casos estudados o resultado de x^* e y^* obtido foi o mesmo. Isso porque, mesmo com as modificações na matriz de pagamentos, as decisões mais vantajosas para cada jogador, ou seja, a estratégia de equilíbrio, continua sendo a mesma. Já o valor do jogo P varia de acordo com os valores de λ e μ . pois como vimos:

$$P = x\tilde{A}y$$

ou seja, para as matrizes parametrizadas, temos:

$$P_I = xA(\lambda)y \quad \text{e} \quad P_{II} = xA(\mu)y$$

ou seja, o calculo de P leva em conta os valores da matriz de pagamentos

parametrizada.

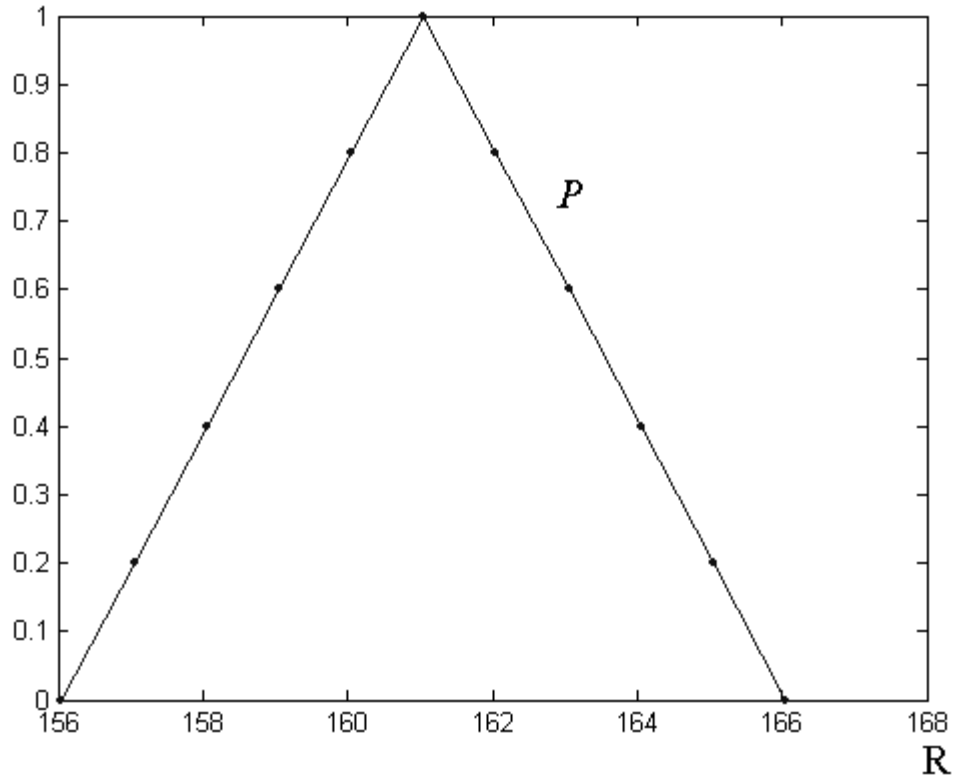


Fig 3.7 Valor do jogo no exemplo 1 pelo modelo de Maeda

Podemos perceber que a nebulosidade acrescentada ao problema nesse caso não afetou de maneira significativa os resultados finais. As soluções de equilíbrio x^* e y^* são as mesmas. Diante desse fato, surgiu a necessidade de verificar como os valores de dispersão dos elementos da matriz de pagamentos influencia o valor do jogo. Para isso, a matriz de dispersão H foi modificada.

Foram testados valores de dispersões mais altos, respectivamente 10, 50 e 100 conforme (3.19). Em todos os casos os valores de x^* e y^* foram os mesmos, variando somente a dispersão do número nebuloso P . O valor do jogo varia conforme a dispersão dos elementos da matriz de pagamentos. Ou seja, se a imprecisão dos números nebulosos triangulares que compõem a matriz de pagamentos for alterada uniformemente, então a solução de equilíbrio é a mesma, aumentando apenas a nebulosidade do valor do jogo, parametrizado por λ e μ .

$$H1 = \begin{bmatrix} 10 & 10 \\ 10 & 10 \end{bmatrix}, \quad H2 = \begin{bmatrix} 50 & 50 \\ 50 & 50 \end{bmatrix}, \quad H3 = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix} \quad (3.19)$$

A Figura 3.8 ilustra o exemplo para a matriz A dada em (3.17) e para as matrizes em (3.19). Os valores $P1$, $P2$ e $P3$ são os valores do jogo para as matrizes $H1$, $H2$ e $H3$, respectivamente:

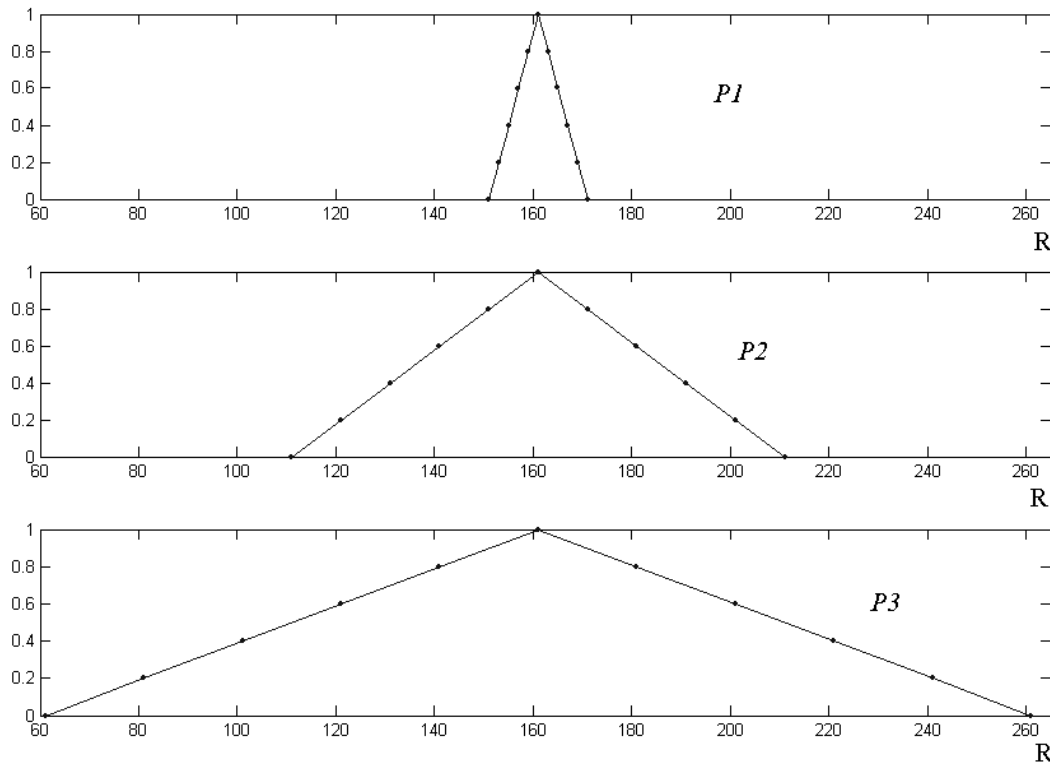


Fig 3.8 Valor do jogo para diferentes dispersões na matriz de pagamentos

Se, no entanto, variações diferentes comporem os elementos da matriz H , isto é, diferentes valores de dispersão para cada elemento da matriz, o resultado do jogo se altera. Por exemplo, considere a seguinte matriz dispersão (3.20).

$$H = \begin{bmatrix} 5 & 10 \\ 5 & 5 \end{bmatrix} \quad (3.20)$$

Neste caso, a solução de equilíbrio varia de acordo com os valores adotados para λ e

μ . Neste exemplo o elemento $a_{12} = 10$ tornou-se vantajoso para o jogador I, maximizador, valores mais baixos de λ , e ao mesmo tempo tornou-se uma opção de risco para o jogador II, valores de μ abaixo de 0.5.

Para H em (3.20), as soluções de equilíbrio variam conforme os valores dos parâmetros λ e μ . são alterados, conforme ilustra a Tabela 3.4.

Situação	λ	μ	x^*	y^*
1	0.3	0.2	0.8036, 0.1964	0.1892, 0.8108
2	0.3	0.9	0.8036, 0.1964	0.2373, 0.7627
3	0.8	0.3	0.7692, 0.2308	0.1964, 0.8036

Tab 3.4 Matriz H assimétrica no modelo de Maeda

Analisando os resultados da Tabela 3.4, na primeira situação os parâmetros μ e λ possuem valores abaixo de 0.5. Essa é uma vantagem para o jogador I porque será utilizada uma dispersão do número nebuloso mais a direita, como já foi mencionado e ilustrado na Figura 3.2. Como o jogador I quer maximizar seu resultado final, torna-se interessante apostar mais na primeira opção. E assim, a estratégia ótima para o jogador I é de $x^* = (0.8036, 0.1964)$.

No segundo caso, o valor de μ aumenta e como o valor de μ faz a parametrização da matriz de pagamentos para o segundo jogador, o valor de x não muda. Já o valor de y se altera de maneira semelhante ao primeiro caso. Como agora será utilizada uma dispersão mais a esquerda (ver Figura 3.1), um número menor, torna-se menos arriscado apostar na primeira opção, e a estratégia do segundo jogador se altera para $y^* = (0.2373, 0.7627)$. A terceira situação é análoga às demais.

Variando-se os valores dos parâmetros μ e λ de 0 a 1 é possível obter o valor nebuloso P do jogo, que continua em torno de 161. A Figura 3.9 ilustra o valor do jogo para o exemplo (3.20).

O método encontra a solução de equilíbrio em todas as situações testadas, independentemente da nebulosidade do problema, ajustando a estratégia ótima de acordo com o cenário estudado.

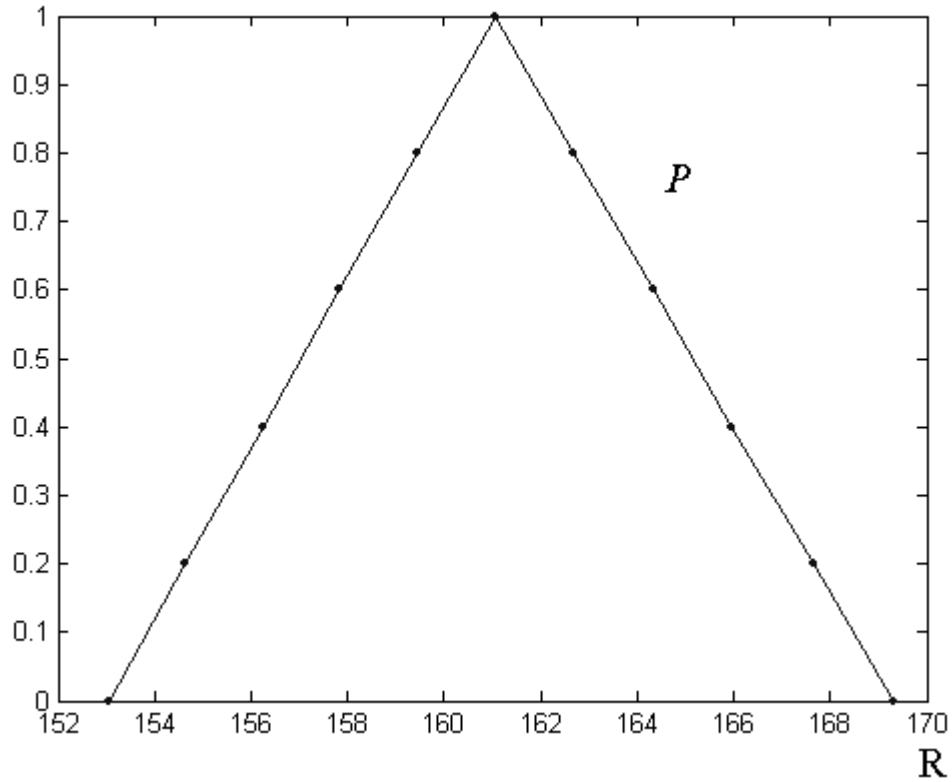


Fig 3.9 Valor do jogo para matriz H assimétrica

Os resultados mostram a eficiência do modelo de Maeda e a importância da parametrização e da dispersão da matriz de pagamentos no resultado final do jogo.

No modelo de Campos os valores de $\tilde{A}$, $\tilde{b}$, $\tilde{p}$ e $\tilde{q}$ utilizados são,

$$A = \begin{bmatrix} 180 & 156 \\ 90 & 180 \end{bmatrix} \quad A^L = \begin{bmatrix} 175 & 151 \\ 85 & 175 \end{bmatrix} \quad A^R = \begin{bmatrix} 185 & 161 \\ 95 & 185 \end{bmatrix} \quad (3.21)$$

$$\tilde{b} = (1, 1, 1) \quad \tilde{p} = (0.10, 0.08, 0.11) \quad \tilde{q} = (0.15, 0.14, 0.17)$$

onde A , A^L e A^R são, respectivamente, a matriz núcleo, limite inferior e superior dos números nebulosos.

Os cinco métodos de ordenação foram implementados. Nesse caso o algoritmo também convergiu em todas as situações testadas, comprovando a adequação do método para encontrar a solução de equilíbrio para os jogos nebulosos de soma zero.

Os resultados obtidos foram $x^* = (0.7895, 0.2105)$ e $y^* = (0.2105, 0.7895)$. O valor do jogo P varia de acordo com o método de ordenação escolhido, conforme as figuras a seguir.

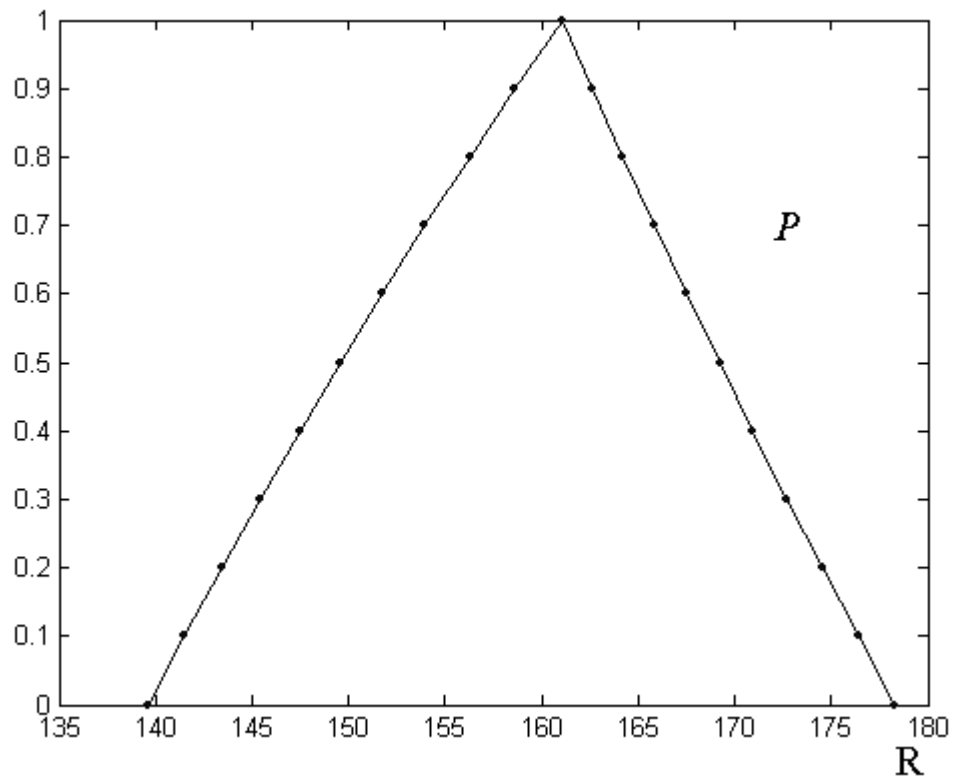


Fig 3.10 Valor do jogo no exemplo 1 pelo modelo de Campos: ordenação 1, 2 e 4.

Os valores de P para os métodos de ordenação 1, 2 e 4 ficaram em torno de 160, como ilustra a Figura 3.10.

Para o método de ordenação 3 o valor de P fica em torno de 162, Figura 3.11.

Para o quinto e último método de ordenação o valor do jogo fica em torno de 158, Figura 3.12.

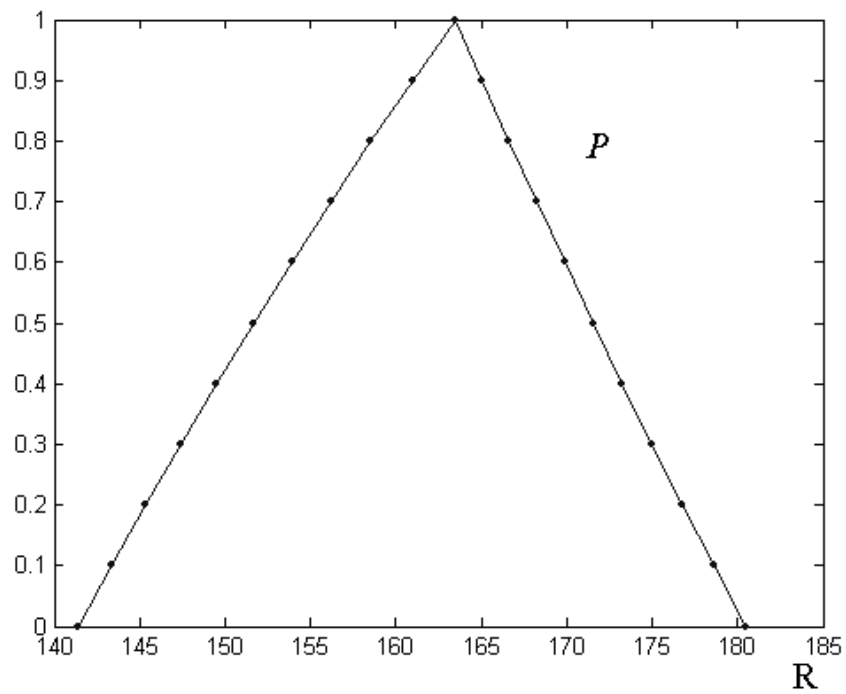


Fig 3.11 Valor do jogo no exemplo 1 pelo modelo de Campos: ordenação 3

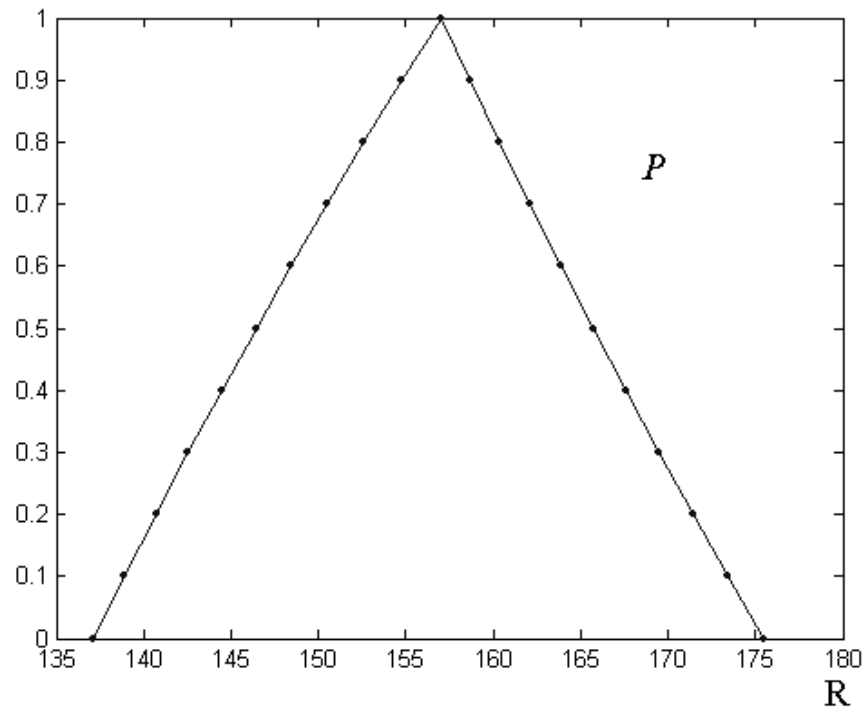


Fig 3.12 Valor do jogo no exemplo 1 pelo modelo de Campos: ordenação 5

Alterações na nebulosidade dos elementos da matriz de pagamentos afetam a

estratégia ótima, similarmente ao modelo de Maeda. Por exemplo, considere a nova matriz de pagamentos nebulosa (3.22).

$$A = \begin{bmatrix} 180 & 156 \\ 90 & 180 \end{bmatrix} \quad A^L = \begin{bmatrix} 175 & 146 \\ 85 & 175 \end{bmatrix} \quad A^R = \begin{bmatrix} 185 & 166 \\ 95 & 185 \end{bmatrix} \quad (3.22)$$

A alteração na nebulosidade da matriz em (3.22) é a equivalente àquela em (3.20) para o modelo de Maeda. Nesse caso, a estratégia de equilíbrio para cada método de ordenação no modelo de Campos foi a seguinte, Tabela 3.5:

ordenação	x^*	y^*
1,2 e 4	0.7895, 0.2105	0.2105, 0.7895
3	0.8000, 0.2000	0.2000, 0.8000
5	0.7725, 0.2275	0.2275, 0.7725

Tab 3.5: Exemplo 1 - resultados pelo modelo de Campos

É possível analisar os resultados obtidos na situação (3.21) e (3.22) com um estudo mais detalhado dos cinco métodos de ordenação propostos no modelo de Campos.

Para o primeiro e o segundo, a ordenação é feita utilizando os limites inferior e superior do número nebuloso. Assim, toda vez que a nebulosidade do número nebuloso for alterada, mas não houver mudança na simetria do número nebuloso, ou seja, quando forem utilizados números nebulosos triangulares simétricos, a estratégia do jogo é a mesma, alterando apenas a nebulosidade do valor do jogo, como ocorreu no modelo de Maeda.

Para o método de ordenação quatro, como se considera apenas os núcleos dos números nebulosos, alterações na nebulosidade não afetam o resultado, nem para a estratégia e nem para o valor de equilíbrio do jogo. Apenas quando o núcleo do número nebuloso for alterado é que a estratégia e o valor de equilíbrio do jogo sofrem alterações.

Já para o terceiro e quinto método, que trabalham apenas com o limite superior e inferior, respectivamente, qualquer alteração no número nebuloso, simétrica ou não, altera a estratégia ótima e o valor do jogo.

Podemos perceber que as alterações na nebulosidade da matriz de pagamentos alteram diferentemente a estratégia de equilíbrio em cada método estudado até agora. O modelo de ordenação de Lee-Kwang e Lee (1999) com a matriz de pagamentos nebulosa

em (3.19) fornece os seguintes resultados, Tabela 3.6:

x^*	y^*	P
0.7120, 0.2880	0.2880, 0.7120	160.3687

Tab 3.6: Exemplo 1 - resultados pelo método de Lee-Kwang e Lee

Nesse método o valor do jogo P é um número real e não um número nebuloso.

Procedendo com as mesmas alterações na matriz de pagamentos nebulosa como em (3.22), a estratégia de equilíbrio sofreu uma leve alteração, na mesma direção que os demais métodos, como mostra a Tabela 3.7.

x^*	y^*	P
0.7133, 0.2867	0.2867, 0.7133	160.3911

Tab 3.7: Exemplo 1 – alteração na matriz de pagamentos

Observe que x_1 passou a ser uma decisão mais interessante para o jogador I e y_1 , agora é menos interessante para o jogador II.

No método de α -cortes, o exemplo 1 para a matriz núcleo A e matriz dispersão H (3.23).

$$A = \begin{bmatrix} 180 & 156 \\ 90 & 180 \end{bmatrix}, H = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} \quad (3.23)$$

resultou na estratégia de equilíbrio $x^* = (0.7895, 0.2105)$ e $y^* = (0.2105, 0.7895)$, independente o valor de α .

Alterando a matriz de pagamentos para (3.24).

$$A = \begin{bmatrix} 180 & 156 \\ 90 & 180 \end{bmatrix}, H = \begin{bmatrix} 5 & 10 \\ 5 & 5 \end{bmatrix} \quad (3.24)$$

Para esse caso, a estratégia ótima variou de acordo com o valor de α , de acordo com a Tabela 3.8.

α	x^*	y^*	P
0.1	0.7563, 0.2437	0.1743, 0.8257	161.1896
0.2	0.7627, 0.2373	0.1818, 0.8182	161.1402
0.3	0.7692, 0.2308	0.1892, 0.8108	161.1019
0.4	0.7692, 0.2308	0.1892, 0.8108	161.1019
0.5	0.7692, 0.2308	0.1892, 0.8108	161.1019
0.6	0.7759, 0.2241	0.1964, 0.8036	161.0745
0.7	0.7826, 0.2174	0.2035, 0.7965	161.0581
0.8	0.7826, 0.2174	0.2035, 0.7965	161.0581
0.9	0.7826, 0.2174	0.2035, 0.7965	161.0581
1	0.7895, 0.2105	0.2105, 0.7895	161.0526

Tab 3.8: Exemplo 1, modelo de α -corte

Exemplo 2

O segundo exemplo é proveniente de Chen e Larbani (2005). No caso, o núcleo da matriz de pagamentos nebulosa é uma matriz 4×5 (3.25).

$$A = \begin{bmatrix} 3.5 & 2.5 & 9 & 7.5 & 5 \\ 7 & 4 & 2.5 & 5.5 & 3 \\ 2 & 8 & 3 & 5 & 2.5 \\ 4 & 5 & 6 & 4 & 6.5 \end{bmatrix} \quad (3.25)$$

e a matriz dispersão é (3.26).

$$H = \begin{bmatrix} 1.5 & 1.5 & 1 & 1.5 & 1 \\ 1 & 1 & 1.5 & 1.5 & 1 \\ 1 & 1 & 2 & 1 & 1.5 \\ 2 & 1 & 1 & 1 & 1.5 \end{bmatrix} \quad (3.26)$$

O modelo de Campos resulta nos valores da Tabela 3.9.

O valor do jogo fica em torno de 4.8. A Figura 3.13 ilustra o valor do jogo para os métodos de ordenação um, dois e quatro. As Figuras 3.14 e 3.15 ilustram, respectivamente, o valor do jogo para a ordenação três e cinco.

Ordenação	x^*	y^*
1	0.05, 0.32, 0.08, 0.53	0.30, 0.36, 0.20, 0.12, 0
2	0.05, 0.32, 0.08, 0.53	0.30, 0.36, 0.20, 0.12, 0
3	0.06, 0.31, 0.10, 0.51	0.30, 0.36, 0.19, 0.12, 0
4	0.05, 0.32, 0.08, 0.53	0.30, 0.36, 0.20, 0.12, 0
5	0.05, 0.35, 0.04, 0.54	0.29, 0.36, 0.21, 0.12, 0

Tab 3.9. Exemplo 2 - jogos de soma zero - modelo de Campos

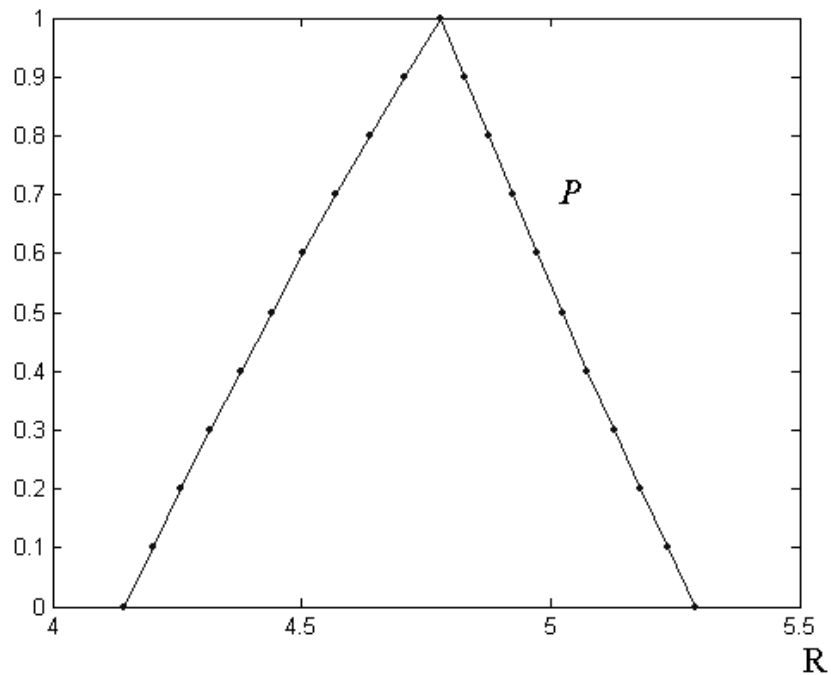


Fig 3.13 Valor do jogo no exemplo 2 pelo modelo de Campos: ordenação por 1, 2 e 4

No modelo de Maeda, como a matriz de pagamentos é bem maior que no primeiro exemplo e as nebulosidades são diferentes para cada elemento da matriz, os resultados são diferentes para diferentes valores dos parâmetros λ e μ . O valor de x^* ficou em torno de (0.7, 0.29, 0.14, 0.48) e o valor de y^* em torno de (0.28, 0.35, 0.22, 0.12, 0), enquanto que o valor do jogo P ficou em torno de 4.8, como ilustra a Figura 3.16

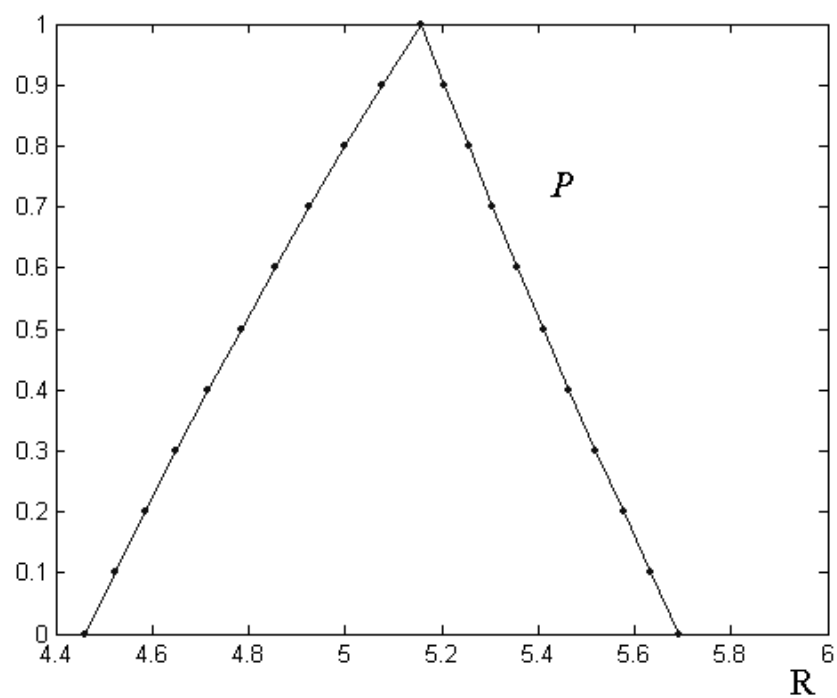


Fig 3.14 Valor do jogo no exemplo 2 pelo modelo de Campos: ordenação por 3

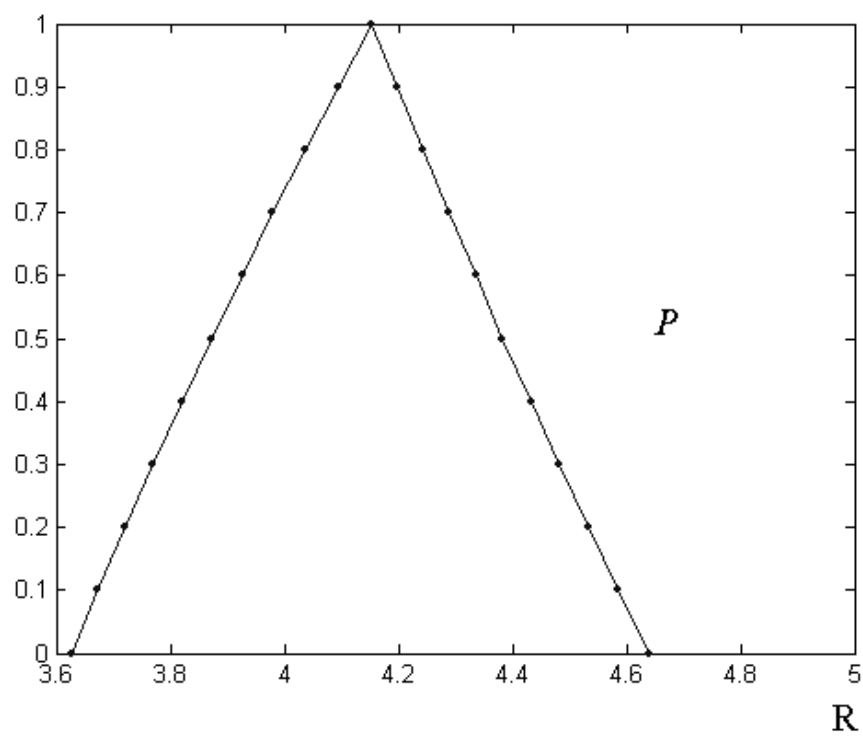


Fig 3.15 Valor do jogo no exemplo 2 pelo modelo de Campos: ordenação por 5

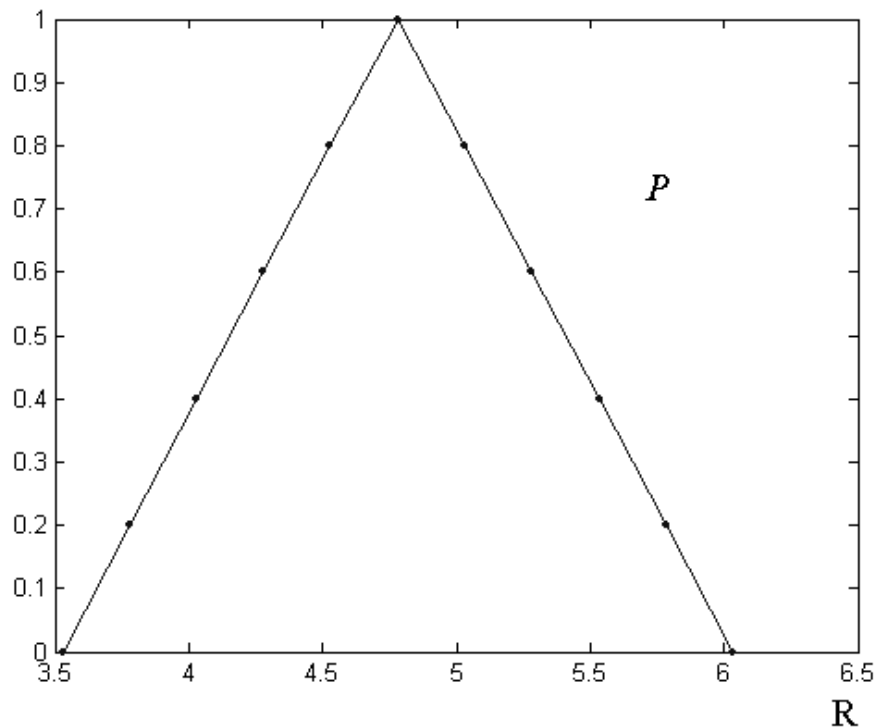


Fig 3.16 Valor do jogo para o exemplo 2 no modelo de Maeda

No modelo de Lee-Kwang e Lee os resultados obtidos são semelhantes, conforme Tabela 3.10:

x^*	y^*	P
0.111, 0.293, 0.143, 0.451	0.179, 0.326, 0.140, 0.354, 0	4.8625

Tab 3.10. Exemplo 2 - jogos de soma zero - modelo de Lee-Kwang e Lee

Para o modelo de α -corte, os resultados obtidos constam na Tabela 3.11:

α	x^*	y^*	P
0.1	0.04, 0.38, 0.01, 0.55	0.31, 0.38, 0.17, 0.12, 0	4.7810
0.2	0.04, 0.37, 0.02, 0.55	0.31, 0.37, 0.18, 0.12, 0	4.7843
0.3	0.04, 0.37, 0.03, 0.54	0.31, 0.37, 0.18, 0.12, 0	4.7841
0.7	0.05, 0.34, 0.06, 0.54	0.30, 0.36, 0.19, 0.14, 0	4.7807
1	0.05, 0.32, 0.08, 0.53	0.30, 0.36, 0.20, 0.12, 0	4.7804

Tab 3.11: Exemplo 1, modelo de α -corte

Nesse exemplo todos os algoritmos também convergiram comprovando a adequação

dos métodos para encontrar a solução de equilíbrio para os jogos nebulosos de soma zero.

Nos quatro modelos o equilíbrio do jogo encontrado foi semelhante, ou seja, os três modelos alcançaram o ótimo do problema. Porém, os modelos de Campos e de Maeda apresentaram desempenho computacional superior ao modelo de Lee-Kwang e Lee. O tempo de processamento nos modelos de Campos e de Maeda foi de quatro segundos, para resolver os problemas variando o valor de α de 0 a 1 e encontrar o número nebuloso correspondente ao valor do jogo P . Já o modelo de Lee-Kwang e Lee levou o dobro do tempo para encontrar os mesmos resultados. O modelo de α -corte levou cinco segundos para encontrar a solução. Apesar da eficiência do modelo de ordenação de Lee-Kwang e Lee ele se mostrou desvantajoso por se tratar de um modelo de difícil implementação e de desempenho computacional inferior aos demais modelos estudados.

Comparação dos resultados

Para analisar os métodos considerados nesse capítulo faz-se necessário um estudo mais detalhado do valor do jogo obtido em cada situação. Nos modelos de Maeda e de Campos o valor final do jogo é um número nebuloso. Existem diversas abordagens de se analisar um resultado na forma de números nebulosos, entre elas a ordenação de números nebulosos, a especificidade e a defuzzificação.

Considere os valores de P do primeiro exemplo estudado, obtidos por cada modelo, Tabela 3.12:

Método	P
Maeda	(151,161,171)
Campos (1,2 e 4)	(140,160,178)
Campos (3)	(140,162,179)
Campos (5)	(138,158,178)
Lee-Kwang e Lee	160

Tab 3.12. Resultado do jogo pelos diferentes métodos de resolução

Pelo método de ordenação de Yager, os valores de P são ordenados conforme

mostra a terceira coluna da Tabela 3.13.

Modelo	P	Método de Ordenação				
		Yager	Princípio da Extensão	α -Corte ($\alpha=0.5, 0.2$)	α -Corte ($\alpha=0.8$)	α -Corte ($\alpha=0.9$)
Maeda	(151,161,171)	1°.	2°.	4°.	2° ou 3°.	2°.
Campos (1,2 e 4)	(139,160,178)	3°.	3°.	2°.	2° ou 3°.	3°.
Campos (3)	(141,162,179)	2°.	1°.	1°.	1°.	1°.
Campos (5)	(137,158,177)	4°.	4°.	3°.	4°.	4°.

Tab 3.13. Ordenação dos resultados do jogo

Não há um padrão conclusivo na Tabela 3.13. Contudo, percebe-se em geral o resultado fornecido pelo modelo de Campos com o quinto método de ordenação apresentou um resultado inferior que os demais métodos. Em geral, para esse exemplo, o modelo de Maeda e o modelo de Campos com o terceiro método de ordenação foram os que obtiveram os melhores resultados.

Uma outra maneira de comparar os resultados é defuzzificar os valores obtidos de P utilizando, por exemplo, o método do centro de gravidade ou CoG. O valor defuzzificado é o valor da função:

$$F(x) = \frac{\sum_{i=1}^n A_i(x) a_i}{\sum_{i=1}^n A_i(x)} \quad (3.27)$$

Aplicando (3.27) ao valor do jogo fornecido por cada modelo obtemos os valores da Tabela 3.14. Observe que o modelo de Lee-Kwang e Lee já fornece um número real para P .

Ordenando os números reais obtidos com o método de CoG, em ordem crescente, temos a classificação indicada na coluna 4 da Tabela 3.14. A defuzzificação indica o mesmo resultado da ordenação de Yager. Como para esse exemplo os valores obtidos para o jogo são muito próximos, é difícil dizer qual o método de solução de jogos nebulosos de soma zero é o mais eficaz. Na verdade, a eficiência do método varia de acordo com o tipo de problema, as informações disponíveis sobre o problema e o grau de precisão que se deseja obter. De uma maneira geral, para os exemplos estudados nesse capítulo os três

modelos apresentaram resultados próximos, convergindo em todas as situações para soluções de equilíbrio semelhantes.

Modelo	P	CoG	Classificação
Maeda	(151,161,171)	161	1º.
Campos (1,2 e 4)	(139,160,178)	159	4º.
Campos (3)	(141,162,179)	160,66	2º.
Campos (5)	(137,158,177)	157,33	5º.
Lee-Kwang e Lee	160	160	3º.

Tab 3.14. Defuzzificação dos diferentes resultados do jogo

3.2.6 Comentários sobre os modelos

Todos os modelos para jogos nebulosos de soma zero com dois jogadores estudados neste capítulo transformam-se em modelos de programação linear e, conseqüentemente, métodos de solução baseados nos algoritmos de programação linear.

O modelo de Campos transforma a matriz nebulosa em uma matriz clássica utilizando métodos de ordenação dos números nebulosos. Campos propõe cinco métodos de ordenação diferentes. Este modelo explora a natureza dos números nebulosos triangulares presentes na matriz de pagamentos, no vetor b , e na tolerância das restrições. Quando α é igual a 1 (3.3), a tolerância das restrições é a menor possível, e quando α é igual a zero a tolerância é a maior possível. Pode-se executar o algoritmo várias vezes para diferentes valores de α e assim obter o valor do jogo, que é um número nebuloso.

Já para o modelo de Maeda as desigualdades são tratadas através da parametrização da matriz de pagamentos. Como vimos, $A(\lambda) = M + (I-2\lambda)H$ e $A(\mu) = M + (I-2\mu)H$ são, no modelo de Maeda, as matrizes de pagamentos para os jogadores I e II respectivamente. Portanto, nesse modelo, o jogo nebuloso é transformado em jogos bi-matrizes parametrizados. O valor final do jogo também é um número nebuloso e é obtido variando-se os valores dos parâmetros λ e μ .

Quando $\lambda = \mu$ temos um jogo com uma única matriz de pagamentos. Com isso, os números nebulosos tornam-se simétricos e o modelo é semelhante ao de Campos. Devido a

esse fato, Maeda afirma que seu modelo é mais geral que o modelo de Campos, pois permite a inserção de números nebulosos diferentes para cada jogador, através dos parâmetros λ e μ .

Por outro lado, Maeda investiga as propriedades do valor do jogo utilizando medidas de possibilidade e necessidade, o que os modelos restantes não fazem.

O terceiro modelo estudado utiliza um método de ordenação, incluindo o conceito de ponto de vista do jogador. Esse método foi aplicado na teoria dos jogos onde os números nebulosos da matriz de pagamentos foram ordenados diferentemente para cada jogador, levando-se em conta os objetivos e o ponto de vista de cada um dos jogadores.

Já o modelo de α -cortes representa a matriz de pagamentos nebulosa do jogo através de matrizes α -corte e também reformula o jogo em problemas de programação linear.

3.2.7 Aplicação em microeconomia

Considere o problema de um investidor, que possui três opções de investimentos, poupança, ações e fundos de investimentos com 100 milhões disponíveis para aplicar. O problema consiste em decidir como distribuir seu capital entre as opções de investimento, dados os possíveis retornos financeiros de cada investimento, Tabela 3.15.

		Situação de crescimento do mercado			
		Alto	Médio	Nulo	Crescimento negativo
Investimentos	Ações	R\$ 12 mil	R\$ 8 mil	R\$ 7 mil	R\$ 3 mil
	Fundos	R\$ 15 mil	R\$ 9 mil	R\$ 5 mil	R\$ -2 mil
	Poupança	R\$ 7 mil	R\$ 7 mil	R\$ 7 mil	R\$ 7 mil

Tab 3.15: Opções de investimento no mercado

O problema de investimento acima pode ser formulado como um jogo de soma zero, onde um jogador é o decisor e outro, o mercado. Ou seja, o decisor joga contra as condições de mercado. As situações possíveis são as de um alto crescimento, de um crescimento moderado, de um crescimento nulo, ou crescimento negativo, ou seja, queda de mercado.

O que nos interessa nesse jogo portanto é o resultado para o investidor, uma vez que queremos saber qual a melhor decisão de investimento que ele pode fazer, sem ter conhecimento sobre a situação do mercado nos próximos anos.

Esse problema, apesar de simplificado, serve para exemplificar como formular um jogo de soma zero a partir de uma situação real. Em decisões como essa outros fatores podem ser levados em conta, como conhecimento prévio das tendências de mercado para os próximos anos.

Considerando a Tabela 3.15, podemos formular um jogo nebuloso de soma zero com dois jogadores. Podemos considerar a matriz nebulosa de pagamentos do jogo cujo núcleo é A (3.28).

$$A = \begin{bmatrix} 12 & 8 & 7 & 3 \\ 15 & 9 & 5 & -2 \\ 7 & 7 & 7 & 7 \end{bmatrix} \quad (3.28)$$

e a matriz dispersão H (3.29).

$$H = \begin{bmatrix} 2 & 1 & 1 & 0.5 \\ 2 & 1.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.29)$$

Neste exemplo, apenas a opção de investimento na poupança foi mantida com valores precisos, pois é possível saber com exatidão qual a porcentagem de rendimento que uma poupança oferece.

O resultado clássico desse jogo, sem a nebulosidade na matriz de pagamentos, é $x^* = (0,0,1)$ e o rendimento esperado pelo decisor será de $P = 7$ milhões. Ou seja, o investidor deve aplicar todo o dinheiro na poupança, não se aventurando em investimentos de risco para um cenário como o descrito acima. O resultado do modelo sugere uma atitude conservadora.

Porém com a matriz de pagamentos nebulosa o jogo apresenta um resultado

diferente, $x^* = (0.2278, 0, 0.7713)$. Nesse caso a solução sugere que o investidor coloque a maior parte do dinheiro (77%) na poupança, que é mais segura, e arriscar investir o restante em ações. Essa estratégia evita que o investidor perca todo o dinheiro, caso faça um investimento de risco e o mercado não cresça e enfatize a primeira opção, pois no caso de um cenário negativo ainda assim ele teria um rendimento de 3% e não uma perda de 2%, como no caso dos fundos.

Alterando o núcleo da matriz de pagamentos (3.28) para (3.30).

$$A = \begin{bmatrix} 12 & 8 & 7 & 3 \\ 15 & 9 & 5 & 3 \\ 7 & 7 & 7 & 7 \end{bmatrix} \quad (3.30)$$

e mantendo a matriz dispersão inalterada (3.29), o resultado é $x^* = (0.0136, 0.1271, 0.8593)$. Nessas condições tornou-se interessante o investimento de pelo menos uma parte do dinheiro na segunda opção, pois agora ela não oferece mais uma perda caso o cenário seja negativo.

3.3 Jogos nebulosos cooperativos com k jogadores

Jogos nebulosos cooperativos são uma extensão dos jogos cooperativos clássicos. Essa seção discute a aplicação da teoria de jogos nebulosos cooperativos num problema de estratégia sugerido por Nishizaki e Sakawa (1998). Neste exemplo diferentes produtores fabricam seus produtos e podem formar coalizão para vender matéria prima entre si. O objetivo é maximizar os rendimentos dos produtores, sujeita às restrições dos recursos disponíveis. O problema é traduzido em um problema de programação linear.

A seguir apresenta-se o modelo de Nishizaki e Sakawa (1998) para a solução de jogos nebulosos cooperativos utilizando o conceito de α -cortes.

3.3.1 Definições

Seja k um inteiro positivo, $K = \{1, 2, \dots, k\}$, e seja O uma família de subconjuntos não vazios de K . Um elemento $i \in K$ é chamado de jogador e um elemento $S \in O$, uma coalizão.

Para um conjunto de números nebulosos $V = \{v_S \mid S \in O\}$, suponha que um pagamento $p \in \mathbb{R}$ e que p é menor que v_S , $p < v_S$, pode ser compartilhado por membros da coalizão S , e que $v \geq \sum_{i \in S} u_i$, onde $\mathbb{R}$ é o conjunto dos números reais e u_i , $i \in S$ é um pagamento para o jogador i . Então, um jogo nebuloso cooperativo é representado por (K, V) . Considere que v_S é um número nebuloso triangular, $v_S = \{v_S^L, v_S^M, v_S^R\}$. Nesse caso, considere $p \leq v_S^L$.

Seja $V_S(x)$ a função de pertinência de um número nebuloso V_S que representa o valor que os membros da coalizão S ganham sem a participação dos membros que não pertencem a S . Conforme Capítulo 1, o α -corte $V_{S\alpha}$ de V_S é:

$$V_{S\alpha} = \{x \in \mathbb{R} \mid V_S(x) \geq \alpha\}$$

e $V_{S\alpha}$ pode ser representado pelo intervalo $[V_{S\alpha}^L, V_{S\alpha}^R]$.

Definição 3.3: (Nishizaki e Sakawa, 1998) Um jogo nebuloso (K, V) é dito ser α -superaditivo se, para um dado $\alpha \in [0, 1]$ e para qualquer par de S e T , no qual $S \cap T = \emptyset$,

$$V_{\alpha}^R(S \cup T) \geq V_{\alpha}^R(S) + V_{\alpha}^R(T)$$

$$V_{\alpha}^L(S \cup T) \geq V_{\alpha}^L(S) + V_{\alpha}^L(T)$$

Conforme comentado no Capítulo 2 nenhum jogador racional aceitará um ganho menor do que o mínimo que ele ganharia por ele mesmo.

Definição 3.4: (Nishizaki e Sakawa, 1998) Para $\alpha \in [0,1]$, um α -vetor de entrada (α -inputation) é um vetor de pagamento u que satisfaz:

$$V_{\alpha}^L(K) \leq u(K) \leq V_{\alpha}^R(K)$$

$$u_i \geq V_{\alpha}^L(\{i\})$$

$$\text{onde } u(K) = \sum_{i \in K} u_i$$

Definição 3.5: (Nishizaki e Sakawa, 1998) Para $\alpha \in [0,1]$, seja u e w dois α -vetores de entrada (α -inputations) e seja $S \subseteq K$ uma coalizão. Dizemos que u α -domina w , u dom $_S w$, na coalizão S se:

$$u_i \geq w_i, \forall i \in S$$

$$u(S) \leq V_{\alpha}^L(S)$$

$$\text{onde } u(S) = \sum_{i \in S} u_i$$

Definição 3.6: (Nishizaki e Sakawa, 1998) Para $\alpha \in [0,1]$, o conjunto de todos os α -vetores de entrada (α -inputations) não dominados para um jogo (K, V) é chamado de α -núcleo do jogo α -C(K, V).

Teorema 3.2: Para um dado $\alpha \in [0,1]$, assuma que um jogo (K, V) satisfaz

$$V_{\alpha}^R(K) - V_{\alpha}^L(K) \geq V_{\alpha}^R(S) - V_{\alpha}^L(S), \forall S \in O$$

então o α -núcleo do jogo é o conjunto de todos os vetores de pagamentos que satisfazem as seguintes condições (3.31).

$$u(S) \geq V_{\alpha}^L(S) \tag{3.31}$$

$$V_{\alpha}^L(K) \leq u(S) \leq V_{\alpha}^R(K)$$

Portanto, para um dado valor de $\alpha \in [0,1]$, se o problema de programação linear (3.32).

$$\begin{aligned} \min z_{\alpha} &= \mu(K) = \mu_1 + \mu_2 + \dots + \mu_p \\ \text{sa : } \mu(S) &\geq V_{\alpha}^L(S), \\ \mu_1, \mu_2, \dots, \mu_p &\geq 0 \end{aligned} \quad (3.32)$$

cujas restrições são as condições (3.31), tiver um mínimo z_{α}^* em que $\sum_{S \in O} V_{\alpha}^R(S) \leq V^R \alpha(N)$, o α -núcleo do jogo nebuloso cooperativo α -superaditivo (K, V) não é vazio. Podemos formular um problema dual correspondente:

$$\begin{aligned} \max q_{\alpha} &= \sum_{S \subset N} V_{\alpha}^L(S) y_S \\ \text{sa : } \sum_{S \subset N_i} y_S &= 1, \forall i \in N \\ y_S &\geq 0 \end{aligned}$$

3.3.2 Método de solução

Num problema (linear) de produção com parâmetros nebuloso, cada decisor k tem um vetor de recursos $\tilde{b}^i = (\tilde{b}_1^i, \dots, \tilde{b}_m^i)$, $i=1, \dots, k$ e p tipos de produtos. Uma coalizão S terá um total de $\tilde{b}_l(S) = \sum_{i \in S} \tilde{b}_l^i$ unidade do l -ésimo recurso. Uma unidade do j -ésimo produto, $j=1, \dots, p$, requer $\tilde{a}_{lj}$ unidades do l -ésimo recurso e o decisor pode obter $\tilde{c}_j$ vendendo uma unidade do produto j . Considere $\tilde{a}$, $\tilde{b}$, $\tilde{c}$ números nebulosos triangulares. Os decisores desejam maximizar seus ganhos, vendendo os produtos sem considerar limitações de demanda e sujeitos às restrições de recursos.

Sendo assim, para uma coalizão S , o jogo cooperativo pode ser formulado como em (3.33).

$$\begin{aligned}
 \max \quad & \sum_{i=1}^p \tilde{c}_i x_i \\
 \text{sa :} \quad & \sum_{i=1}^p \tilde{a}_{1i} x_i \leq \tilde{b}_{S1} \\
 & \vdots \\
 & \sum_{i=1}^p \tilde{a}_{mi} x_i \leq \tilde{b}_{Sm} \\
 & \sum_{i=1}^p x_i \geq 0
 \end{aligned} \tag{3.33}$$

Para encontrar os valores das coalizões para o problema acima, Nishizaki e Sakawa (1998) utilizam um método baseado em α -cortes. Dessa maneira, os números nebulosos do problema (3.33) são definidos em função dos respectivos α -cortes como:

$$\begin{aligned}
 (\tilde{c}, \tilde{b}_S, \tilde{A})_\alpha = \{ & (c, b_S, A) \mid A(c_j) \geq \alpha, j = 1, \dots, p \\
 & A(b_{Sl}) \geq \alpha, l = 1, \dots, m \\
 & A(a_{lj}) \geq \alpha, l = 1, \dots, m, j = 1, \dots, p \}.
 \end{aligned}$$

que podem ser representados pelos conjuntos $[c_\alpha^L, c_\alpha^L]$, $[b_{S\alpha}^L, b_{S\alpha}^L]$ e $[a_\alpha^L, a_\alpha^L]$, onde $\tilde{A}$ é a matriz cujos elementos são números nebulosos triangulares, $\tilde{A} = [\tilde{a}_{ij}]$.

Assim podemos formular os problemas de programação linear a seguir:

$$\begin{aligned}
 \max \quad & \sum_{i=1}^p c_i^L x_i \\
 \text{sa :} \quad & \sum_{i=1}^p a_{1i}^R x_i \leq b_{S1}^L \\
 & \vdots \\
 & \sum_{i=1}^p a_{mi}^R x_i \leq b_{Sm}^L \\
 & \sum_{i=1}^p x_i \geq 0
 \end{aligned} \tag{3.34}$$

$$\begin{aligned}
 \max \quad & \sum_{i=1}^p c_i^R x_i \\
 \text{sa : } \quad & \sum_{i=1}^p a_{li}^L x_i \leq b_{S1}^R \\
 & \vdots \\
 & \sum_{i=1}^p a_{mi}^L x_i \leq b_{Sm}^R \\
 & \sum_{i=1}^p x_i \geq 0
 \end{aligned} \tag{3.35}$$

O valor da coalizão S para o jogo é V_S que pode ser construído através de $[V_{S\alpha}^L, V_{S\alpha}^R]$ variando-se o valor de $\alpha \in [0, 1]$, onde $V_{S\alpha}^L = \sum_{j=1}^n c_j^L x_{Sj}^{\alpha L}$ e $V_{S\alpha}^R = \sum_{j=1}^n c_j^R x_{Sj}^{\alpha R}$, com $x_{Sj}^{\alpha L}$ e $x_{Sj}^{\alpha R}$ que são os vetores da solução ótima dos problemas (3.34) e (3.35). A solução de equilíbrio de (3.34) e (3.35) são, respectivamente, z_α^L e z_α^R .

Para determinar o vetor de pagamentos no α -núcleo, Nishizaki e Sakawa (1998) utilizam as idéias de dualidade e análise de sensibilidade. Considere o problema a seguir:

$$\begin{aligned}
 \max \quad & \sum_{i=1}^p \tilde{c}_i x_i \\
 \text{sa : } \quad & \sum_{i=1}^p \tilde{a}_{ki} x_i \leq \tilde{b}_{Sk} + 1 \\
 & \sum_{i=1}^p \tilde{a}_{ki} x_i \leq \tilde{b}_{Sk} \\
 & k \neq \hat{k}, k = 1, \dots, m \\
 & \sum_{i=1}^p x_i \geq 0
 \end{aligned} \tag{3.36}$$

Utilizando a idéia de α -cortes vista acima, o problema (3.36) pode ser reescrito como:

$$\begin{aligned}
 \max \quad & \sum_{i=1}^p c_i^L x_i \\
 sa : \quad & \sum_{i=1}^p a_{ki}^R x_i \leq b_{Sk}^L + 1 \\
 & \sum_{i=1}^p a_{ki}^R x_i \leq b_{Sk}^L \\
 & k \neq \hat{k}, k = 1, \dots, m \\
 & \sum_{i=1}^p x_i \geq 0
 \end{aligned} \tag{3.37}$$

$$\begin{aligned}
 \max \quad & \sum_{i=1}^p c_i^R x_i \\
 sa : \quad & \sum_{i=1}^p a_{ki}^L x_i \leq b_{Sk}^R + 1 \\
 & \sum_{i=1}^p a_{ki}^L x_i \leq b_{Sk}^R \\
 & k \neq \hat{k}, k = 1, \dots, m \\
 & \sum_{i=1}^p x_i \geq 0
 \end{aligned} \tag{3.38}$$

Seja $z_{S\alpha}^{L\hat{l}}$ e $z_{S\alpha}^{R\hat{l}}$ soluções ótimas para os problemas (3.37) e (3.38). Tem-se, para todo $l=1, \dots, m$:

$$\begin{aligned}
 y_{S\alpha}^{Ll} &= z_{S\alpha}^{Ll} - z_{S\alpha}^L \\
 y_{S\alpha}^{Rl} &= z_{S\alpha}^{Rl} - z_{S\alpha}^R
 \end{aligned}$$

que são equivalentes a variáveis duais dos problemas (3.34) e (3.35).

Teorema 3.3: *Para todo $\alpha \in [0, 1]$ e para $i = 1, \dots, k$ um vetor de pagamentos que esteja no seguinte intervalo*

$$[u_{\alpha}^{iL}, u_{\alpha}^{iR}] = [\sum_{k=1}^m b_k^{iL} \times y_{N\alpha}^{iL}, \sum_{k=1}^m b_k^{iR} \times y_{N\alpha}^{iR}]$$

pertence ao α -núcleo.

3.3.3 Aplicação à problemas de produção

Considere cinco decisores $dm_1, \dots, dm_5$, cinco tipos de recursos $r_1, \dots, r_5$ e seis produtos $p_1, \dots, p_6$. O vetor de custo (c_i) é o quanto cada decisor vai lucrar vendendo o produto x_i . A matriz de pagamentos representa a quantidade do recurso r é necessário para produzir uma unidade do produto x . Por exemplo:

$$\begin{aligned} \max \quad & \tilde{8}x_1 + \tilde{5}x_2 + \tilde{3}x_3 + \tilde{4}x_4 + \tilde{7}x_5 + \tilde{6}x_6 \\ \text{sa:} \quad & \tilde{2}x_1 + \tilde{3}x_2 + \tilde{8}x_3 + \tilde{2}x_4 + \tilde{5}x_5 + \tilde{6}x_6 \leq b_1 \\ & \tilde{8}x_1 + \tilde{4}x_2 + \tilde{1}x_3 + \tilde{1}x_4 + \tilde{2}x_5 + \tilde{5}x_6 \leq b_2 \\ & \tilde{4}x_1 + \tilde{3}x_2 + \tilde{6}x_3 + \tilde{5}x_4 + \tilde{3}x_5 + \tilde{1}x_6 \leq b_3 \\ & \tilde{5}x_1 + \tilde{2}x_2 + \tilde{5}x_3 + \tilde{8}x_4 + \tilde{1}x_5 + \tilde{2}x_6 \leq b_4 \\ & \tilde{6}x_1 + \tilde{1}x_2 + \tilde{4}x_3 + \tilde{3}x_4 + \tilde{10}x_5 + \tilde{1}x_6 \leq b_5 \\ & x_i \geq 0, i = 1, \dots, 6 \end{aligned}$$

onde o valor de b vai depender da coalizão escolhida, conforme Tabela 3.16:

	dm_1	dm_2	dm_3	dm_4	dm_5
r_1	16	16	16	16	16
r_2	6	7	8	9	10
r_3	12	11	10	9	8
r_4	13	16	17	15	14
r_5	13	12	10	11	14

Tab 3.16: Recursos dos decisores

com os valores sendo números nebulosos.

Por exemplo, para a grande coalizão $S = K$, teremos o problema:

$$\begin{aligned}
 \max \quad & \tilde{8}x_1 + \tilde{5}x_2 + \tilde{3}x_3 + \tilde{4}x_4 + \tilde{7}x_5 + \tilde{6}x_6 \\
 \text{sa:} \quad & \tilde{2}x_1 + \tilde{3}x_2 + \tilde{8}x_3 + \tilde{2}x_4 + \tilde{5}x_5 + \tilde{6}x_6 \leq \tilde{80} \\
 & \tilde{8}x_1 + \tilde{4}x_2 + \tilde{1}x_3 + \tilde{1}x_4 + \tilde{2}x_5 + \tilde{5}x_6 \leq \tilde{40} \\
 & \tilde{4}x_1 + \tilde{3}x_2 + \tilde{6}x_3 + \tilde{5}x_4 + \tilde{3}x_5 + \tilde{1}x_6 \leq \tilde{50} \\
 & \tilde{5}x_1 + \tilde{2}x_2 + \tilde{5}x_3 + \tilde{8}x_4 + \tilde{1}x_5 + \tilde{2}x_6 \leq \tilde{75} \\
 & \tilde{6}x_1 + \tilde{1}x_2 + \tilde{4}x_3 + \tilde{3}x_4 + \tilde{10}x_5 + \tilde{1}x_6 \leq \tilde{60} \\
 & x_i \geq 0, i = 1, \dots, 6
 \end{aligned}$$

Variando o parâmetro α , podemos construir o valor de $V=\{S\}$, conforme Tabela 3.17.

	$\alpha=0.8$	$\alpha=0.6$	$\alpha=0.4$	$\alpha=0.2$
$V(\{1\})$	[14.6393, 15.2684]	[14.2069, 15.8502]	[13.5995, 16.7088]	[13.2244, 17.3476]
$V\{1, 2, 3, 4, 5\}$	[81.8685, 84.3988]	[79.1133, 87.4743]	[75.8241, 91.1414]	[73.4791, 94.0269]

Tab 3.17: Valor das coalizões

O valor das coalizões corresponde a soma do valor do jogo para todos os jogadores participantes da coalizão. Assim, a segunda linha da Tabela 3.17, o valor de $V\{1, 2, 3, 4, 5\}$, corresponde a soma do valor do jogo para o jogador um, dois, três, quatro e cinco, se eles optarem por cooperar entre si. A Figura 3.17 ilustra o valor nebuloso da coalizão $S\{1, 2, 3, 4, 5\}$.

O decisor um, por exemplo, tem a opção de jogar sozinho ou de formar coalizões entre um ou mais jogadores. Dessa forma, para esse exemplo, o jogador um tem as opções de coalizão como mostra a primeira coluna da Tabela 3.18. As demais colunas da Tabela 3.18 mostram o valor do jogo para as respectivas coalizões, utilizando os α -cortes 0.2, 0.6 e 0.8.

Ou seja, para a grande coalizão $S=\{1,2,3,4,5\}$, para $\alpha=0.8$, o valor do jogo para o decisor um é [15.3557, 15.8873], conforme a Tabela 3.18. Para o jogador 2, o valor do jogo é [15.7443, 16.1510], [15.5594, 16.0551] para o jogador 3, [16.6366, 17.2239] para o jogador 4 e [18.5726, 19.0815] para o jogador 5.

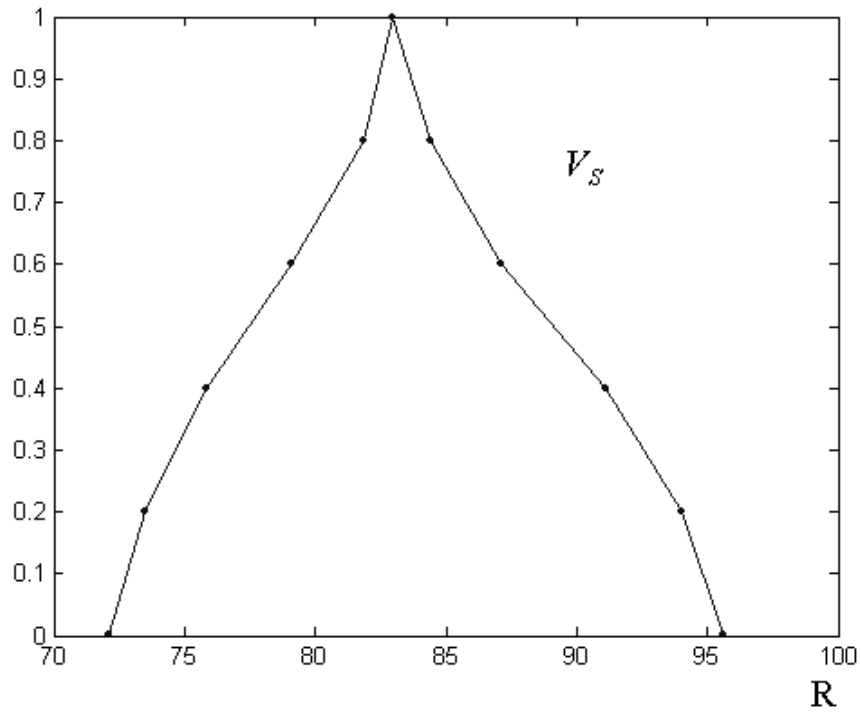


Fig 3.17: Valor da coalizão $S=\{1, 2, 3, 4,5\}$.

Coalizão	Valor do jogo		
	$\alpha=0.8$	$\alpha=0.6$	$\alpha=0.2$
$V(\{1\})$	[14.6393, 15.2684]	14.2069, 15.8502	[13.2244, 17.3476]
$V(\{1,2\})$	[14.6469, 15.3161]	14.2210, 15.9229	[13.2392, 17.4637]
$V(\{1,3\})$	[14.6469, 15.3161]	14.2210, 15.9229	[12.4512, 17.4637]
$V(\{1,4\})$	[14.6469, 15.3161]	14.2210, 15.9229	[12.6584, 17.4637]
$V(\{1,5\})$	[12.6292, 13.4473]	13.2649, 13.5662	[13.1235, 16.6399]
$V(\{1,2,3\})$	[14.6469, 15.3161]	13.8634, 15.9229	[11.1587, 17.4637]
$V(\{1,2,4\})$	[14.6469, 15.3161]	13.8840, 15.9229	[11.5700, 17.4637]
$V(\{1,2,5\})$	[12.9830, 14.1831]	13.8931, 14.2491	[13.7189, 17.4637]
$V(\{1,3,4\})$	[12.5871, 13.4726]	13.8034, 13.7380	[13.7189, 17.4637]
$V(\{1,3,5\})$	[15.3557, 15.8873]	14.8259, 16.4066	[13.7189, 16.9366]
$V(\{1,4,5\})$	[15.3557, 15.8873]	14.8259, 16.4066	[13.7189, 16.7152]
$V(\{1,2,3,4\})$	[12.9409, 14.0217]	14.4316, 13.9493	[13.7189, 17.4637]
$V(\{1,2,3,5\})$	[15.3557, 15.8873]	14.8259, 16.4066	[13.7189, 16.6060]
$V(\{1,2,4,5\})$	[15.3557, 15.8873]	14.8259, 16.4066	[13.7189, 16.8590]
$V(\{1,3,4,5\})$	[15.3557, 15.8873]	14.8259, 16.4066	[13.7189, 17.7828]
$V(\{1,2,3,4,5\})$	[15.3557, 15.8873]	14.8259, 16.4066	[13.7189, 17.7828]

Tab 3.18: Valor do jogo para o decisor 1

Para o caso do decisor um agir sozinho, sem formar coalizão com nenhum outro jogador, temos $S = \{1\}$ e a Figura 3.18 ilustra o valor da coalizão.

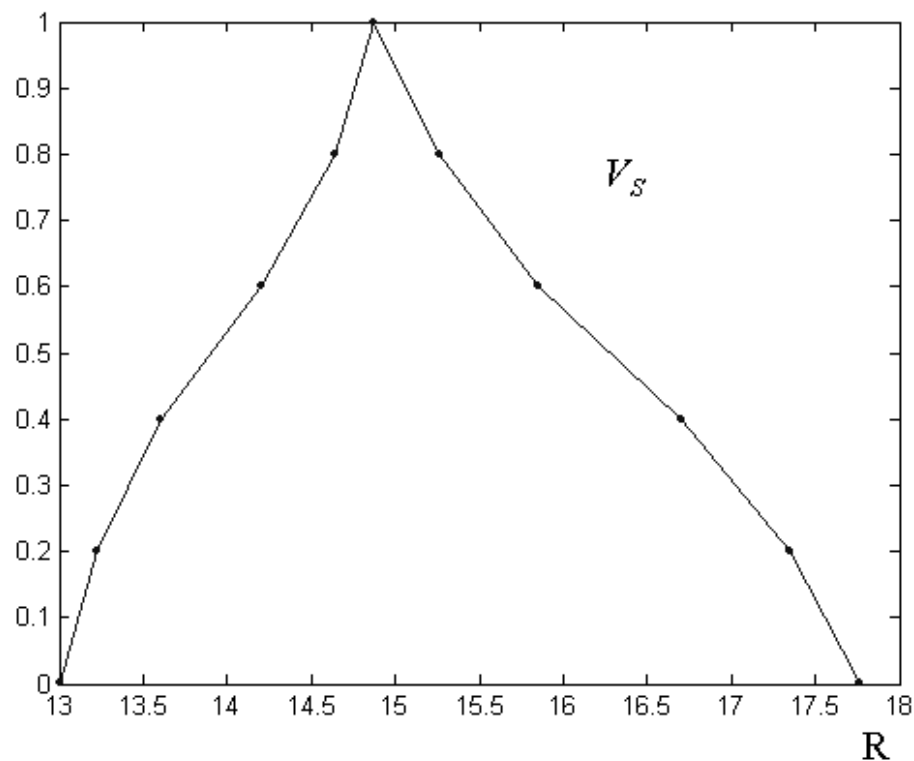


Fig 3.18: Valor da coalizão $S=\{1\}$

Nesse caso, o valor do jogo para o decisor um é o próprio valor da coalizão. O valor acima é obtido variando α de 0 a 1. Por exemplo, para $\alpha = 0.8$, o valor do jogo é igual a $[14.6393, 15.2684]$.

É baseado nos valores do jogo para as diferentes coalizões que o decisor escolhe qual a melhor maneira de realizar a produção.

3.3.4 Comentários sobre o método

Essa seção discutiu jogos nebulosos cooperativos, e uma metodologia baseada em α -cortes para solução. Foi mostrado como é possível formular o jogo nebuloso cooperativo como um problema de programação linear. Um exemplo ilustrativo de um problema de estratégia de produção foi considerado.

Jogos cooperativos possuem muitas outras aplicações. A teoria do "Altruísmo Recíproco", como é conhecida a cooperação nas ciências sociais, é um conceito amplamente estudado e atua como ferramenta para indivíduos que convivem com situações de conflito de interesses (Pindyck e Rubinfeld, 2002)..

A teoria dos jogos cooperativos pode ser aplicada ainda na política, em relações internacionais, estratégias de mercado, e outras aplicações que envolvem conflito de interesse entre entidades onde a cooperação é um mecanismo de solução.

3.4 Resumo

Esse capítulo considerou algoritmos para a solução de jogos nebulosos não cooperativos de soma zero e jogos nebulosos cooperativos. Os capítulos seguintes sugerem algoritmos para jogos nebulosos não cooperativos de soma não zero.

Foi mostrado como é possível formular um jogo nebuloso como uma família de problemas de programação linear. Exemplos ilustrativos foram apresentados para a análise da eficiência dos modelos.

Os quatro primeiros modelos consideram jogos nebulosos de soma zero não cooperativos. O modelo de Campos transforma a matriz nebulosa em uma matriz clássica utilizando cinco métodos de ordenação dos números nebulosos. Já o modelo de Maeda trata as desigualdades através da parametrização da matriz de pagamentos. O terceiro modelo estudado utiliza um método de ordenação, incluindo o conceito de ponto de vista do jogador, e o quarto modelo trabalha com as matrizes α -corte do pagamento nebuloso do jogo.

Este capítulo também tratou de jogos nebulosos cooperativos. Uma aplicação de um problema de estratégia de produção foi abordado, e um modelo sugerido por Nishizaki e Sakawa (1998) foi analisado.

O próximo capítulo apresenta um modelo para a solução de jogos nebulosos de soma não zero.

Capítulo 4

Algoritmo para resolução de jogos nebulosos de soma não zero

4.1 Introdução

Esse capítulo desenvolve um algoritmo para a resolução de jogos nebulosos não cooperativos de soma não zero para dois jogadores (Amaral e Gomide, 2007a). O algoritmo é baseado no modelo de α -cortes de Chen e Larbani (2005). Para tratar números nebulosos nas matrizes de pagamentos, o método decompõe um problema de programação bilinear associado em dois sub-problemas, e respectivos α -cortes. O algoritmo baseia-se em um esquema sugerido por Konno (1976) para resolver problemas de otimização bilineares.

Jogos nebulosos de soma não zero, conforme apresentado no Capítulo 2, podem ser formulados como problemas de programação bilinear, como segue (Nishizaki e Sakawa, 2001):

$$\begin{aligned}
 \max \quad & x\tilde{A}y + x\tilde{B}y - p - q \\
 \text{sa : } & \tilde{A}y \leq pe^m \\
 & \tilde{B}'x \leq qe^n \\
 & \sum_{i=1}^m x_i = 1 \\
 & \sum_{j=1}^n y_j = 1 \\
 & x_i \geq 0, i = 1, \dots, m. \\
 & y_j \geq 0, j = 1, \dots, n.
 \end{aligned} \tag{4.1}$$

onde e^m e e^n são vetores m - e n -dimensionais cujos elementos são 1, p e q são constantes, e $\tilde{A}$ e $\tilde{B}$ são matrizes de pagamento nebulosas dos jogadores I e II, respectivamente.

4.2 Algoritmo de Konno

O algoritmo de Konno resolve problemas de programação bilinear. A idéia é decompor o problema não linear em dois sub problemas lineares. Konno afirma que seu algoritmo converge para um ponto de Karush-Kuhn-Tucker em um número finito de iterações.

Definição 4.1 (Luenberger, 1984). *Seja x^* um ponto de mínimo local para o problema:*

$$\begin{aligned} \min f(x) \\ \text{sa : } h(x) = 0 \\ g(x) \leq 0 \end{aligned}$$

onde $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $g: \mathbb{R}^n \rightarrow \mathbb{R}^p$ são funções diferenciáveis.

Suponha que x^* é um ponto regular para as restrições. Então existe um vetor $\lambda \in \mathbb{R}^m$ e um vetor $\mu \in \mathbb{R}^p$ com $\mu \geq 0$ tal que:

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0$$

$$\mu^T \nabla g(x^*) = 0$$

Definição 4.2 (Rardin, 1998). *Qualquer $x \in \mathbb{R}^n$ para o qual existe os correspondentes $\lambda \in \mathbb{R}^m$ e $\mu \in \mathbb{R}^p$, $\mu \geq 0$ tal que as condições da Definição 4.1 sejam satisfeitas é chamado de um ponto de Karush-Kuhn-Tucker (KKT).*

As condições estabelecidas pela definição 4.1 são condições necessárias de otimalidade para problemas de otimização não linear diferenciáveis que satisfazem certas restrições de regularidade (qualificação de restrições). Estas condições são atribuídas à Karush-Kuhn-Tucker (Bazaraa et al, 1979). Para problemas de programação linear, estas condições são necessárias e suficientes (Bazaraa et al, 2005).

O algoritmo de Konno é descrito a seguir.

Considere o problema de programação bilinear para minimizar $\phi(x, y) = cx + dy + xHy$ sujeito a $x \in X$ e $y \in Y$, onde X e Y são poliedros limitados em $\mathbb{R}^n$ e $\mathbb{R}^m$, respectivamente. O algoritmo de Konno segue os passos seguintes:

Passo 1: Inicializar

Selecionar $x^1 \in \mathbb{R}^n$ e $y^1 \in \mathbb{R}^m$ iniciais. Seja $k=1$.

Passo 2:

Para x^k fixo, resolver o problema linear para minimizar $dy + x^k Hy$ sujeito a $y \in Y$.

Passo 3:

Seja $\hat{y}$ a solução ótima do problema. Atualizar y^{k+1} conforme (4.2).

$$y^{k+1} = \begin{cases} y^k & \text{se } \phi(x^k, \hat{y}) = \phi(x^k, y^k) \\ \hat{y} & \text{se } \phi(x^k, \hat{y}) < \phi(x^k, y^k) \end{cases} \quad (4.2)$$

Passo 4:

Par y^{k+1} fixo, resolver o problema linear para minimizar $cx + xHy^{k+1}$ sujeito a $x \in X$.

Passo 5:

Seja $\hat{x}$ a solução ótima do problema. Atualizar x^{k+1} conforme (4.3).

$$x^{k+1} = \begin{cases} x^k & \text{se } \phi(\hat{x}, y^{k+1}) = \phi(x^k, y^{k+1}) \\ \hat{x} & \text{se } \phi(\hat{x}, y^{k+1}) < \phi(x^k, y^{k+1}) \end{cases} \quad (4.3)$$

Passo 6:

Se $x^{k+1} = x^k$ e $y^{k+1} = y^k$: pare, com (x^k, y^k) como um ponto de Karush-Kuhn-Tucker. Senão, adicione 1 a k e volte ao passo 2.

O algoritmo acima inicializa as variáveis x e y e decompõe o problema de programação bilinear em dois problemas de programação linear. A idéia básica do algoritmo consiste na resolução iterativa desses dois problemas. O processamento termina quando a solução não muda de uma iteração para outra.

Segundo Konno (1976) esse algoritmo converge para um ponto de Karush-Kuhn-Tucker (KKT). A seguir sugere-se uma prova alternativa àquela de Konno.

Primeiramente, observa-se que, para x^1 fixo, o problema de programação linear

$$\begin{aligned} \min & dy + x^1 Hy \\ \text{sa} & : y \in Y \end{aligned}$$

possui solução ótima $\hat{y}$ pois Y é um poliedro limitado. Devido à (4.2), y^1 é um ótimo local de $\phi(x^1, y)$.

Similarmente, para y^1 fixo, o problema de programação linear

$$\begin{aligned} \min & cx + xHy^1 \\ \text{sa} & : x \in X \end{aligned}$$

possui solução ótima $\hat{x}$ pois X também é um poliedro limitado e devido a (4.3), x^2 é um ótimo local de $\phi(x, y^1)$.

Segundo, tem-se que $\phi(x^1, y^1) \leq \phi(x^1, y^2) \leq \phi(x^2, y^2) \leq \dots \leq \phi(x^{k+1}, y^{k+1})$ e, como X e Y possuem um número finito de vértices, o algoritmo termina em um mínimo local de $\phi(x, y)$ e, portanto, satisfaz as condições de KKT. Notar que a situação $\phi(x^{k+1}, y^{k+1}) < \phi(x^k, y^k)$ não pode ocorrer infinitamente. $\square$

4.3 Método de α -Cortes

O jogo formulado em (4.1) possui números nebulosos em seus coeficientes. Antes de se aplicar o algoritmo de Konno ao problema é necessário definir uma abordagem para tratar a matriz de pagamentos nebulosa. Nessa seção sugere-se um algoritmo baseado no método de α -cortes proposto por Chen e Larbani (2005).

Em seu artigo, Chen e Larbani (2005) propõem o uso de α -cortes para resolver um problema de decisão nebuloso formulando-o como um jogo com dois jogadores de soma zero. Os autores definem os jogadores como sendo o decisor e a natureza. O decisor visa maximizar seu ganho, o pagamento do jogo, enquanto que o segundo jogador visa minimizar seu gasto.

Nesse trabalho a idéia proposta por Chen e Larbani (2005) é estendida para jogos nebulosos de soma não zero. O valor nebuloso do jogo é obtido variando o valor de α entre 0 e 1.

Chen e Larbani consideram jogos cuja matriz de pagamentos é formada por números nebulosos triangulares simétricos. Esses números, como vimos na seção 1.5, podem ser representados por $A = (A^L, A^C, A^R)$, onde A^L , A^C e A^R são números reais e representam, respectivamente, o limite inferior, o núcleo e o limite superior do número nebuloso A .

Sabemos que o α -corte de A :

$$A_\alpha = \{x \mid A(x) \geq \alpha\}$$

pode ser representado por $[A_\alpha^L, A_\alpha^R]$, onde $A_\alpha^L = \inf(A)_\alpha$ e $A_\alpha^R = \sup(A)_\alpha$, onde $\sup(A)$ e $\inf(A)$ são, respectivamente, o supremo e o ínfimo de A_α .

De maneira similar, podemos representar uma matriz nebulosa $\tilde{A}_{m \times n}$ por meio de seu α -corte, $\tilde{A} = (A^L, A^C, A^R)$, onde $A^L = [a_{ij}^L]_{m \times n}$, $A^C = [a_{ij}^C]_{m \times n}$ e $A^R = [a_{ij}^R]_{m \times n}$ são matrizes de números reais correspondentes aos limites inferior, o núcleo e o limite superior dos números nebulosos da matriz $\tilde{A}$.

Nesse método é necessário escolher um valor para $\alpha \in [0,1]$ para obter o α -corte das matrizes de pagamento para os dois jogadores. Assim, $\tilde{A}$ é substituída por $[A_\alpha^L, A_\alpha^R]$ e $\tilde{B}$ é substituído por $[B_\alpha^L, B_\alpha^R]$. Dessa forma, o jogo (4.1) pode ser reformulado como:

$$\begin{aligned}
 \max \quad & x[A_\alpha^L, A_\alpha^R]y + x[B_\alpha^L, B_\alpha^R]y - p - q \\
 \text{sa} : \quad & [A_\alpha^L, A_\alpha^R]y \leq pe^m \\
 & [B_\alpha^L, B_\alpha^R]x \leq qe^n \\
 & \sum_{i=1}^m x_i = 1 \\
 & \sum_{j=1}^n y_j = 1 \\
 & x_i \geq 0, i = 1, \dots, m. \\
 & y_j \geq 0, j = 1, \dots, n.
 \end{aligned} \tag{4.4}$$

e os valores de x^* e y^* serão a solução ótima do problema para cada α .

Para resolver o problema (4.4), considerando a formulação como um jogo nebuloso com dois jogadores, supondo que ambos os jogadores desejam maximizar seus ganhos, o problema para encontrar a solução de equilíbrio para o jogador I consiste em resolver dois problemas de programação não linear:

$$\begin{aligned}
 \max \quad & x[A_\alpha^L + B_\alpha^L]y - p \\
 \text{sa} : \quad & A_\alpha^L x \leq pe^n \\
 & \sum_{i=1}^m x_i = 1 \\
 & \sum_{j=1}^n y_j = 1 \\
 & x_i \geq 0, i = 1, \dots, m.
 \end{aligned} \tag{4.5}$$

e

$$\begin{aligned}
 \max \quad & x[A_\alpha^R + B_\alpha^R]y - p \\
 \text{sa} : \quad & A_\alpha^R x \leq pe^n \\
 & \sum_{i=1}^m x_i = 1 \\
 & \sum_{j=1}^n y_j = 1 \\
 & x_i \geq 0, i = 1, \dots, m.
 \end{aligned} \tag{4.6}$$

De forma similar, para o jogador II:

$$\begin{aligned}
 \max \quad & x[A_{\alpha}^L + B_{\alpha}^L]y - q \\
 \text{sa : } & A_{\alpha}^L y \leq qe^m \\
 & \sum_{j=1}^n y_j = 1 \\
 & \sum_{i=1}^m x_i = 1 \\
 & y_j \geq 0, j = 1, \dots, n.
 \end{aligned} \tag{4.7}$$

e

$$\begin{aligned}
 \max \quad & x[A_{\alpha}^R + B_{\alpha}^R]y - q \\
 \text{sa : } & A_{\alpha}^R y \leq qe^m \\
 & \sum_{j=1}^n y_j = 1 \\
 & \sum_{i=1}^m x_i = 1 \\
 & y_j \geq 0, j = 1, \dots, n.
 \end{aligned} \tag{4.8}$$

A solução dos problemas (4.5) a (4.8) fornecem uma solução de equilíbrio do jogo para um determinado valor de α . Para encontrar o número nebuloso correspondente ao valor do jogo é necessário variar α entre 0 e 1.

No entanto, observee que o problema (4.4) é derivado do problema (4.1), cujas restrições são desigualdades nebulosas:

$$\begin{aligned}
 \tilde{A}y &\leq pe^m \\
 \tilde{B}'x &\leq qe^n
 \end{aligned} \tag{4.9}$$

Como dito no Capítulo 2, para esse trabalho, essas desigualdades são definidas utilizando:

$$\tilde{a} \geq \tilde{b} \text{ sse } (a_{\alpha}^L, a_{\alpha}^R) \geq (b_{\alpha}^L, b_{\alpha}^R) \forall \alpha \in [0,1] \tag{4.10}$$

Entretanto, p e q são escalares não nebulosos, ou seja, $p^L = p^R = q^L = q^R$.

Conjectura : Se p e q são escalares e $\tilde{A}$ e $\tilde{B}$ são matrizes nebulosas então:

$$\tilde{a}_i y \leq p_i \text{ sse } \begin{cases} (a_i)_\alpha^L y \leq (p_i)^L \\ (a_i)_\alpha^R y \leq (p_i)^R \end{cases}, \forall \alpha \in [0,1]$$

como $p_i^L = p_i^R$, temos:

$$\tilde{a}_i y \leq p_i \text{ sse } (a_i)_\alpha^R y \leq (p_i)^R, \forall \alpha \in [0,1]$$

O mesmo é feito para $\tilde{B}$ e q .

Sendo assim, o problema em (4.4) pode ser resolvido encontrando a solução ótima apenas para os problemas (4.6) e (4.8).

A partir dos itens apresentados o algoritmo pode ser construído seguindo os passos abaixo:

Passo 1:

Entrar com valor de α .

Passo 2

Calcular matriz de α -cortes para o jogador I.

Passo 3:

Resolver o jogo (4.6)

Passo 4:

Calcular matriz de α -cortes para o jogador II.

Passo 5:

Resolver o jogo (4.8)

Esses passos são repetidos até que o resultado não mais se altere e o algoritmo encontre a solução de equilíbrio do jogo.

4.4 Exemplos

Essa seção apresenta dois exemplos para ilustrar o comportamento do algoritmo de Konno combinado com o método de α -cortes de Chen e Larbani (2005).

Exemplo 1: este caso é adaptado de (Basar e Olsder, 1982), cuja matriz de pagamentos são as seguintes (4.11):

$$A = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \quad (4.11)$$

onde ambos os jogadores desejam maximizar seus ganhos.

Como nessa dissertação o foco é jogos nebulosos, as matrizes A e B são alteradas, substituindo números reais por números nebulosos. Assim, as matrizes de pagamento para o jogo nebuloso do exemplo são (4.12):

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix} & H_A &= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \\ B &= \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} & H_B &= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \end{aligned} \quad (4.12)$$

onde A e B são matrizes núcleo da matriz nebulosa e as matrizes H_A e H_B , as matrizes de dispersão para os jogadores I e II, respectivamente.

Assim sendo, as matrizes A_α^L , A_α^R e B_α^L , B_α^R são:

$$A_\alpha^L = A - H_A$$

$$A_\alpha^R = A + H_A$$

$$B_\alpha^L = B - H_B$$

$$B_\alpha^R = B + H_B$$

Repare que o algoritmo proposto neste capítulo resolve também jogos clássicos, ou seja, jogos onde os elementos das matrizes de pagamento não são números nebulosos. Para isso, o valor de α é igual a 1. A Tabela 4.1 ilustra os resultados obtidos para o caso clássico.

α	Iterações	x	y	p	q	P_I	P_{II}
1	2	(0.5, 0.5)	(0.5, 0.5)	0.5	1.5	0.5	1.5

Tab 4.1: Resultado do Exemplo 1 para o caso clássico.

Para o caso clássico, o algoritmo encontra a solução de equilíbrio em duas iterações. Os valores obtidos coincidem com o resultado clássico obtido por Basar e Olsder (1982).

O exemplo nebuloso foi testado com as matrizes (4.12). Os resultados constam na Tabela (4.2).

α	Iterações	x	y	p	q	P_I	P_{II}
0.5	2	(0.5, 0.5)	(0.5, 0.5)	0.75	1.75	0.75	1.75

Tab 4.2: Resultado do Exemplo 1, para $\alpha=0.5$, com ponto inicial factível.

Assim para o caso clássico, o algoritmo convergiu em duas iterações. O ponto inicial foi de $x = (0.5, 0.5)$ e $y = (0.5, 0.5)$, ambas soluções iniciais factíveis. Diante dos resultados obtidos, encontrou-se a necessidade de testar o mesmo exemplo, mas começando com uma solução infactível. Para $x = (1, 1)$ e $y = (0, -1)$ como solução inicial o resultado obtido foi (Tabela 4.3):

α	Iterações	x	y	p	q	P_I	P_{II}
0.5	3	(0.5, 0.5)	(0.5, 0.5)	0.75	1.75	0.75	1.75

Tab 4.3: Resultado do Exemplo 1, para $\alpha=0.5$, com ponto inicial infactível.

A Tabela 4.3 ilustra os resultados obtidos. Repare que o algoritmo convergiu para a mesma solução, porém com uma iteração a mais do que no caso em que a solução inicial era factível.

Diferentes valores de α foram testados. A Tabela 4.4 ilustra os resultados obtidos.

α	Iterações	x	y	p	q	P_I	P_{II}
0	2	(0.5000, 0.5000)	(0.5000, 0.5000)	1	2	1	2
0.1	2	(0.5000, 0.5000)	(0.5000, 0.5000)	0.9	1.9	0.9	1.9
0.2	2	(0.5000, 0.5000)	(0.5002, 0.4998)	0.89	1.90	0.90	1.89
0.3	2	(0.5000, 0.5000)	(0.4997, 0.5003)	0.84	1.84	0.84	1.84
0.4	2	(0.4998, 0.5002)	(0.5000, 0.5000)	0.79	1.79	0.79	1.79
0.5	2	(0.5000, 0.5000)	(0.5000, 0.5000)	0.75	1.75	0.75	1.75
0.6	2	(0.5000, 0.5000)	(0.5000, 0.5000)	0.69	1.69	0.69	1.69
0.7	2	(0.5000, 0.5000)	(0.5002, 0.4998)	0.64	1.65	0.65	1.64
0.8	2	(0.5000, 0.5000)	(0.4997, 0.5003)	0.59	1.59	0.59	1.59
0.9	2	(0.4998, 0.5002)	(0.5000, 0.5000)	0.54	1.54	0.54	1.54
1	2	(0.5000, 0.5000)	(0.5000, 0.5000)	0.5	1.5	0.5	1.5

Tab 4.4: Resultado do Exemplo 1, para diferentes valores de $\alpha=0.5$.

Para uma solução inicial infactível, com diferentes valores de α , o algoritmo se comportou de maneira semelhante, e os resultados obtidos foram os mesmos da Tabela 4.4, exceto pelo fato de que o algoritmo levou 3 iterações para encontrar a solução de equilíbrio.

O algoritmo convergiu para a solução de equilíbrio nos casos testados. A seguir, serão apresentadas aplicações de jogos nebulosos de soma não zero, e o algoritmo de α -cortes é utilizado para encontrar a solução de equilíbrio dos exemplos.

4.5 Aplicações

A seguir discutem-se aplicações dos jogos nebulosos de soma não zero.

4.5.1 Estratégia de investimento preemptivo

O Wal-Mart Stores, Inc. é uma cadeia de lojas de varejo dos Estados Unidos, que se destacou das demais pela sua expansão na década de 60, em um mercado altamente competitivo (Pindyck e Rubinfeld, 2002).

O sucesso da Wal-Mart encontra-se em sua estratégia de expansão. Na época era

comum pensar que lojas de varejo como a Wal-Mart só obteriam sucesso se fossem instaladas em cidades com população igual ou maior que cem mil habitantes. Sam Walton, proprietário da Wal-Mart decidiu contrariar essa idéia e abrir suas lojas em pequenas cidades dos Estados Unidos. Em 1970, já eram trinta lojas em pequenas cidades. Isso garantiu a Wal-Mart trinta “monopólios locais”, enquanto as lojas que abriam em cidades grandes tinham que competir com as outras lojas de varejo, resultando na redução dos preços dos produtos e conseqüentemente na queda da margem de lucro.

A estratégia diferenciada de Sam Walton chamou a atenção das demais redes de lojas varejistas, o que gerou um jogo preemptivo entre as concorrentes, onde o segredo era conseguir abrir a loja em uma cidade pequena antes que alguma outra loja o fizesse.

Essa disputa pelas cidades pequenas pode ser traduzida em um jogo com a seguinte matriz de pagamentos, Tabela 4.5:

		Empresa X	
		Entra na cidade	Não entra na cidade
Wal-Mart	Entra na cidade	-10/-10	20/0
	Não entra na cidade	0/20	0/0

Tab 4.5: Estratégia de investimento preemptivo

Como pode ser visto na matriz acima, se a Wal-Mart se estabelecer em uma cidade e a concorrente X não, o Wal-Mart lucra 20 e a empresa X lucra 0. Se ocorrer o oposto, a empresa X se estabelecer na cidade e a Wal-Mart não, a empresa X lucra 20 e a Wal-Mart 0. Mas se as duas empresas abrirem suas lojas na mesma cidade, as duas terão um prejuízo de 10.

Esse jogo possui dois equilíbrios de Nash. Isso acontece quando apenas uma das empresas se estabelece na cidade. O resultado dessas situações é representado no canto superior direito e no canto inferior esquerdo da matriz de pagamentos.

A estratégia ótima está portanto em agir preemptivamente, ou seja, a empresa abrir a loja antes que sua concorrente o faça. Esse jogo também pode ser formulado como um jogo nebuloso de soma não zero, com as matrizes de pagamentos de (4.13).

$$A = \begin{bmatrix} -10 & 20 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -10 & 0 \\ 20 & 0 \end{bmatrix} \quad (4.13)$$

$$H_A = \begin{bmatrix} 10 & 10 \\ 10 & 10 \end{bmatrix} \quad H_B = \begin{bmatrix} 10 & 10 \\ 10 & 10 \end{bmatrix}$$

Este jogo foi resolvido utilizando o algoritmo apresentado na seção 4.3, com $\alpha = 0.5$, e a solução de equilíbrio depende da inicialização das variáveis. Quando $x^0 = (1,0)$ e $y^0 = (1,0)$ o resultado obtido é $x^* = (1,0)$ e $y^* = (0,1)$ enquanto que quando $x^0 = (0,1)$ e $y^0 = (0,1)$ o resultado é $x^* = (0,1)$ e $y^* = (1,0)$.

O equilíbrio resultante dependerá de quem fizer o primeiro movimento. Isso é o que a Wal-Mart fez. Em meados dos anos 80 ele já possuía 1.009 lojas em operação e seu lucro anual era de \$450 milhões. Em 1999, o Wal-Mart tinha 2454 lojas espalhadas pelos Estados Unidos e outras 729 pelo mundo, faturando anualmente \$138 bilhões.

4.5.2 Aplicação no mercado de energia

Uma das aplicações de jogos de soma não zero é no mercado de energia. Na literatura essa aplicação é abordada pela teoria dos jogos através de um modelo de Cournot. (Singh, 1999)

No mercado de oligopólio existem vendedores que competem entre si para conseguir conquistar os compradores e um bom preço de seus produtos no mercado. No caso especial de dois vendedores esse mercado é conhecido como duopólio. Em um oligopólio, os compradores não podem influenciar no preço ou quantidade oferecidos. Os vendedores podem escolher o preço de seus produtos e a quantidade produzida. Esse exemplo se refere ao que é conhecido como duopólio de Cournot, como no caso de duas empresas com produtos idênticos que operam em um mercado de energia cuja demanda é conhecida (Basar e Olsder, 1982).

Esse exemplo consiste em cada empresa decidir quanto produzir, sem saber a qual a quantidade de produção que a outra empresa vai adotar. O exemplo, considerado por Singh (1999), analisa como contratos de diferenças (*CfD – contracts for differences*) influenciam o mercado de energia elétrica.

O termo *CfD* é bastante utilizado em economia. Um *CfD* é um contrato bilateral onde um produtor de energia recebe um preço fixo para cobrir qualquer diferença entre o preço fixado por acordo e o preço atual do mercado de energia quando ela é entregue. Ou seja, é um acordo feito entre o comprador e o vendedor onde eles concordam em pagar a diferença quando o preço está favorável para um e desfavorável para outro.

O contrato de diferença é uma operação arriscada e seu conceito envolve inúmeras outras variáveis, sendo seu entendimento bastante complexo. Os contratos por diferença são utilizados no mercado de energia e respondem por cerca de 70% dos contratos negociados na Inglaterra (Hoare, 1995).

Nesse trabalho estamos interessados em analisar o comportamento de duas empresas do mercado de energia através de uma abordagem da teoria dos jogos. Para ilustrar como os *CfDs* podem eliminar os incentivos para elevar os preços, será utilizado um modelo simples de Cournot, proposto por Singh (1999) com dois geradores A e B. Cada gerador tem um custo operacional igual a 10 por MW e suas saídas dependem do nível de produção, baixo ou alto, como mostra a Tabela 4.6:

Saídas em MW		Gerador B		
		Alto	Baixo	
Gerador A	Alto	75	75	Geração de A
		75	20	Geração de B
	Baixo	20	20	Geração de A
		75	20	Geração de B

Tab 4.6: Saídas das decisões de produção de A e B

O modelo considera que os geradores têm dois níveis de produção, alta (75 MW) e baixa (20 MW). O problema é decidir qual a estratégia ótima de operação dos geradores, isto é, a estratégia que maximize seus lucros. Os preços correspondentes a cada nível de produção são ilustrados pela Tabela 4.7:

Preço em MW/hora	Gerador B	
	Alto	Baixo

Gerador A	Alto	40	45
	Baixo	45	150

Tab 4.7: Preços correspondentes a cada nível de produção de A e B

Um nível baixo de produção pode ser interpretado como uma tentativa de diminuir a oferta, aumentando os preços. O lucro final de cada gerador é calculado subtraindo o custo operacional do preço correspondente a opção de produção. Assim, quando não há contrato de diferenças o lucro de cada gerador corresponde a Tabela 4.8:

Lucro (A/B)		Gerador B	
		Alto	Baixo
Gerador A	Alto	2250 / 2250	2625 / 700
	Baixo	700 / 2625	2800 / 2800

Tab 4.8: Lucro, sem contratos de diferença

Singh (1999) aponta para um equilíbrio de Nash, $x^* = (0, 1)$ e $y^* = (0, 1)$, onde os dois geradores escolhem o nível baixo de produção. Essa estratégia é de fato a mais atraente para os jogadores, uma vez que ambos querem maximizar o lucro e essa opção fornece o maior lucro. O valor do jogo é de 2800 para ambos os jogadores.

O que o autor não aponta, porém, é o fato de que esse jogo possui mais de um equilíbrio de Nash, com $x^* = (1, 0)$ e $y^* = (1, 0)$, com o valor do jogo de 2250 para ambos os jogadores. Observe que o jogo apresentado acima é semelhante ao dilema do prisioneiro. A estratégia segura para os dois jogadores é manter a produção alta e obter um lucro de 2250, pois manter a produção em nível baixo exige confiança de que o outro jogador fará a mesma coisa, pois do contrário os prejuízos serão muito maiores.

No entanto, na prática, em um mercado competitivo de energia elétrica um acordo bilateral entre as duas partes para manter a produção baixa e ambos lucrarem mais, na situação descrita acima e levando em conta apenas as variantes citadas, é sem dúvida mais vantajoso que ambos os geradores produzam em baixa escala. Na verdade, em situações reais de duopólio essa estratégia é bastante comum.

No caso dessa dissertação o jogo foi formulado como um jogo nebuloso de soma não zero com dois jogadores, onde os jogadores são os decisores relativos ao gerador A e

ao gerador B. Supondo que as matrizes de pagamento do jogo nebuloso são os lucros de cada gerador:

$$A = \begin{bmatrix} 2250 & 2625 \\ 700 & 2800 \end{bmatrix} \quad B = \begin{bmatrix} 2250 & 700 \\ 2625 & 2800 \end{bmatrix} \quad (4.14)$$

$$H_A = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix} \quad H_B = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix}$$

A solução de equilíbrio encontrada foi de $x^* = (0, 1)$ e $y^* = (0, 1)$ $P = 2800$ para ambos os jogadores, no caso de $\alpha = 1$. Vários valores de α foram testados e em todos os casos os valores de x e y obtidos foram iguais, alterando-se apenas o valor do jogo P :

α	Iterações	x	y	p	q	P_I	P_{II}
0.1	2	(0, 1)	(0, 1)	2890	2890	2890	2890
0.4	2	(0, 1)	(0, 1)	2860	2860	2860	2860
0.7	2	(0, 1)	(0, 1)	2829	2829	2829	2829

Tab 4.9: Resultados para o exemplo de mercado de energia

O algoritmo se mostrou eficiente em todos os exemplos testados. Dentre as principais vantagens destacam-se a fácil compreensão e implementação do algoritmo, e ainda sua rápida execução. Outra vantagem é a garantia que o algoritmo sempre encontra um ponto de Kuhn-Tucker. O algoritmo bilinear possui ainda a característica de generalizar a solução de problemas de jogos, podendo ser aplicado para encontrar a solução de equilíbrio para jogos nebulosos ou não.

4.6 Resumo

Nesse capítulo discutiu-se jogos nebulosos com dois jogadores de soma não zero, isto é, jogos nebulosos bimatriciais. Esses jogos podem ser formulados como problemas de programação bilinear e resolvidos, conforme proposta deste trabalho, através do algoritmo de Konno (1976). A nebulosidade que aparece nos coeficientes do problema foi tratada

através de um método de alfa cortes inspirado no método sugerido por Chen e Larbani (2005).

Sucintamente, o algoritmo implementado é uma combinação do algoritmo de Konno com o método de Chen e Larbani adaptado para um problema de programação bilinear. Nos exemplos implementados os resultados convergiram e uma estratégia de equilíbrio de cada jogador foi encontrada. O valor ótimo do jogo foi obtido variando o valor de α .

Além disso, esse capítulo também apresentou exemplos de aplicações de jogos nebulosos bimatriciais e sua resolução, entre eles, aplicações em microeconomia e no mercado de energia elétrica.

Capítulo 5

Abordagem co-evolutiva para solução de jogos nebulosos

5.1 Introdução

Esse capítulo propõe um algoritmo co-evolutivo para a solução de problemas de jogos nebulosos (Amaral e Gomide, 2007b). Algoritmos evolutivos derivam de meta-heurísticas de otimização populacional inspirados nos mecanismos de evolução biológica de Darwin, como reprodução, mutação, recombinação, adaptação ao meio, entre outros. Através desses mecanismos o algoritmo evolui uma população de indivíduos, cada um deles representando uma possível solução para o problema.

O uso da computação evolutiva na solução de problemas de otimização é extremamente interessante. Isso se dá principalmente pela possibilidade que os processos evolutivos oferecem de se resolver problemas descrevendo-os matematicamente, sem a necessidade de se programar explicitamente os passos do processo de resolução, que em outros métodos muitas vezes são específicos para cada problema.

Tanto a computação evolutiva quanto a teoria de conjuntos nebulosos são consideradas ramos da inteligência computacional. Atualmente existem inúmeros estudos (Buckleya, 1998, Delgado, 2002, Cordon e Gomide, 2004) que combinam esses dois paradigmas para a solução de problemas complexos.

Nesse capítulo serão estudados algoritmos evolutivos aplicados na solução de problemas em teoria de jogos nebulosos.

5.2 Algoritmos evolutivos

Os algoritmos evolutivos em sua forma geral são constituídos de vários elementos e procedimentos que podem ser adaptados para resolver problemas distintos. Os principais elementos utilizados são uma população de indivíduos, que representam soluções candidatas para o problema e uma função de avaliação, que determina a adequação de cada

indivíduo em relação aos demais, conhecida como função de *fitness*.

Darwin (1859) propôs algumas hipóteses para explicar o processo de evolução. Em uma determinada população geralmente os filhos tendem a ser em maior número que os pais. Porém o número de indivíduos de uma espécie permanece constante. Se essas duas afirmativas ocorrem, conclui-se que existe uma competição entre os indivíduos da espécie, em busca da sobrevivência. Outra observação de Darwin foi que em uma mesma espécie os indivíduos costumam apresentar diferenças, apesar das características semelhantes entre pais e filhos. Darwin concluiu que a seleção natural privilegia indivíduos cujas características melhor se adaptam ao ambiente, pois estas terão uma maior chance de sobreviver e de se reproduzir.

Atualmente, estas hipóteses não são suficientes para explicar os processos adaptativos em nichos ecológicos. É necessário considerar, por exemplo, que existe algum processo de variação continuada responsável pela introdução de novas informações junto à carga genética dos organismos. A seleção natural é um mecanismo para preservação de novas informações que correspondem a uma maior adaptação.

Portanto, a seleção natural tem uma natureza incerta e embora ela ocorra com os indivíduos de uma maneira particular, seu efeito resultante se manifestará na espécie como um todo. Ou seja, a espécie é o beneficiário final do processo evolutivo (Mayr, 1988).

Uma das etapas principais na resolução de um problema através de métodos evolutivos é a formulação do problema. O problema deve ser formulado adequadamente, representando as soluções candidatas, ou seja, os indivíduos da população, de maneira que eles possam ser manipulados pelos operadores do programa e mesmo assim preservar a factibilidade da solução. A representação da solução nos algoritmos evolutivos corresponde a uma descrição de cada indivíduo da população através de uma lista ordenada de atributos, descrita a partir de um alfabeto finito. Essa representação é conhecida como cromossomo.

Nos algoritmos genéticos (GA, *genetic algorithm*), a representação utilizada é a de um vetor binário, como mostra a Figura 5.1:



Fig 5.1: Representação de um cromossomo em um GA

Onde cada casa é chamada de loci, e os valores possíveis para cada casa são chamados de alelos. Assim, o cromossomo da Figura 5.1 possui seis loci e dois alelos, 0 e 1.

Para que seja introduzida diversificação na população existem vários operadores que podem ser aplicados durante a execução do programa. Os operadores mais comuns são *crossover* e *mutação*.

O *crossover* ou cruzamento consiste na troca de material genético entre dois cromossomos. A *mutação* é uma pequena modificação nos atributos do cromossomo que cria um novo indivíduo. A Figura 5.2 ilustra o cruzamento entre dois cromossomos e, como resultado, a criação de dois novos cromossomos.

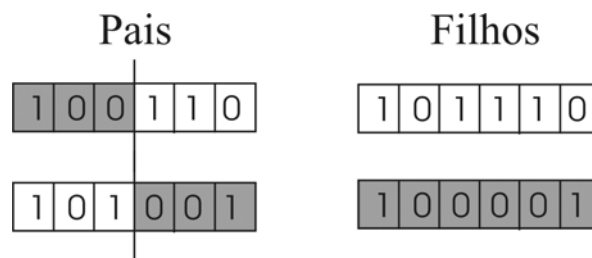


Fig 5.2: Cruzamento entre cromossomos binários

A Figura 5.3 ilustra a mutação em um indivíduo.

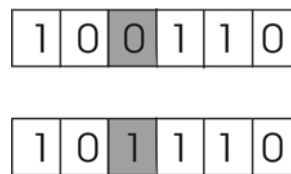


Fig 5.3: Mutação em um cromossomo binário

A seleção dos indivíduos mais aptos para formar a próxima geração é um dos passos mais críticos em um algoritmo evolutivo. Se não for bem formulado, o algoritmo de seleção tende a reduzir drasticamente a diversidade da população, fazendo o algoritmo convergir para um ótimo local.

Na sessão seguinte serão descritas as principais características da computação evolutiva relevante para este trabalho.

5.3 Co-evolução

Os algoritmos evolutivos são amplamente utilizados nos estudos atuais de otimização. Porém, eles podem ser extremamente difíceis de se aplicar em classes específicas de problemas, quando por exemplo não há uma medida clara de *fitness*, dificultando a avaliação dos indivíduos da população. Esse é o caso de problemas que envolvem estratégias para teoria de jogos.

Surge então a necessidade de modelos mais complexos. Os modelos co-evolutivos oferecem novas perspectivas para utilizar algoritmos evolutivos para determinadas classes de problemas (Olsson, 2001).

Um algoritmo co-evolutivo (*coevolutionary algorithm*) é um algoritmo evolutivo no qual o *fitness* de um indivíduo depende da relação entre ele e os demais indivíduos da população. Ou seja, na co-evolução os indivíduos não são avaliados em si, pois o que é importante é avaliar a interação entre indivíduos (Wiegand, 2003). Portanto, nos algoritmos co-evolutivos a solução evolui na forma de sub-componentes, interagindo de maneira co-adaptativa. (Delgado, 2002).

Os algoritmos co-evolutivos adotam populações distintas evoluindo simultaneamente utilizando medida de desempenho acopladas, ou seja, as respectivas funções de *fitness* de cada população relacionam-se entre si

A co-evolução e a teoria de jogos estão fortemente relacionadas. Co-evolução e competição são temas de estudo de inúmeros trabalhos (Ficici. e Pollack, 2000, Rosin e Belew, 1995, Pollack.e Blair, 1998) onde se analisa a cooperação e o aprendizado de estratégias em jogos iterativos.

Nessa seção exemplos de jogos considerados nos capítulos anteriores são resolvidos com um algoritmo co-evolutivo. Os problemas serão formulados não mais como problemas de programação linear, mas como populações de soluções que competem entre si.

5.4 Descrição do algoritmo

A computação evolutiva abrange diversas técnicas como os algoritmos genéticos

(Holland, 1975), as estratégias evolutivas (Rechenberg, 1973), a programação evolutiva (Fogel, 1966) e a programação genética (Koza, 1992).

O que distingue os algoritmos genéticos dos outros métodos evolutivos é a ênfase dada ao operador de cruzamento. Diferentes implementações de algoritmos genéticos adotam diferentes métodos de seleção, de cruzamento, e forma de representação dos indivíduos. Recentemente as estratégias evolutivas adotaram o cruzamento em seus algoritmos. Desse modo, pouco a pouco, a distinção entre os diferentes métodos evolutivos tornou-se pouco nítida (Bäck, 2000 a).

A seguir são descritas as características do algoritmo co-evolutivo proposto neste trabalho.

5.4.1 Representação

A representação de indivíduos de uma população envolve a codificação, pois dela cria-se o cromossomo, uma das principais etapas na construção de algoritmos genéticos. A forma como os alelos são distribuídos no cromossomo é importante e a escolha dos operadores de cruzamento e mutação dependerá diretamente da representação adotada para os cromossomos.

Como já foi dito anteriormente (2.3), uma estratégia mista para um jogador é definida como:

$$X = \{x = (x_1, \dots, x_m) \in R^m \mid \sum_{i=1}^m x_i = 1, x_i \geq 0, i = 1, \dots, m\} \quad (5.1)$$

No algoritmo proposto neste trabalho, cada cromossomo representa uma estratégia pra um dos jogadores. Os cromossomos serão representado com alelos reais, sendo que o número de *loci* no cromossomo é igual às opções de cada jogador, respeitando a restrição de que a soma dos valores dos *loci* de um cromossomo deve ser igual a 1, como ilustra a Figura 5.4:

0.21	0.79
0.10	0.90
0.34	0.66

Fig 5.4 Representação de um cromossomo para um jogo com duas opções

É importante ressaltar que na representação acima o valor de cada *loci* está completamente relacionado com os demais *loci* do cromossomo, uma vez que a restrição de que a soma desses valores tem que ser igual a 1 deve sempre ser respeitada. Esse fato é de extrema relevância para a concepção dos operadores genéticos, pois estes deverão manter a factibilidade dos indivíduos da população, respeitando (5.1).

Dessa forma, para um jogo com dois jogadores, serão criadas duas populações de indivíduos, sendo que cada indivíduo representa uma estratégia candidata ao problema.

5.4.2 Operadores genéticos

Para que o processo de evolução seja imitado é necessário que ocorram variações nos indivíduos da população. Isso garante diversidade da população, possibilitando ao algoritmo explorar diferentes soluções no espaço de busca.

Os operadores mais utilizados em computação evolutiva são o cruzamento (*crossover*) e a mutação. A mutação acrescenta diversidade na população quando modifica aleatoriamente os alelos dos cromossomos. O cruzamento é um operador que intensifica a busca em regiões promissoras, uma vez que promove a criação de cromossomos filhos com partes de cromossomos pais que possuam os melhores valores de *fitness*. Os indivíduos que possuem um bom desempenho no algoritmo, ou seja, que tem um valor de solução melhor que os demais, são selecionados para realizar o cruzamento. Dessa forma, o operador de cruzamento promove uma busca local, trocando informações entre os indivíduos promissores, promovendo uma busca por soluções melhores nas regiões promissoras.

A escolha de qual operador utilizar e de como implementá-lo depende do tipo de problema tratado e da representação adotada.

No algoritmo genético clássico a mutação ocorre a taxas reduzidas, como, por exemplo, um por cento. O alelo modificado geralmente é escolhido aleatoriamente. Todos os alelos tem igual probabilidade de sofrer mutação.

No algoritmo descrito nessa seção, o operador de mutação utilizado é similar ao adotado por algoritmos genéticos clássicos, adaptado para o caso onde os alelos tem valores reais ao invés de binários.

Seja m o número de decisões possíveis em um jogo, $x_1, x_2, \dots, x_n$, e seja x^k o indivíduo original e x^{k+1} o indivíduo após sofrer mutação. O indivíduo mutado é tal que:

$$x_z^{k+1} = \begin{cases} x_{z+1}^k, & \text{se } z \neq m \\ x_1^k, & \text{se } z = m \end{cases} \quad (5.2)$$

A diversidade da população mede o quanto os indivíduos são diferentes e é importante para facilitar a exploração eficiente do espaço de busca. Quanto mais diferentes forem as soluções da população, maior é a probabilidade do algoritmo encontrar um ótimo local. O operador de mutação é um dos mecanismos utilizados para auxiliar o algoritmo no controle da diversidade da população e é importante para evitar os ótimos locais do espaço de busca (Michalewicz, 1996). Evitando os ótimos locais o algoritmo co-evolutivo possibilita a criação de indivíduos em regiões diferentes do espaço de busca e tem mais chances de encontrar o ótimo global. Ou seja, em um algoritmo evolutivo, manter alguns indivíduos com *fitness* ruim na população pode ser uma estratégia interessante, pois ajuda o algoritmo a convergir para o ótimo global e não local.

O operador de cruzamento geralmente ocorre com maior probabilidade do que a mutação. Um operador de cruzamento bastante utilizado na literatura para cromossomos com representação real é o *crossover* aritmético (Michalewicz, 1996), que será utilizado nesse trabalho (5.3).

$$x = x_1\alpha + x_2(1 - \alpha) \quad (5.3)$$

onde x_1 e x_2 são dois indivíduos pais e x é o indivíduo resultante da operação de recombinação, com $\alpha \in [0, 1]$.

O *crossover* aritmético cria um indivíduo que é uma combinação convexa dos pais e garante a factibilidade das soluções resultantes, para cromossomos com representação no universo real.

5.4.3 Avaliação e seleção

No algoritmo proposto nesta seção, cada indivíduo da população é uma estratégia para um jogador. Para avaliar a estratégia de um indivíduo de uma população I, este deve jogar com outros indivíduos da população adversária. Em cada jogada o valor do jogo é anotado, e o *fitness* final do indivíduo será a média dos valores de todos os jogos que ele participou.

Seja $\tilde{A}$ a matriz de pagamentos nebulosa do jogo, x e y as estratégias mistas dos jogadores I e II respectivamente. Tem-se

$$F_{x^i} = \frac{\sum_{j=1}^n x^i a(\lambda)_{ij} y^j}{n} \quad \text{e} \quad F_{y^j} = \frac{\sum_{i=1}^m x^i a(\mu)_{ij} y^j}{m} \quad (5.4)$$

onde n é o número de estratégias da população II e F_{x^i} é o valor da função de *fitness* associada a decisão x^i . F_{x^i} é a média aritmética de todos os jogos que o jogador I participou com a decisão x^i em uma geração. Analogamente, para a população II temos F_{y^j} , que é o valor da função de *fitness* correspondente a decisão y^j , e m é o número de decisões da população I. Os valores de λ e μ são ajustados de acordo com o grau de pertinência utilizado, assim como em Maeda, 2003, sendo λ e $\mu \in [0,1]$. Por exemplo, seja o elemento $\tilde{a}_{ij} = (m_{ij}, h_{ij})$ da matriz de pagamentos nebulosa $\tilde{A}$. Sejam $m_{ij} = 5$ e $h_{ij} = 1$. Para $\lambda = 0.2$, temos:

$$\begin{aligned} a_{ij}(\lambda) &= m_{ij} + (1 - 2\lambda)h_{ij} \\ a_{ij}(\lambda) &= 5 + (1 - 2(0.2))1 \\ a_{ij}(\lambda) &= 5 + 0.6 = 5.6 \end{aligned}$$

Considere $\lambda = 0.8$. Dessa forma temos:

$$a_{ij}(\lambda) = 5 + (1 - 2(0.8))1$$

$$a_{ij}(\lambda) = 5 - 0.6 = 4.4$$

Portanto, fica claro que para valores de λ menores que 0.5, o valor de $a_{ij}(\lambda)$ será maior que m_{ij} . De forma análoga, para valores de λ maiores que 0.5, o valor $a_{ij}(\lambda)$ será menor que m_{ij} e quando $\lambda = 0.5$, $a_{ij}(\lambda)$ é igual ao núcleo do numero nebuloso $\tilde{a}_{ij}$. O mesmo acontece para $a_{ij}(\mu)$. A Figura 3.1 ilustra a relação entre $A(\lambda)$ e $\tilde{A}$.

Foi implantado no algoritmo um procedimento de busca local para sinalizar ao algoritmo que um individuo A está localizado em uma região promissora no espaço de busca. Essa busca encontra os indivíduos que estão na vizinhança de A em um raio r , no nosso caso $r \in [0.01, 0.05]$. Alguns indivíduos da vizinhança são selecionados aleatoriamente, e o *fitness* desses indivíduos é calculado. Se a media do *fitness* da vizinhança de A for maior do que o *fitness* do próprio individuo A , então aumentamos o *fitness* de A . Dessa forma, mostramos ao algoritmo que A provavelmente está localizado em uma região promissora do espaço de busca.

É importante ressaltar que apesar de distintas, as duas populações evoluem simultaneamente, de acordo com a característica co-evolutiva do algoritmo. O *fitness* de uma população está diretamente relacionado à outra. Se uma das populações apresentar problemas de convergência, certamente os indivíduos da outra população também terão problemas para convergir, ou seja, o algoritmo terá dificuldade para caminhar em direção ao ótimo. Por isso, faz-se necessário a escolha de bons operadores e um ajuste adequado das taxas de probabilidade que esses operadores ocorrerão na população.

O algoritmo proposto nesta seção também pode ser utilizado para resolver problemas de jogos nebulosos de soma não zero. Para isso, basta calcular o *fitness* de cada população como em (5.5).

$$F_{x_i} = \frac{\sum_{j=1}^n x_i a(\lambda)_{ij} y_j}{n} \quad \text{e} \quad (5.5)$$

$$F_{y_j} = \frac{\sum_{i=1}^m x_i b(\mu)_{ij} y_j}{m}$$

onde $a(\lambda)$ e $a(\mu)$ são os pagamentos nebulosos parametrizados do jogador I e II, respectivamente. De maneira análoga a (5.4), os valores de λ e μ são ajustados de acordo com o grau de pertinência desejado, sendo λ e $\mu \in [0,1]$.

O processo de seleção é fundamental na evolução e define quais elementos constituirão a próxima geração, ou seja, os indivíduos da população seguinte. Se o método de seleção utilizado não for eficiente, é possível que a cada geração os indivíduos se concentrem em uma região específica e convirja para um ótimo local. Por esse motivo nem sempre a melhor alternativa é selecionar apenas os melhores indivíduos. O algoritmo de seleção deve ser inteligente o suficiente para manter indivíduos na população que contribuem para escapar de ótimos locais.

A seleção ocorre geralmente antes do processo de cruzamento e de mutação, e pode ser probabilística ou determinística. Esse trabalho adota o algoritmo de seleção por torneio. Nesse método, escolhe-se aleatoriamente l indivíduos da população e o indivíduo com melhor *fitness* é selecionado. Essa operação é repetida r vezes, para uma população de tamanho r (Bäck, 2000 a). A Figura 5.5 ilustra o método de seleção por torneio.

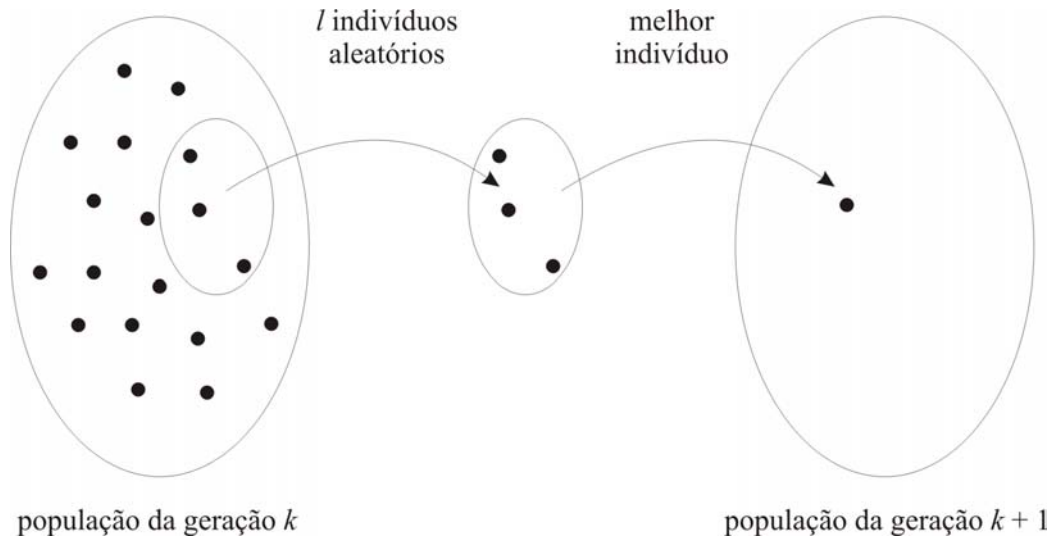


Fig 5.5 Método de seleção por torneio

Dessa maneira é possível selecionar os indivíduos com alto *fitness* e ainda manter a diversidade na população. Quanto menor o valor de l maior tende a ser a diversidade da nova geração. É claro que se l for igual a r , todos os indivíduos da população serão iguais, e a diversidade desaparece. Por outro lado se l for igual a 1 os indivíduos estarão sendo escolhidos aleatoriamente. Por isso é importante escolher um valor de l cuja relação de custo e benefício entre processamento e manutenção de diversidade seja equilibrada.

5.4.4 Reprodução

Após selecionar os indivíduos mais aptos e aplicar os operadores de cruzamento e mutação, é feita a reprodução. Nesse processo novas soluções são criadas, semelhantes às soluções existentes, porém com pequenas modificações.

A proximidade da solução pai com as soluções filhos pode variar de algoritmo para algoritmo. Nos exemplos discutidos nesse trabalho, como os valores dos alelos são pequenos, menores do que um, um valor considerado razoável seria de 0.01 a 0.05. A Figura 5.6 ilustra soluções pai e soluções filhos na região factível do problema.

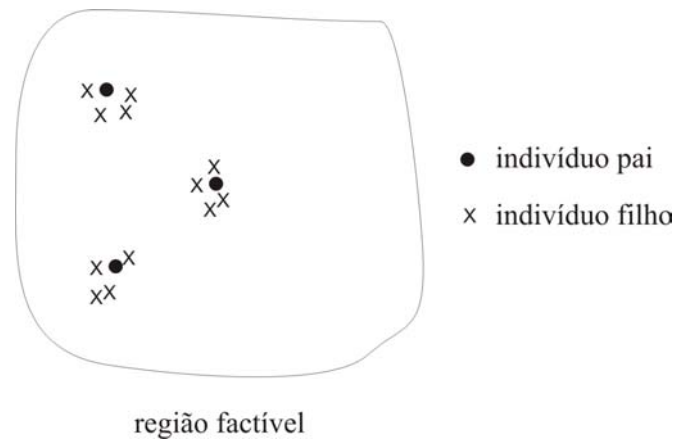


Fig 5.6 Indivíduos que representam soluções candidatas e a criação de indivíduos filhos

Por exemplo, um indivíduo $x = (0.8, 0.2)$ poderia gerar um indivíduo filho como $x = (0.83, 0.17)$.

O número de filhos criados, assim como os demais parâmetros probabilísticos, pode ser ajustado observando-se o desempenho do algoritmo ao longo do processamento. Podemos perceber pela figura que uma solução pai gera mais de uma solução filho. Por isso, o algoritmo de seleção, nesse caso o torneio, deve escolher um número de indivíduos bem menor que o tamanho da população, para que, na hora da reprodução, seja possível manter o número de indivíduos da população constante.

A partir dos itens apresentados acima, um algoritmo co-evolutivo pode ser construído seguindo os passos abaixo:

Passo 1:

Iniciar o processamento criando m vetores para cada população, com valores aleatórios que satisfaçam a condição (5.1). Essa é a primeira geração de indivíduos. Cada população representa estratégias para um dos jogadores.

Passo 2

Simular torneio. Indivíduos de cada população jogam contra indivíduos da população adversária, e seu *fitness* é calculado pelas equações (5.4), para os jogos de soma zero e (5.5) para os jogos de soma não zero. A busca local é aplicada e os indivíduos que estão localizados em um local promissor do espaço de busca recebem um acréscimo em seu *fitness*.

Passo 3:

Selecionar l indivíduos da população com o método de seleção por torneio.
Aplicar os operadores de cruzamento e seleção.

Passo 4:

Criar indivíduos filhos.

Passo 5:

Se (condição de parada) Fim, senão, voltar ao passo 2.

A condição de parada pode variar dependendo do problema. Nesse trabalho foi utilizado o número de iterações como condição de parada.

Dessa maneira, inicialmente cria-se uma população para cada jogador, com indivíduos representando possíveis estratégias para o jogo. O algoritmo manipula os indivíduos através dos operadores e a cada iteração uma nova geração de indivíduos é criada para cada população. Assim sendo, as populações evoluem simultaneamente e a tendência é que o *fitness* dos indivíduos melhore a cada iteração sendo que, na ultima iteração, o individuo com o melhor valor de *fitness* de uma população será a melhor estratégia para o jogador que representa aquela população.

5.5 Implementação e resultados

Diversos exemplos foram implementados para verificar o comportamento do algoritmo co-evolutivo na resolução de problemas de jogos nebulosos. A seguir, serão mostrados os resultados relevantes e as principais diferenças entre o algoritmo co-evolutivo e os métodos tradicionais.

5.5.1 Jogos nebulosos de soma zero

Os exemplos estudados nos capítulos anteriores foram resolvidos pelo algoritmo co-evolutivo. No primeiro exemplo, foi considerada a matriz de pagamentos nebulosa de Campos (1989). A matriz núcleos A e a matriz dispersão H são dadas em (5.6).

:

$$A = \begin{bmatrix} 180 & 156 \\ 90 & 180 \end{bmatrix} \quad H = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} \quad (5.6)$$

sendo um jogo de soma zero, onde o jogador I é o maximizador, e os valores dos elementos da matriz são números nebulosos triangulares. Portanto, A é o núcleo dos números nebulosos, e H é a dispersão utilizada.

A Figura 5.7 ilustra a evolução dos indivíduos em relação a média de *fitness* da população, por 300 gerações.

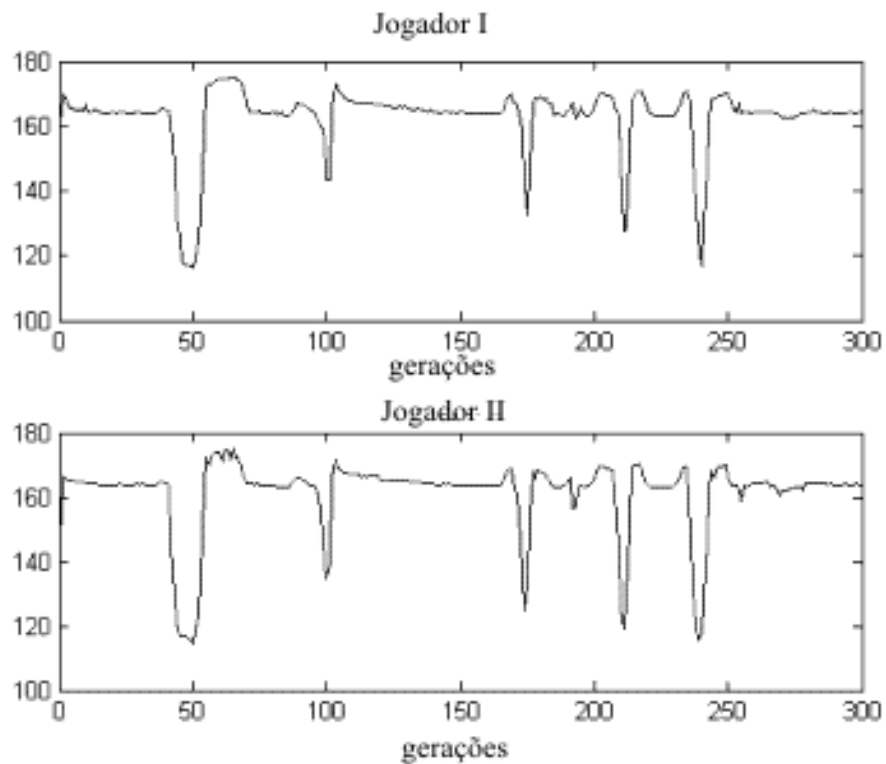


Fig 5.7. Fitness médio da população

Os parâmetros do algoritmo foram ajustados no decorrer dos testes. A taxa de mutação utilizada foi de 0.01 e a taxa de cruzamento de 0.5. Vários tamanhos de população foram implementados. Em geral uma população com 60 indivíduos teve um melhor custo benefício em termos de tempo de processamento e taxa de convergência. A solução clássica para esse problema é de $x = (0.78, 0.22)$ e $y = (0.22, 0.78)$, estratégias dos jogadores I e II, respectivamente, e o valor do jogo de 161. Os resultados obtidos pelo algoritmo co-

evolutivo são semelhantes aos resultados obtidos através dos métodos estudados no capítulo três.

O algoritmo foi executado várias vezes e em todas elas o algoritmo convergiu para a solução de equilíbrio. A Tabela 5.1 ilustra os valores correspondentes a melhor solução encontrada cada vez que o algoritmo foi executado

x	y	Valor do jogo
(0.7941, 0.2059)	(0.1550, 0.8450)	161.0233
(0.5975, 0.4025)	(0.1289, 0.8711)	162.8394
(0.7228, 0.2772)	(0.1378, 0.8622)	161.6052
(0.8968, 0.1032)	(0.1279, 0.8721)	160.0422
(0.8245, 0.1755)	(0.2033, 0.7967)	161.8260
(0.8327, 0.1673)	(0.1371, 0.8629)	160.6905

Tab 5.1 Estratégias para o jogo de soma zero (5.6)

Compare esses resultados com os resultados obtidos nos algoritmos do Capítulo 3, da Tabela 5.2, repetida abaixo por conveniência:

x	y	P	Método
(0.7120, 0.2880)	(0.2880, 0.7120)	160	Lee-Kwang e Lee
(0.7895, 0.2105)	(0.2105, 0.7895)	161	Campos 1,2 e 4
(0.7895, 0.2105)	(0.2105, 0.7895)	161	Maeda
(0.7692, 0.2308)	(0.1892, 0.8108)	161.1019	α -cortes ($\alpha = 0.5$)

Tab 5.2 Resultado do exemplo (5.6) com outros métodos

Percebe-se que o mesmo cenário leva o algoritmo de Campos e Maeda a convergirem para o mesmo ótimo. O modelo de ordenação de Lee-kwang e Lee retorna um valor ligeiramente diferente, porém sempre que o algoritmo for executado nessas condições o resultado será o mesmo.

Já o algoritmo co-evolutivo oferece soluções diferentes a cada execução, mas sempre com valores de jogo parecidos. Essas soluções formam um conjunto de soluções para o problema e oferece ao decisor maior flexibilidade na decisão.

Para verificar a diversidade populacional D foi feita a diferença entre as notas do

melhor e do pior indivíduo da população a cada iteração:

$$D = P_{\text{melhor indivíduo}} - P_{\text{pior indivíduo}}$$

Pequenas variações nos valores de ocorrência de cruzamento e mutação não afetaram as soluções significativamente. No entanto, uma mudança drástica na taxa de mutação, de 0.01 para 0.1 por exemplo, gera uma perturbação muito grande na população, permitindo a perda de soluções muito boas e impedindo a convergência do algoritmo. A Figura 5.8 ilustra a influência de uma alta taxa de mutação na diversidade da população.

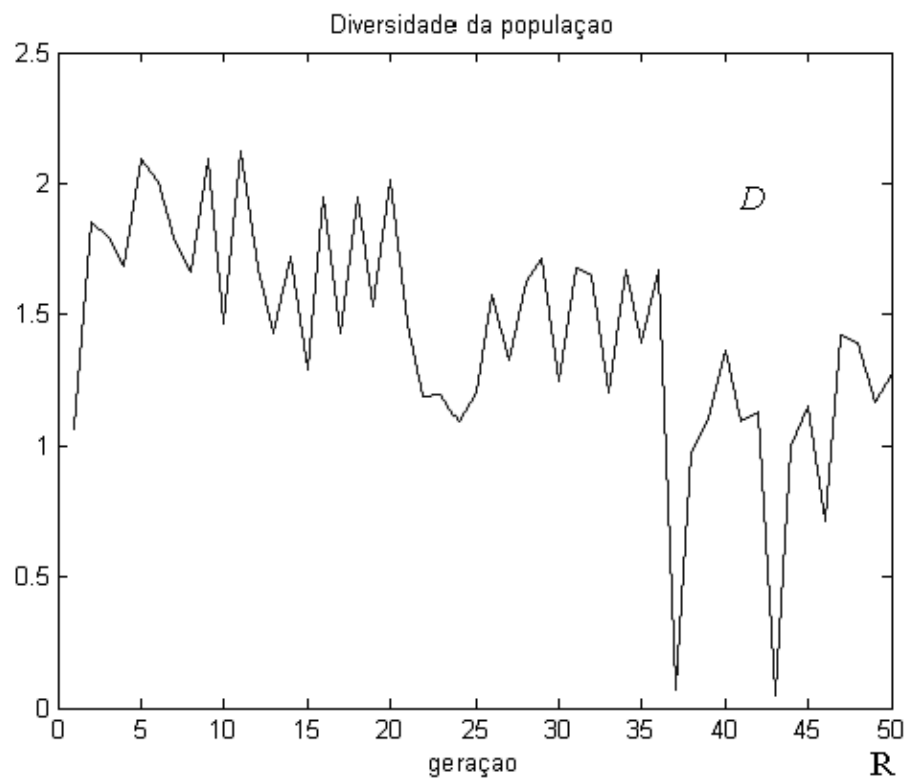


Fig 5.8. Alta taxa de mutação proporciona maior diversidade.

Por outro lado, valores muito baixos para a ocorrência de mutação, como 0.001, fazem com que a diversidade da população seja reduzida drasticamente, comprometendo a convergência do algoritmo. A Figura 5.9 ilustra a influência de uma baixa taxa de mutação na diversidade da população.

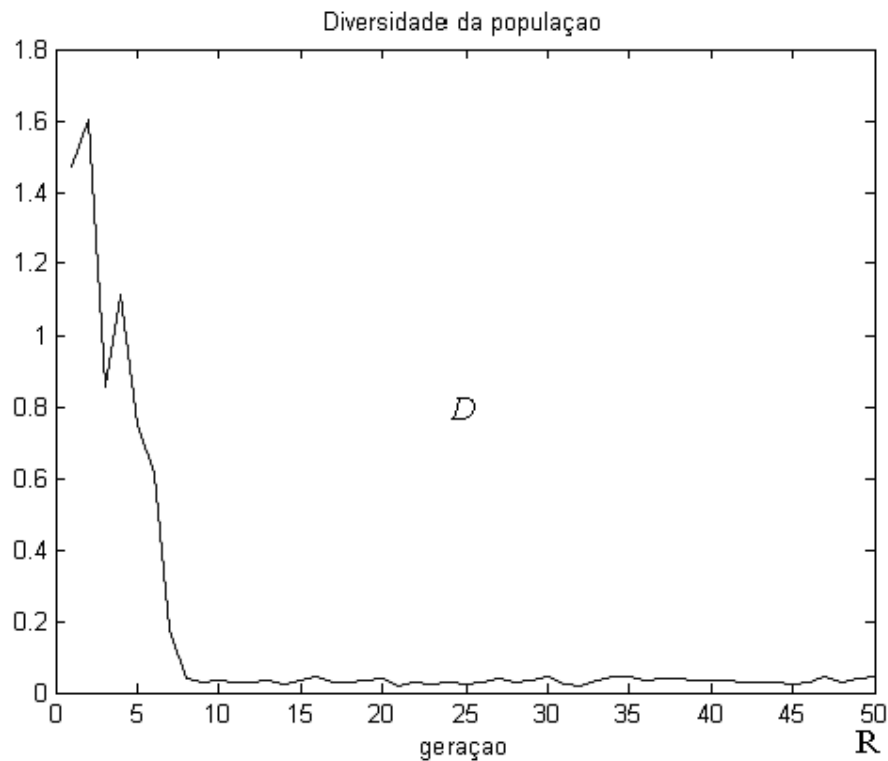


Fig 5.9. Baixa taxa de mutação proporciona menor diversidade

Percebe-se que com uma baixa taxa de mutação na décima geração a diversidade da população é praticamente nula.

Alterações na taxa de cruzamento também foram analisadas. Taxas de ocorrência maiores que cinquenta por cento causaram uma perturbação muito grande no algoritmo. Por outro lado, taxas inferiores a cinquenta por cento afetaram a exploração do espaço de busca uma vez que impediram a troca de informações entre as soluções e muitas vezes fizeram com que o algoritmo não encontrasse a solução de equilíbrio.

5.5.2 Jogos nebulosos de soma não zero

Essa sessão apresenta resultados obtidos na solução de jogos nebulosos de soma não zero. As matrizes núcleo para o jogador I e II são, respectivamente (5.7).

$$A = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \quad (5.7)$$

e as matrizes dispersão H_A e H_B (5.8):

$$H_A = \begin{bmatrix} 1 & 1 \\ 1.5 & 1 \end{bmatrix} \quad H_B = \begin{bmatrix} 1 & 1 \\ 1 & 1.5 \end{bmatrix} \quad (5.8)$$

Após aproximadamente 30 gerações a solução de equilíbrio do jogo foi encontrada pelo algoritmo co-evolutivo. As Figuras 5.10 e 5.11 ilustram, respectivamente, a evolução do valor médio de *fitness* da população e do melhor indivíduo ao longo das gerações.

O valor das estratégias encontrado pelo algoritmo co-evolutivo foi semelhante ao encontrado pelo método de programação bilinear apresentado no Capítulo 4. O algoritmo co-evolutivo, como no exemplo anterior, foi executado várias vezes e os resultados são apresentados na Tabela 5.3.

x	y	Valor do jogo Jogador I	Valor do jogo Jogador II
(0.5028, 0.4972)	(0.4956, 0.5044)	0.4913	1.5056
(0.5496, 0.4504)	(0.5552, 0.4448)	0.6050	1.6046
(0.5616, 0.4384)	(0.5438, 0.4562)	0.5823	1.6286
(0.5343, 0.4657)	(0.5321, 0.4679)	0.5621	1.5709
(0.4804, 0.5196)	(0.5054, 0.4946)	0.5111	1.4605
(0.5179, 0.4821)	(0.5205, 0.4795)	0.5403	1.5365

Tab 5.3 Estratégias obtidas para o jogo de soma não zero

Pelos resultados obtidos conclui-se que o algoritmo co-evolutivo proposto resolve satisfatoriamente os problemas de jogos nebulosos de soma zero e de soma não zero.

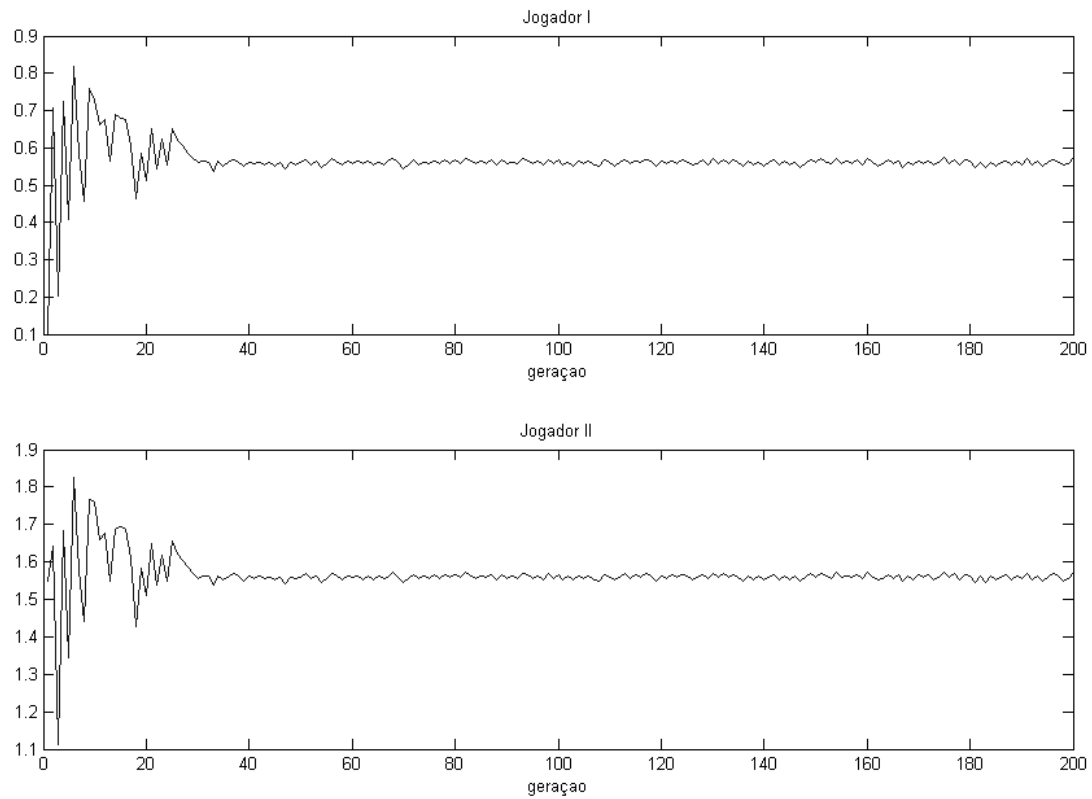


Fig 5.10. Evolução do *fitness* médio da população em um jogo de soma não zero

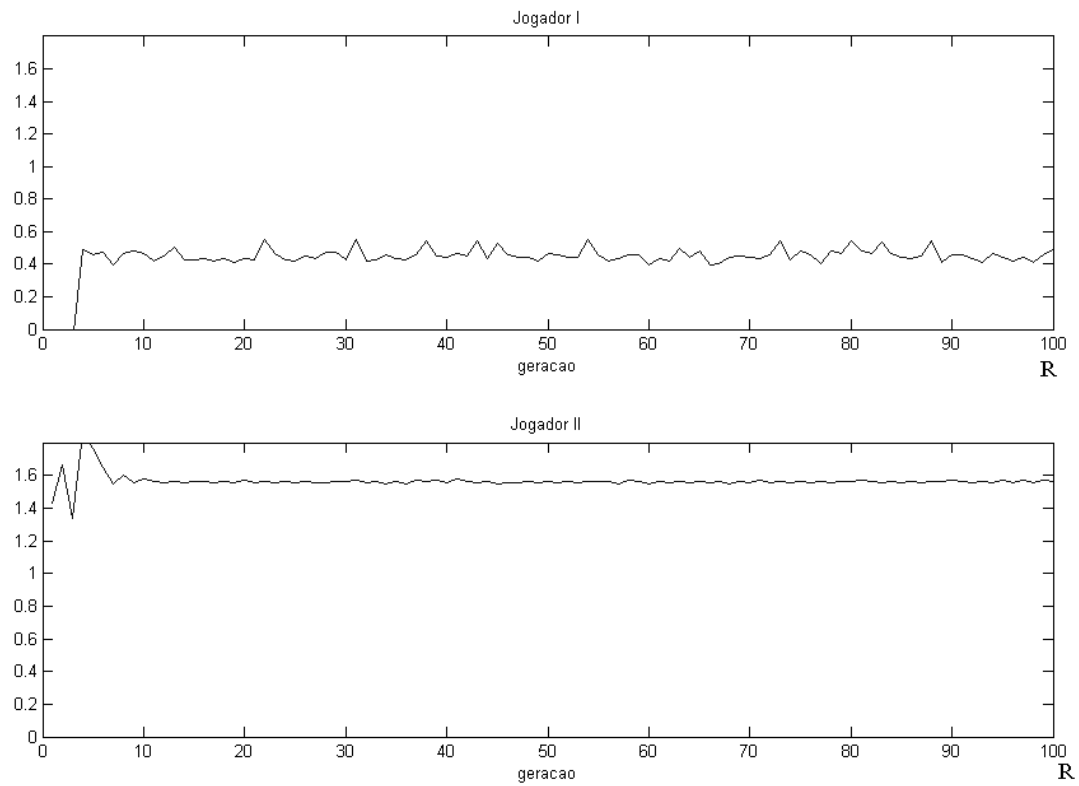


Fig 5.11. *Fitness* do melhor individuo

Diante do fato de o algoritmo co-evolutivo considerar uma população de soluções, onde o ponto (ou os pontos) de equilíbrio é encontrado, sentiu-se a necessidade de testar o algoritmo para o exemplo de mercado de energia de Singh (1999), resolvido pelo algoritmo de α -cortes do Capítulo 4. Como explicado no Capítulo 4, o exemplo de Singh possui duas soluções de equilíbrio, conforme a matriz de pagamentos nebulosa (5.9).

$$A = \begin{bmatrix} 2250 & 2625 \\ 700 & 2800 \end{bmatrix} \quad B = \begin{bmatrix} 2250 & 700 \\ 2625 & 2800 \end{bmatrix} \quad (5.9)$$

$$H_A = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix} \quad H_B = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix}$$

O exemplo em (5.9) é interessante para verificar se o algoritmo amostra uniformemente os dois pontos de equilíbrio, respectivamente $x = (1,0)$, $y = (1,0)$ e $x = (0,1)$, $y = (0,1)$. A Figura 5.12 ilustra a evolução do *fitness* da população no exemplo de Singh.

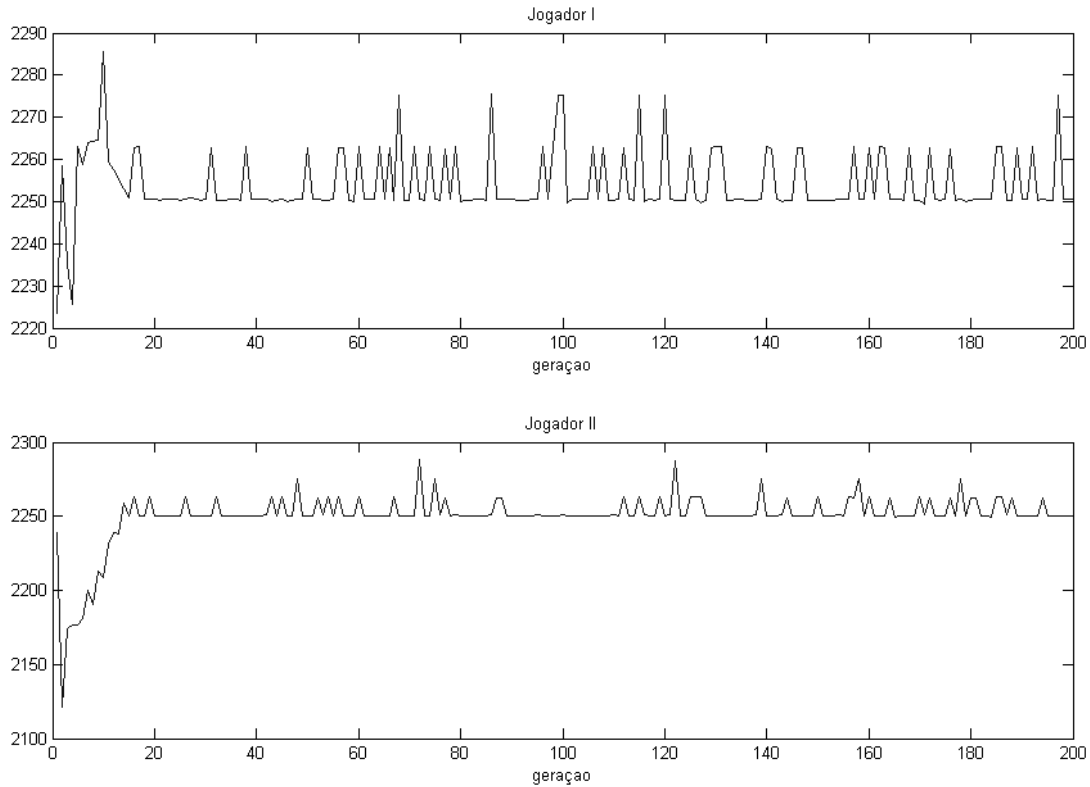


Fig 5.12. Evolução do *fitness* da população no exemplo de Singh

O resultado foi interessante. Sempre que o algoritmo foi executado, diferentemente do imaginado inicialmente, o algoritmo não amostrou os dois pontos de equilíbrio, e a solução de equilíbrio obtida foi $x = (1,0)$, $y = (1,0)$. Esse resultado tem contudo uma explicação simples.

Como já mencionado no Capítulo 4, o ponto de equilíbrio apontado por Singh (1999), $x = (0,1)$ e $y = (0,1)$, é o melhor resultado do jogo para ambos os jogadores. No entanto, deverá haver um acordo entre os dois jogadores, ou então eles terão que confiar um no outro, para que ambos saiam ganhando. Isso porque, se qualquer um dos jogadores escolher a estratégia $(0,1)$, digamos, o jogador α , e o jogador adversário (jogador β) não o fizer, o prejuízo de α será muito maior do que se o jogador α tivesse escolhido a estratégia $(1,0)$. Em outras palavras, sem conhecer a estratégia do jogador adversário e se não houver comunicação entre os jogadores, não compensa para nenhum deles arriscar a estratégia $(0,1)$. Apesar do lucro ser maior, o risco de perder é maior ainda.

Este fato se refletiu no algoritmo co-evolutivo. Isso porque, quando um indivíduo da população tinha a estratégia $(0,1)$, ao participar do torneio e jogar com todos os tipos de estratégia do jogador adversário, o valor de *fitness* final, ou seja, a média aritmética de todas as jogadas, fazia com que a nota do jogador fosse baixa, inviabilizando sua existência na população. Daí o fato de o algoritmo co-evolutivo ter amostrado apenas o ponto de equilíbrio $(1,0)$, o equilíbrio mais vantajoso, para ambos os jogadores. Esse fato é interpretado como uma vantagem do algoritmo co-evolutivo uma vez que para uma matriz muito grande e com vários pontos de equilíbrio, o algoritmo amostrará apenas o equilíbrio mais vantajoso, onde a recompensa dos jogadores é maior.

Para comprovar esse fato, foi criada a matriz de pagamentos nebulosa (5.10).

$$\begin{aligned} A &= \begin{bmatrix} 8 & 10 \\ 10 & 8 \end{bmatrix} & B &= \begin{bmatrix} 8 & 10 \\ 10 & 8 \end{bmatrix} \\ H_A &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & H_B &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{aligned} \tag{5.10}$$

No jogo (5.10) existem dois equilíbrios, o primeiro com $x = (0,1)$ e $y = (1,0)$, e o

segundo com $x = (1,0)$ e $y = (0,1)$. Para ambos os casos o pagamento do jogo é igual a 10 para os dois jogadores, ou seja, os valores do jogo são iguais para os dois pontos de equilíbrio.

Nesse caso o algoritmo co-evolutivo foi capaz de amostrar os dois pontos de equilíbrio, um para cada execução. O jogo (5.10) foi executado inúmeras vezes e ele sempre convergiu ou para $x = (0,1)$ e $y = (1,0)$, ou para $x = (1,0)$ e $y = (0,1)$. Dessa maneira, o algoritmo co-evolutivo se mostrou mais eficiente que os algoritmos convencionais por permitir amostrar todos os equilíbrios de um jogo, quando este possuir mais de um equilíbrio, com valores de jogo semelhantes.

Esse exemplo foi importante para entender melhor o funcionamento do algoritmo co-evolutivo e o comportamento de indivíduos não cooperativos através de uma abordagem co-evolutiva.

5.6 Comparação com os métodos clássicos

Nos capítulos anteriores foram apresentadas técnicas de resolução de jogos nebulosos não cooperativos de soma zero e de soma não zero. Foi demonstrado o paralelo entre a formulação desses jogos e a programação linear e bilinear.

Nesse capítulo foi apresentada uma abordagem co-evolutiva para a resolução dos mesmos jogos nebulosos resolvidos pelos métodos tradicionais.

Uma das principais diferenças observadas foi que o algoritmo co-evolutivo nem sempre convergiu para o mesmo valor de estratégia ótima, como aconteceu com os métodos baseados em programação linear. Isso se deve à natureza probabilística dos algoritmos genéticos. Outra causa é a característica populacional do algoritmo, que permite explorar regiões distintas do espaço de busca simultaneamente ao final abrangendo a vizinhança do ótimo, e não apenas uma única solução ótima. Dessa forma o algoritmo co-evolutivo aponta soluções diferentes para o mesmo problema, com pequenas alterações no resultado final do jogo. Essa característica pode ser observada como uma vantagem dos algoritmos co-evolutivos sobre os algoritmos clássicos uma vez que, na prática, o decisor se beneficiará de um conjunto de boas estratégias e não apenas de uma única estratégia ótima. Em problemas reais essa diversificação de resultados pode ser fundamental na escolha da

melhor decisão, uma vez que dificilmente o problema formulado matematicamente abrangerá todas as peculiaridades de uma situação real.

Outra vantagem do algoritmo co-evolutivo é a possibilidade de tratar diferentes tipos de problemas. Enquanto métodos tradicionais precisam de informações de primeira e segunda ordem, como vetor gradiente e matriz Hessiana, o algoritmo co-evolutivo depende apenas da existência de uma função de *fitness* bem definida (que pode ou não ser linear) e de uma representação adequada. Isso permite considerar diferentes tipos de problemas, sem se preocupar com linearidade, convexidade e outras propriedades de funções.

Talvez a característica mais interessante do algoritmo co-evolutivo seja a capacidade de amostrar todos os pontos de equilíbrio do jogo, quando este possuir mais de um equilíbrio.

Outra característica do algoritmo apresentado é a capacidade de resolver jogos de soma zero ou de soma não zero, apenas modificando a função de *fitness* utilizada.

Uma desvantagem do método co-evolutivo em relação aos métodos clássicos é o tempo de processamento. Em todos os casos, o algoritmo evolutivo demorou mais para encontrar a estratégia ótima do que os métodos baseados em programação linear. Em geral, para uma população com 60 indivíduos, o algoritmo levou em média 15 segundos para alcançar 100 gerações. O algoritmo de programação linear de Maeda, por exemplo, levou 2 segundos para encontrar a estratégia ótima do mesmo jogo. Os testes foram feitos em um computador com Windows 2000 Professional, processador AMD de 800 Mhz e 256 MB de memória RAM. Dentre as causas do processamento mais lento no algoritmo co-evolutivo temos o procedimento de busca local, a aplicação de operadores, as taxas e o tamanho da população. Todos esses fatores podem aumentar ou diminuir o tempo de processamento do algoritmo co-evolutivo.

5.7 Resumo

Esse capítulo apresentou um algoritmo co-evolutivo para obter soluções de equilíbrio para jogos nebulosos não cooperativos com dois jogadores.

Em co-evolução, cada população está associada a um jogador, e um indivíduo da população é avaliado considerando a interação desse indivíduo com a população adversária,

através de simulações de torneios de jogos.

Mostrou-se que os operadores convencionais da computação evolutiva, como *crossover*, mutação e seleção podem ser utilizados. No entanto, a taxa de ocorrência de cada operador deve ser minuciosamente ajustada como parâmetros específicos de cada problema. Com a escolha correta dos operadores, vários testes foram implementados e os resultados convergiram para o ótimo em todos os casos.

O algoritmo proposto resulta não apenas em uma solução ótima, mas em um conjunto de soluções próximas ao ótimo, oferecendo ao decisor uma ferramenta eficaz na hora de escolher a melhor solução para um problema de decisão.

É importante ressaltar que o algoritmo apresentado pode ser utilizado para jogos nebulosos de soma zero e de soma não zero, ou ainda para jogos sem a nebulosidade na matriz de pagamentos. O algoritmo é capaz também de encontrar, quando existir, mais de um ponto de equilíbrio do jogo.

Capítulo 6

Conclusão

Esse trabalho abordou conceitos da teoria de jogos nebulosos. Foram estudadas as propriedades e características das soluções de equilíbrio para os diferentes tipos de jogos nebulosos, incluindo jogos cooperativos e suas coalizões, mas enfatizando jogos não cooperativos de soma zero e de soma não zero.

Através desse trabalho foi possível analisar e comparar diferentes técnicas de soluções de jogos nebulosos encontrados na literatura e verificar a sua aplicabilidade em através de exemplos de aplicação.

6.1 Contribuições

Foram sugeridos dois algoritmos para a resolução de jogos nebulosos. O primeiro é um algoritmo bilinear baseado em α -cortes para a resolução de jogos nebulosos não cooperativos de soma não zero. O algoritmo apresentado difere do trabalho de Chen & Larbani (2005) por resolver jogos nebulosos de soma não zero, utilizando o algoritmo de Konno (1976) modificado para representar os limites superior e inferior dos α -cortes das matrizes de pagamentos.

O segundo traduz-se em uma abordagem co-evolutiva para jogos nebulosos em geral. A abordagem utiliza conceitos de algoritmos evolutivos, combinando técnicas de algoritmos genéticos e estratégia de evolução para gerar um algoritmo capaz de encontrar uma solução de equilíbrio para jogos nebulosos não cooperativos de soma zero e de soma não zero.

Uma comparação entre o método de co-evolução e os métodos implementados encontrados na literatura também foi elaborada.

As duas abordagens sugeridas nesse trabalho foram desenvolvidas para a resolução de jogos nebulosos mas podem, evidentemente, ser aplicadas na solução de jogos clássicos.

6.2 Trabalhos futuros

A teoria de jogos nebulosos é um tema ainda pouco estudado na literatura, porém existem inúmeras aplicações emergentes que a torna uma área promissora de pesquisa e desenvolvimento.

Alguns dos temas não abordados aqui e que merecem atenção em trabalhos futuros são os jogos nebulosos iterativos com memória, onde as decisões são armazenadas e o jogador pode avaliar decisões passadas para aprender a melhor estratégia a cada situação enfrentada.

Existem alguns trabalhos nessa direção, principalmente seguindo os passos de Axelrod (1984). No entanto, nesse caso foram encontrados trabalhos que considerassem apenas métodos iterativos com estratégia pura, ou melhor, não foi encontrado um método que considerasse estratégias mistas, como no Capítulo 5, e memória de decisões. Essa possivelmente é uma extensão a ser feita no estudo de jogos nebulosos com estratégia mista.

Outros trabalhos poderiam utilizar os métodos apresentados aqui ou outros da literatura aplicados a diferentes problemas, como em robótica, sistemas de energia elétrica, bio-combustíveis, sistemas multiagentes, biologia. Por exemplo, a teoria de jogos pode se traduzir em um mecanismo para analisar a cooperação entre espécies. A idéia é analisar, à luz da teoria de jogos, as diferentes situações de cooperação e antagonismo encontrados na natureza.

Uma extensão interessante desse trabalho consiste em resolver o problema bilinear para jogos nebulosos de soma não zero com um outro algoritmo que não o algoritmo de Konno para, assim, obter um algoritmo de natureza global e não local. Um estudo mais detalhado sobre a Conjectura estabelecida do Capítulo 4 também se faz necessário.

Referências bibliográficas

Amaral, W. M. e Gomide, F. (2007a), "An algorithm to solve two-person non-zero sum fuzzy games", em Castillo, O., Melin, P., Ross, O., Cruz, R., Pedrycz, W. e Kacprzyk, J. (Eds), *Theoretical Advances and Applications of Fuzzy Logic and Soft Computing, Advances in Soft Computing*, Springer, vol. 42, pp. 296-302, 2007.

Amaral, W. M. e Gomide, F. (2007b), "A coevolutionary approach to solve fuzzy games", em Bello, R., Falcon, R., Pedrycz, W. e Kacprzyk, J. (Eds.) , *Granular Computing: at the Junction of Fuzzy Sets and Rough Sets, Studies in Fuzziness and Soft Computing*, Springer.

Andreoni, J., e Varian, H. (1993), "Pre-play contracting in the prisoner's dilemma" - *Research Report*, University of Wisconsin, USA.

Aubin, J. P. (1974a), "Coeur et valeur des jeux flous à paiements latéraux", *C. R. Acad. Sci, Paris Ser.A-B*, 279:891-896.

Aubin, J. P. (1974b), "Coeur et equilibres des jeux flous sans paiements latéraux", *C. R. Acad. Sci. Paris Ser. A-B*, 279:963-966.

Aubin, J. P. (1979), *Mathematical Methods of Game and Economic Theory*, North-Holland.

Aubin, J. P. (1984), "Cooperative fuzzy game: the static and dynamic points of view", *TIMS/Studies in the Management Science*, 20:407-428.

Axelrod, Robert M. (1984), *The Evolution of Cooperation*. Basic Books, New York, USA.

Bäck, T., Fogel, D.B. e Michalewicz, Z. (2000a), *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing, Bristol, United Kingdom.

Bäck, T., Fogel, D.B. e Michalewicz, Z. (2000b.), *Evolutionary Computation 2: Advanced Algorithms and Operators*, Institute of Physics Publishing, Bristol, United Kingdom.

Basar, T. e Olsder, G. J. (1982), *Dynamic Noncooperative Game Theory*. Academic Press, New York, USA

Bazaraa, M., Jarvis, J., Sherali, H. (2005). *Linear Programming and Network Flows*, 3rd Edition, Wiley Interscience, Hoboken, NJ, EUA.

Bazaraa, M. S. e Shetty, C. (1979), *Nonlinear Programming Theory and Algorithms*. John Wiley and Sons, New York, USA.

Bellman, R. E. e Zadeh, L. A. (1970), "Decision making in a fuzzy environment", *Management Science*, 17:141-164.

Bergstresser, K. e Yu, P.L. (1977), "Domination structure and multicriteria problems in n-person games", *Theory and Decision*, 8:5-48.

Billot, A. (1992), *Economic Theory of Fuzzy Equilibria*, Springer-Verlag, Berlin Heidelberg New York.

Blackwell, D. (1956), "An analog of the minimax theorem for vector payoffs", *Pacific Journal of Mathematics*, 98:1-8.

Borges, P., Pacheco, R., Khator, S. e Barcia, R. (1995), "A fuzzy approach to the prisoner's dilemma" - *BioSystems*, July, 1995.

Buckleya , James J. e Feuringb, Thomas (1998), "Evolutionary algorithm solution to fuzzy problems: fuzzy linear programming". *Fuzzy Sets and Systems*, 109:35–53.

Buckleya , James J. e Feuringb, Thomas (1998), "Linear and non-linear fuzzy regression: evolutionary algorithm solutions". *Fuzzy Sets and Systems*, 112:381–394.

Buckley, J. J. (1984), "Multiple goal non-cooperative conflicts under uncertainty: a fuzzy set approach". *Fuzzy Sets and Systems*, 13:107-124.

Butnariu, D. (1978), "Fuzzy games; a description of the concept", *Fuzzy Sets and Systems*, 1:181-192.

Butnariu, D. (1980) "Stability and Shapley value for an n-persons fuzzy game", *Fuzzy Sets and Systems*, 4:63-72.

Butnariu, D. (1987), "Values and cores of fuzzy games with infinitely many players", *International Journal of Game Theory*, 16:43-68.

Campos, L. (1989), "Fuzzy linear programming models to solve fuzzy matrix games." *Fuzzy Sets and Systems*, 32:275–289.

Chankong, V e Haimes, Y. (1983), "*Multiobjective Decision Making, Theory and Methodology*". North-Holland, New York, USA.

Chen, Y. e Larbani, M. (2005), "Two-person zero-sum game approach for fuzzy multiple attribute decision making problems." *Fuzzy Sets and Systems*, 157:34–51.

Chen, H., Wong, K., Nguyen, D., e Chung, C. (2006), "Analyzing oligopolistic electricity market using coevolutionary computation.", *IEEE Trans. Power Systems*, 21:1:143-152.

Contini, M. Olivtti, I. e Milano, C. (1966), "A decision model under certainty with multiple payoffs", in A. Mensch (ed.) *Theory of Games; Techniques and Applications*, American Elsevier, New York, pp. 50-63.

Cook, W. D. (1976), “Zero-sum games with multiple goals”, *Naval Research Logistics Quarterly*, 23:615-622.

Cordón, O. e Gomide, F. e Herrera, F. e Hoffmann, F. e Magdalena, L. (2004), “Ten years of genetic fuzzy systems: current framework and new trends.” *Fuzzy Sets and Systems*, 141:5–31.

Darwin, C (1859), *The Origin of Species*, John Murray, London, England.

Delgado, M. R. (2002), “Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva”. *Tese de Doutorado. Faculdade De Engenharia Elétrica e de Computação*, UNICAMP.

Derks, J.J.M. e Tijs, S.H. (1986a), “Totally balanced multi-commodity games and flow games”, *Methods of Operations Research*, 54:335-347.

Derks, J.J.M. e Tijs, S.H. (1986b), “Stable outcome for multi-commodity flow games”, *Methods of Operations Research*, 55:493-504.

Ficici, S. G., Melnik, O. e Pollack, J. B. (2000), “A game-theoretic investigation of selection methods used in evolutionary algorithms” *Congress on Evolutionary Computation*, IEEE Press, 2:880-887.

Ficici, S.G. e Pollack, J. B. (2000), “A game-theoretic approach to the simple coevolutionary algorithm.” *Parallel Problem Solving from Nature VI*, M. Schoenauer, Springer Verlag. p 467-476.

Ficici, S. G. e Pollack, J. B (2001), “Game theory and the simple coevolutionary algorithm: Some preliminary results on fitness sharing” *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*, p 2-8

Fogel, L. J., Owens, A. J. e Walsh, M. J. (1966), *Artificial Intelligence Through Simulated Evolution*. Wiley, New York.

Goh, C. K., Quek, H. Y., Teoh E. J. e Tan, K. C. (2005), “Evolution and incremental learning in the iterative prisoner's dilemma” *The 2005 IEEE Congress on Evolutionary Computation*,.3:2629:2636

Golbeck, J. (2002), “Evolving strategies for the prisoner's dilemma: advances in intelligent systems”, *Fuzzy Systems, and Evolutionary Computation*. February 2002, p 299-306.

Harsanyi, J. C. (1977), *Rational Behavior and Bargaining Equilibrium in Games and Social Situations*, Cambridge University Press, New York.

Hingston, P. e Kendall, G. (2004), “Learning versus evolution in iterated prisoner's dilemma”, *Congress on Evolutionary Computation, CEC*, 1:364-372

Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1st ed.

Klir, G. e Yuan, B. (1995), *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, New Jersey.

Konno, H. (1976), “A cutting plane algorithm for solving bilinear programs.” *Mathematical Programming*. 11:14-27.

Koza, J. R. (1992), *Genetic Programming*. MIT Press. Cambridge, MA

Lee-Kwang, H. e Lee, J. H. (1999), “A method for ranking fuzzy numbers and its application to decision-making.” *IEEE Trans. on Fuzzy Systems*. 7:677-685.

Luenberger, D. G. (1984), *Linear and Nonlinear Programming*, 2nd Ed., Addison Wesley, Massachusetts.

Maeda, T. (2003), “On characterization of equilibria strategy of two-person zero-sum games with fuzzy payoffs.” *Fuzzy Sets and Systems*, 139:283–296.

Maeda, T. (2000), “Characterization of the equilibrium strategy of the bimatrix game with fuzzy payoff.” *Journal of Mathematical Analysis and Applications*, 151:885–896.

Mayr, E. (1988), *Toward a New Philosophy of Biology: Observations of an Evolutionist*, The Belknap Press of Harvard University Press, Cambridge, MA, and London, GB.

Michalewicz, Z. (1996), *Genetic algorithms + Data Structures = Evolution Programs*, 3rd edition, Springer-Verlag Berlin Heidelberg New York.

Neumann, J von e Morgenstern, O. (1944), *Theory of Games and Economic Behavior*, Wiley, New York.

Nishizaki, I. e Sakawa, M. (1995), “Equilibrium solutions for multiobjective bimatrix games incorporating fuzzy goals.” *Journal of Optimization Theory and Applications*, 86:433-458.

Nishizaki, I. e Sakawa, M. (1998), “Fuzzy cooperative games arising from linear production programming problems with fuzzy parameters.” *Fuzzy Sets and Systems*, 114:11-21.

Nishizaki, I. e Sakawa, M. (2000), “Equilibrium solutions in multiobjective bimatrix games with fuzzy payoffs and fuzzy goals” *Fuzzy Sets and Systems*, 111:99-116.

Nishizaki, I. e Sakawa, M. (2001), *Fuzzy and Multiobjective Games for Conflict Resolution*. New York. Physica-Verlag.

- Nguyen, H. T. e Walker, E. A. (1999), *A First Course in Fuzzy Logic*. CRC Press.
- Olsson, B. (2001), “Co-evolutionary search in asymmetric spaces.” *Information Sciences*, 133:103–125.
- O’Riordan, C., Griffith, J., Newell, J. e Sorensen, H. (2004), “Co-evolution of strategies for an n -player dilemma” - *Congress on evolutionary computation, CEC*, Portland, Oregon, 2:1625-1630.
- Pedrycz, W. e Gomide, F. (1998), *An Introduction to Fuzzy Sets: Analysis and Design* The MIT Press. Cambridge, Massachusetts.
- Pindyck, R. S. e Rubinfeld, D. L. (2002), *Microeconomia*. Pearson Education do Brasil, São Paulo.
- Pollack, J. e A. Blair (1998), “Coevolution in the successful learning of backgammon strategy.” *Machine Learning* 32:225–240.
- Ponsard, C. (1986), “Spatial oligopoly as fuzzy games”, In C. Ponsard and B. Fustier (eds.) *Fuzzy Economics and Spatial Analysis*, Coll. I.M.E, Librairie de l’Université, Dijon, pp 57-67.
- Ponsard, C. (1987), “Nash fuzzy equilibrium theory and applications to a spatial duopoly”, *European Journal of Operational Research*, 31:376-384.
- Ramík, J. e Rimanek, J. (1985), “Inequality relation between fuzzy numbers and its use in fuzzy optimization”, *Fuzzy Sets and Systems*, 16:123–150.
- Rardin, R. (1998), *Optimization in Operations Research*. Prentice Hall, Upper Saddle River, NJ, EUA.

Rechenberg, I. (1973), *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart.

Riechamann, T. (2000), “Genetic algorithms learning and evolutionary games”. *Journal of Economics Dynamics and Control*, 25:1019-1037.

Rosin, C. e R. Belew (1995), “Methods for competitive co-evolution: finding opponents worth beating.” L. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA)*, pag 373–380. Morgan Kaufmann.

Sakawa, M. e Nishizaki, I. (1992), “A solution concept based on fuzzy decision in n-person cooperative games”, in R. Trappl (ed.) *Cybernetics and Systems Research '92*, World Scientific Publishing, pp. 423-430.

Shapley, L.S. (1953), “A value for n -person games” in H.W.Kuhn and A. W. Tucker (eds.), *Contribution to the Theory of Games, 2, Annals of Math. Studies 28*, Princeton University Press.

Shapley, L.S. (1959), “Equilibrium points in games with vector payoff”, *Naval Research Logistics Quarterly*, 6:57-61.

Singh, H. (1999), Introduction to game theory and its application in electric power systems. *IEEE Computer Applications in Power*, 12(4):18–22.

Wei-Liu, J., Kunyue, N. e Hong, Y. (2006), “An approach for solving fuzzy games.” *Fuzziness and Knowledge-Based Systems*, Vol 14, No 3:277–292.

Wiegand, R. P.(2003), *An Analysis of Cooperative Coevolutionary Algorithms*. George Mason University, Fairfax, Virginia.

Yao, X., Darwen, P. J. (2001), “Why more choices cause less cooperation in iterated prisoner’s dilemma”, *Congress on Evolutionary Computation, CEC*, 2:987-994.

Zadeh, L. (1965), Fuzzy sets. *Information and Control*, 8:338–353.

Zeleny, M. (1975), “Games with multiple payoffs”, *International Journal of Game Theory*, 4:179-191.

Zhu, Qing e Lee, E.S.(1992), “Comparison and ranking of fuzzy numbers. *Fuzzy regression analysis*.” J. Kacprzyck and M. Fedrizzi, Omnitech Press, Heidelberg, 21-44